

函数 workflow

API 参考

文档版本 01
发布日期 2025-02-08



版权所有 © 华为云计算技术有限公司 2025。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 使用前必读	1
2 函数模型介绍	3
2.1 FunctionGraph 函数模型	3
2.2 函数 Trigger Management 触发器模型	9
3 API 概览	16
4 如何调用 API	17
4.1 构造请求	17
4.2 认证鉴权	20
4.3 返回结果	21
5 应用示例	23
5.1 示例 1: 创建函数和 Timer 触发器实现定时从 OBS 桶中下载文件	23
5.2 示例 2: 创建函数和 APIG 触发器获取静态页面	26
5.3 示例 3: 上传代码到 OBS 桶, 并使用 OBS 链接创建函数	29
5.4 示例 4: 函数互调	31
6 扩展及遥测 API	34
6.1 扩展 API	34
6.1.1 概述	34
6.1.2 操作步骤	34
6.1.3 制作扩展依赖包	34
6.1.4 扩展 API 参考	35
6.1.4.1 API 方法	35
6.1.5 如何通过扩展完成 SkyWalking 的对接	37
6.2 遥测 API	38
6.2.1 概述	38
6.2.1.1 如何在扩展中使用 OpenTelemetry API	39
6.2.1.2 注册 OpenTelemetry API	39
6.2.1.3 处理遥测	40
7 API	43
7.1 函数调用	43
7.1.1 同步执行函数	43
7.1.2 异步执行函数	50

7.1.3 函数异步执行并返回预留实例 ID (已废弃)	55
7.2 函数配额	57
7.2.1 查询租户配额	58
7.3 函数依赖包	63
7.3.1 获取依赖包列表	63
7.3.2 创建依赖包 (待废弃)	71
7.3.3 创建依赖包版本	78
7.3.4 获取依赖包版本列表	89
7.3.5 获取依赖包版本详情	97
7.3.6 删除依赖包版本	103
7.3.7 删除指定的依赖包 (待废弃)	107
7.3.8 获取指定依赖包 (待废弃)	110
7.3.9 更新指定依赖包	113
7.4 函数测试事件	120
7.4.1 获取指定函数的测试事件列表	120
7.4.2 创建测试事件	125
7.4.3 删除指定测试事件	131
7.4.4 获取测试事件详细信息	135
7.4.5 更新测试事件详细信息	140
7.5 函数调用链	145
7.5.1 获取函数调用链配置	146
7.5.2 修改函数调用链配置	150
7.6 函数生命周期管理	155
7.6.1 获取函数列表	155
7.6.2 创建函数	165
7.6.3 删除函数/版本	195
7.6.4 获取指定函数代码信息	200
7.6.5 修改函数代码	210
7.6.6 获取函数的 metadata	225
7.6.7 修改函数的 metadata 信息	240
7.6.8 更新函数最大实例数	265
7.6.9 查询函数标签列表	279
7.6.10 禁用/启动函数快照	284
7.6.11 获取指定函数绑定的 servicebridge 函数列表	289
7.6.12 查询函数快照制作状态	298
7.6.13 查询资源标签	303
7.6.14 查询资源实例	308
7.6.15 删除资源标签	316
7.6.16 创建资源标签	321
7.6.17 创建下沉入口	326
7.6.18 删除下沉入口	332
7.6.19 更新函数置顶状态	335

7.6.20 获取 servicebridge 可用的版本.....	339
7.7 函数版本别名.....	344
7.7.1 获取指定函数所有版本别名列表.....	344
7.7.2 创建函数版本别名.....	349
7.7.3 删除函数版本别名.....	362
7.7.4 获取函数版本的指定别名信息.....	366
7.7.5 修改函数版本别名信息.....	372
7.7.6 获取指定函数的版本列表.....	381
7.7.7 发布函数版本.....	393
7.8 函数指标.....	407
7.8.1 获取按指定指标排序的函数列表.....	407
7.8.2 租户函数统计信息.....	411
7.8.3 获取指定时间段的函数运行指标.....	419
7.8.4 查询函数实例使用情况指标.....	424
7.8.5 查询函数实例流量指标.....	429
7.9 函数日志.....	435
7.9.1 查询异步日志详情.....	435
7.9.2 获取指定函数的 lts 日志组日志流配置.....	439
7.9.3 开通 lts 日志上报功能.....	443
7.9.4 允许异步状态通知.....	447
7.10 函数模板.....	450
7.10.1 获取函数模板列表.....	450
7.10.2 获取指定函数模板.....	457
7.11 函数预留实例.....	463
7.11.1 获取函数预留实例配置列表.....	464
7.11.2 修改函数预留实例数量.....	471
7.11.3 获取函数预留实例数量.....	478
7.12 函数导入导出.....	484
7.12.1 导出函数.....	484
7.12.2 导入函数.....	489
7.13 函数触发器.....	498
7.13.1 删除指定函数的所有触发器.....	498
7.13.2 获取指定函数的所有触发器.....	503
7.13.3 创建触发器.....	516
7.13.4 删除触发器.....	556
7.13.5 获取指定触发器的信息.....	561
7.13.6 更新触发器.....	575
7.14 函数流.....	588
7.14.1 同步执行工作流.....	589
7.14.2 异步执行工作流.....	595
7.14.3 删除函数流.....	601
7.14.4 查询函数流.....	606

7.14.5 创建函数流.....	612
7.14.6 获取指定函数流执行实例列表.....	627
7.14.7 获取指定函数流执行实例.....	633
7.14.8 获取指定函数流实例的元数据.....	641
7.14.9 修改指定函数流实例的元数据.....	652
7.14.10 获取函数流指标.....	667
7.14.11 获取指定函数流指标.....	672
7.14.12 重试函数流.....	677
7.14.13 停止函数流.....	682
7.14.14 分页查询指定函数流记录.....	686
7.14.15 回调工作流.....	693
7.15 函数异步配置.....	698
7.15.1 获取函数异步配置信息.....	698
7.15.2 删除函数异步配置信息.....	704
7.15.3 设置函数异步配置信息.....	707
7.15.4 获取函数异步配置列表.....	715
7.15.5 获取函数异步调用请求列表.....	721
7.15.6 获取函数活跃异步调用请求列表.....	728
7.15.7 停止函数异步调用请求.....	735
7.16 函数应用中心.....	741
7.16.1 查询应用程序模板详情.....	741
7.16.2 查询应用程序模板列表.....	746
7.16.3 查询应用程序列表.....	752
7.16.4 创建应用程序.....	758
7.16.5 查询应用程序详情.....	764
7.16.6 删除应用程序.....	770
8 权限及授权项说明.....	776
9 附录.....	780
9.1 状态码.....	780
9.2 错误码.....	781
9.3 获取账号、IAM 用户、项目、用户组、区域、委托的名称和 ID.....	802
9.4 FunctionGraph 服务监控指标说明.....	803
10 修订记录.....	805

1 使用前必读

欢迎使用函数工作流（FunctionGraph）。FunctionGraph是一项基于事件驱动的函数托管计算服务。通过函数工作流，只需编写业务函数代码并设置运行的条件，无需配置和管理服务器等基础设施，函数以弹性、免运维、高可靠的方式运行。此外，按函数实际执行资源计费，不执行不产生费用。

您可以使用本文档提供API对函数资源进行相关操作，如创建、删除、查询、执行函数等。支持的全部操作请参见[API概览](#)。

FunctionGraph提供了REST（Representational State Transfer）风格API，支持您通过HTTPS请求调用，调用方法请参见[如何调用API](#)。

约束与限制

- 您能创建的函数数量与配额有关系，具体请参见[关于配额](#)。
- 更详细的限制请参见具体API的说明。

终端节点

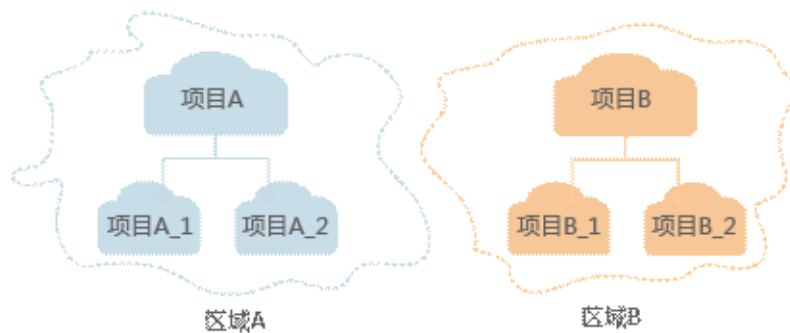
终端节点即调用API的[请求地址](#)，不同服务不同区域的终端节点不同，您可以从[地区和终端节点](#)中查询服务的终端节点。

基本概念

- 账号
您注册公有云服务时的账号，账号对其所拥有的资源及云服务具有完全的访问权限，可以重置用户密码、分配用户权限等。由于账号是付费主体，为了确保账号安全，建议您不要直接使用账号进行日常管理工作，而是创建用户并使用其进行日常管理工作。
- 用户
由账号在IAM中创建的用户，是云服务的使用人员，具有身份凭证（密码和访问密钥）。
在[我的凭证](#)下，您可以查看账号ID和用户ID。通常在调用API的鉴权过程中，您需要用到账号、用户和密码等信息。
- 区域
指云资源所在的物理位置，同一区域内可用区间内网互通，不同区域间内网不互通。通过在不同地区创建云资源，可以将应用程序设计的更接近特定客户的要求，或满足不同地区的法律或其他要求。

- 可用区
一个可用区是一个或多个物理数据中心的集合，有独立的风火水电，AZ内逻辑上再将计算、网络、存储等资源划分成多个集群。一个Region中的多个AZ间通过高速光纤相连，以满足用户跨AZ构建高可用性系统的需求。
- 项目
公有云区域默认对应一个项目，这个项目由系统预置，用来隔离物理区域间的资源（计算资源、存储资源和网络资源），以默认项目为单位进行授权，用户可以访问您账号中该区域的所有资源。如果您希望进行更加精细的权限控制，可以在区域默认的项目中创建子项目，并在子项目中购买资源，然后以子项目为单位进行授权，使得用户仅能访问特定子项目中资源，使得资源的权限控制更加精确。

图 1-1 项目隔离模型



- 企业项目
企业项目是项目的升级版，针对企业不同项目间资源的分组和管理，是逻辑隔离。企业项目中可以包含多个区域的资源，且项目中的资源可以迁入迁出。关于企业项目ID的获取及企业项目特性的详细信息，请参见《[企业管理服务用户指南](#)》。

2 函数模型介绍

本节内容主要介绍FunctionGraph函数模型和Trigger Management触发器模型的返回字段。通过提前熟悉返回字段的参数说明，方便您快速了解API接口的功能，详细信息请参考[API](#)。

2.1 FunctionGraph 函数模型

介绍FunctionGraph函数模型的返回字段。

函数模型

FunctionGraph函数模型如下。

```
{
  "functions": [
    {
      "func_urn":
"urn:fss:xxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",
      "func_name": "test",
      "domain_id": "cff01_hk",
      "namespace": "7aad83af3e8d42e99ac194e8419e2c9b",
      "project_name": "xxxxxxxx",
      "package": "default",
      "runtime": "Node.js6.10",
      "timeout": 3,
      "handler": "test.handler",
      "memory_size": 128,
      "cpu": 300,
      "code_type": "inline",
      "code_url": "",
      "code_filename": "index.js",
      "code_size": 272,
      "user_data": "",
      "digest":
"decfce6939297b0b5ec6d1a23bf9c725870f5e69fc338a89a6a4029264688dc26338f56d08b6535d
e47f15ad538e22ca66613b9a46f807d50b687bb53fded1c6",
      "version": "latest",
      "image_name": "latest-5qe8e",
      "xrole": "cff",
      "app_xrole": null,
      "description": "111",
      "version_description": ""
    }
  ]
}
```

```

    "last_modified": "2018-03-28T11:30:32+08:00",
    "func_code": {
      "file": "",
      "link": ""
    },
    "func_vpc": null,
    "mount_config": null,
    "depend_list": null,
    "strategy_config": {
      "concurrency": -1
    },
    "extend_config": "",
    "dependencies": null,
    "initializer_handler": "index.initializer",
    "initializer_timeout": 3
  }
],
"next_marker": 45
}

```

字段说明

FunctionGraph函数字段说明如表2-1所示。

表 2-1 FunctionGraph 函数字段说明

名称	说明
func_urn	函数的URN (Uniform Resource Name) , 唯一标识函数。
func_name	函数名称。
domain_id	租户名称。
namespace	租户的Project ID。
project_name	租户的Project Name。
package	函数所属的分组Package, 用于用户针对函数的自定义分组。
runtime	FunctionGraph函数的执行环境, 支持Node.js 6.10、Node.js 8.10、Node.js 10.16、Node.js 12.13、Node.js 14.18、Node.js 16.17、Node.js 18.15、Python 2.7、Python 3.6、Python 3.9、Python 3.10、Java 8、Java 11、Go 1.x、C#.NET Core 2.1、C#.NET Core 3.1、Cangjie 1.0、PHP 7.3、Custom: 自定义运行时、http: HTTP函数和Custom Image: 自定义镜像函数。
timeout	函数执行超时时间, 超时函数将被强行停止, 范围3~900秒。
handler	函数执行入口, 规则: xx.xx, 必须包含“.”。 举例: 对于Node.js函数: myfunction.handler, 则表示函数的文件名为myfunction.js, 执行的入口函数名为handler。
memory_size	函数消耗的内存, 单位M。 取值范围为: 128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096、8192、10240。最小值为128, 最大值为10240。

名称	说明
cpu	函数占用的CPU资源。
code_type	函数代码类型，取值有4种： <ul style="list-style-type: none"> inline：UI在线编辑代码。 zip：函数代码为zip包。 jar：函数代码为jar包，主要针对Java函数。 obs：函数代码来源于OBS存储。
code_url	<ul style="list-style-type: none"> 当“code_type”为“obs”时，该值为函数代码包在OBS上的地址。 当“code_type”为“inline”、“zip”或“jar”时，该字段为空。
code_filename	函数的文件名。 <ul style="list-style-type: none"> 当“code_type”为“zip”或“jar”时，必须提供该字段。 当“code_type”为“obs”或“inline”，不需要提供该字段。
code_size	函数大小，单位：字节。
user_data	用户自定义的name/value信息，在函数中使用的参数。 举例：如函数要访问某个主机，可以设置自定义参数：Host={host_ip}，最多定义20个，总长度不超过4KB。
digest	函数代码SHA512 hash值，用于判断函数是否变化。
version	函数版本号，由系统自动生成，规则：vYYYYMMDD-HHMMSS（v+年月日-时分秒）。
image_name	函数版本的内部标识。
xrole	函数使用的权限委托名称，需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。
app_xrole	函数app使用的权限委托名称，需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。
description	函数描述。
version_description	函数版本描述。
last_modified	函数最后一次更新时间。
func_code	函数代码，请参考 表2-2 。
depend_list	依赖包列表。
strategy_config	函数策略配置，请参考 表2-3 。
extend_config	函数扩展配置。
dependencies	依赖代码包列表，请参考 表2-5 。

名称	说明
initializer_handler	函数初始化入口，规则：xx.xx，必须包含“.”。 举例：对于Node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。
initializer_timeout	初始化超时时间，超时函数将被强行停止，范围1 ~ 300秒。
func_vpc	vpc配置，请参考表2-4。
mount_config	文件系统配置，请参考表6 mount_config参数说明。

表 2-2 func_code 参数说明

参数	说明
file	函数代码内容，接口返回空。
link	函数代码链接，接口返回空。

表 2-3 strategy_config 参数说明

参数	说明
concurrency	<ul style="list-style-type: none"> 0：函数被禁用。 -1：函数被启用。

表 2-4 func_vpc 参数说明

参数	类型	是否必选	说明
vpc_name	String	否	虚拟私有云名称
vpc_id	String	当func_vpc非空时必选	虚拟私有云唯一标识
subnet_name	String	否	子网名称
subnet_id	String	当func_vpc非空时必选	子网编号
cidr	String	否	子网掩码
gateway	String	否	网关

表 2-5 dependency 参数说明

参数	类型	是否必选	说明
owner	String	否	依赖包属主的domainId
link	String	否	依赖包在OBS上的链接
runtime	String	否	依赖包语言类型（仅作为分类条件）
etag	String	否	依赖包md5值
size	Int	否	依赖包大小
name	String	否	依赖包名称
description	String	否	依赖包描述
file_name	String	否	依赖包文件名（如果创建方式为zip时）

表 2-6 mount_config 参数说明

参数	类型	是否必选	说明
mount_user	mount_user	否	文件系统用户配置
func_mounts	func_mounts	否	文件系统列表

表 2-7 mount_user 参数说明

参数	类型	是否必选	说明
user_id	Int	mount_user非空时必选	用户ID，-1 ~ +65534的整数(0、1000、1002除外)
user_group_id	Int	mount_user非空时必选	用户组ID，-1 ~ +65534的整数(0、1000、1002除外)

表 2-8 func_mounts 参数说明

参数	类型	是否必选	说明
mount_type	String	func_mounts非空时必选	挂载类型（sfs/sfsTurbo/ecs）
mount_resource	String	func_mounts非空时必选	挂载资源ID（对应云服务ID）
mount_share_path	String	“mount_type”为“ecs”时，必选	远端挂载路径（例如192.168.0.12:/data）

参数	类型	是否必选	说明
local_mount_path	String	func_mounts非空时必选	函数访问路径

funcurn 格式说明

```
urn:fss:<region_id>:<project_id>:function:<package>:<function_name>[:<version>]:!<alias>]
```

📖 说明

Funcurn使用冒号分隔为8个域，region_id为系统配置的值（测试时可以填成与后台相同的值），最后一个中括号的内容表示可以为函数的版本号，也可以为该版本指向的别名，如果为别名时需要在名称前加上感叹号！以示区分。

当作为API参数时，可以提供简化格式的Funcurn，具体如下。

- 提供1个域：解析为<function_name>，“project_id”从token中获取，“package”为“default”，“version”为“latest”。
- 提供2个域：解析为<package>:<function_name>，“project_id”从token中获取，“version”为“latest”。
- 提供3个域：解析为<project_id>:<package>:<function_name>，“version”为“latest”。
- 提供4个域：解析为<project_id>:<package>:<function_name>:<Version or Alias>。
- 提供7个域：解析为
urn:fss:<region_id>:<project_id>:function:<package>:<function_name>，“version”为“latest”。
- 提供8个域：解析为
urn:fss:<region_id>:<project_id>:function:<package>:<function_name>:<Version or Alias>。

函数实例数据

```
{
  "func_urn": "urn:fss:xxxxxxxx:73d69ae0cfcf460190522d060f05ad:function:default:auto_testfunc93749",
  "func_name": "auto_testfunc93749",
  "domain_id": "b8aca445e0d04d81a34bb59de5280c72",
  "namespace": "73d69ae0cfcf460190522d060f05ad",
  "project_name": "xxxxxxxx",
  "package": "default",
  "runtime": "Python2.7",
  "timeout": 5,
  "handler": "index.handler",
}
```

```
"memory_size":128,
"cpu":300,
"code_type":"inline",
"code_filename":"index.py",
"code_size":1992,
"version":"latest",
"image_name": "latest-200731100126@obffv",
"description": "使用API网关调用函数，在函数中演示了返回不同的内容，如HTML页面和json结构体。",
"last_modified": "2020-07-31T10:01:26+08:00",
"func_code": {},
"FuncCode": {},
"concurrency": -1,
"strategy_config": {
"concurrency": -1
},
"enterprise_project_id": "0"
}
```

2.2 函数 Trigger Management 触发器模型

介绍Trigger Management触发器模型的返回字段。

触发器类型模型

```
{
  "trigger_type_code":"string",
  "display_name":"string",
  "status":"string",
  "event_codes":"array of string",
  "description":"string"
}
```

触发器类型模型字段说明如[表2-9](#)所示。

表 2-9 触发器类型模型字段说明

字段名称	字段说明
trigger_type_code	触发器类型代码，取值：SMN、APIG、TIMER、DMS、DIS、LTS、DDS、CTS、kafka、DEDICATEDGATEWAY、APIC。
display_name	触发器类型显示值。

字段名称	字段说明
status	触发器类型状态，可能的状态如下： <ul style="list-style-type: none"> • DISABLED：禁用该触发器。 • TEST：触发器处于测试状态，用户端应设为不可见。 • ACTIVE：触发器可用。
event_codes	触发器事件属性字段。
description	触发器说明。

触发器实例模型

```
{
  "trigger_id":"string",
  "trigger_type_code":"string",
  "event_type_code":"string",
  "status":"string",
  "event_data":"json struct",
  "last_updated_time":"string",
  "created_time":"string"
}
```

触发器实例模型字段说明如[表2-10](#)所示。

表 2-10 触发器实例模型字段说明

字段名称	字段说明
trigger_id	触发器ID。
trigger_type_code	触发器类型代码，取值：SMN、APIG、TIMER、DMS、DIS、LTS、DDS、CTS、kafka。
event_type_code	事件类型代码，必填字段，可以填任意非空字符串（ deprecated ）。
status	触发器状态：ACTIVE / DISABLED。
event_data	触发器相关定义数据，以JSON结构形式出现。
last_updated_time	最后一次更新时间。
created_time	创建时间。

触发器实例数据

- SMN触发器数据如下。

```
{
  "topic_urn":"string",
  "subscription_status":"string"
}
```

SMN触发器数据说明如[表2-11](#)所示。

表 2-11 SMN 触发器数据说明

字段名称	字段说明
topic_urn	SMN服务的topic_urn，创建时必须填。
subscription_status	topic_urn的订阅状态：Unconfirmed / Confirmed。

- DMS触发器数据如下。

```
{
  "queue_id": "string",
  "consumer_group_id": "string",
  "polling_interval": "int"
}
```

DMS触发器数据说明如表2-12所示。

表 2-12 DMS 触发器数据说明

字段名称	字段说明
queue_id	DMS队列名，创建时必须填。
consumer_group_id	DMS消费组名，创建时必须填。
polling_interval	检查消息间隔，以秒为单位，缺省为30秒，创建时必须填。

- DIS触发器数据如下。

```
{
  "stream_name": "dis-qYPJ",
  "polling_interval": 30,
  "batch_size": 100,
  "sharditerator_type": "TRIM_HORIZON"
}
```

DIS触发器数据说明如表2-13所示。

表 2-13 DIS 触发器数据说明

字段名称	字段说明
stream_name	通道名称，必填。
polling_interval	拉取周期，取值1 ~ 60，选填（不填会取默认值30）。
batch_size	每次拉取的记录数量，取值1 ~ 10000，选填（不填时，取默认值100）。
sharditerator_type	取值TRIM_HORIZON（从头开始拉取）/LATEST（从当前位置开始拉取），必填。

- APIG触发器数据如下。

```
{
  "group_id": "string",
  "env_id": "string",
  "auth": "string",
  "protocol": "string",
  "name": "string",
  "path": "string",
  "match_mode": "string",
  "req_method": "string",
  "backend_type": "string",
  "type": int,
  "sl_domain": "string",
  "instance_id": "string"
}
```

APIG触发器数据说明如表2-14所示。

表 2-14 APIG 触发器数据说明

字段名称	字段说明
group_id	API分组，必填。
env_id	API发布环境，必填。
auth	API认证方式，三种可选模式： NONE/IAM/APP，必填。
protocol	访问协议，HTTP或HTTPS，必填。
name	API名称，必填。
path	API访问地址，需要服从URL规范，如/a/b，必填。
match_mode	匹配模式，当前只支持前缀匹配模式（SWA），必填。
req_method	API请求方式，枚举如： GET、POST、PUT，必填。
backend_type	后端类型： FUNCTION，必填。
type	API类型，“1”表示开放API类型，当前只支持该类型，必填。
sl_domain	子域名，必填。
instance_id	实例ID，触发器类型"trigger_type_code"取值为“DEDICATEDGATEWAY”或者“APIC”必填

- TIMER触发器数据如下。

```
{
  "name": "string",
  "schedule_type": "string",
  "schedule": "string",
  "user_event": "string"
}
```

TIMER触发器数据说明如表2-15所示。

表 2-15 TIMER 触发器数据说明

字段名称	字段说明
name	触发器名称，必填。
schedule_type	调度类型，两种可选模式：Rate/Cron，必填。
schedule	调度配置，分别根据不同的调度类型填写，必填。 当选择Rate模式时，格式为数字加上m（分钟），h（小时），d（天），如3分钟为3m。
user_event	调用附加信息，作为调用函数时的参数，可选。

- LTS触发器数据如下。

```
{
  "trigger_type_code": "LTS",
  "event_type_code": "MessageCreated",
  "trigger_status": "ACTIVE",
  "event_data": {
    "log_group_id": "3e4d3bf7-7bad-11e9-92c5-fa163e6216be",
    "log_topic_id": "41d90375-7bad-11e9-8bcf-fa163ea23ac3",
    "log_group_name": "lts-group-5b42",
    "log_topic_name": "lts-topic-5f3e"
  }
}
```

LTS数据说明如所表2-16示。

表 2-16 LTS 触发器数据说明

字段名称	字段说明
trigger_type_code	触发器类型。
event_type_code	事件类型。
trigger_status	触发器状态。
event_data	触发器相关定义数据，以JSON结构形式出现。
log_group_id	LTS日志组，创建时必选。
log_topic_id	LTS日志主题，创建时必选。
log_group_name	LTS日志组名称。
log_topic_name	LTS日志主题名称。

- CTS触发器数据如下。

```
{
  "name": "eqwrwe",
```

```
"operations": ["AAD:addprotocolrule:addProtocolRule", "BCS:baas-apiserver:scalePeers",
"ARS:ars:setConfigArs"]
}
```

CTS触发器数据说明如表2-17所示。

表 2-17 CTS 触发器数据说明

字段名称	字段说明
name	关键通知名称。
operations	操作列表。 格式为：服务类型:资源类型A;资源类型B:操作名称1;操作名称2，例如：["ECS:ecs;server:restartServer;deleteServer",...]。

- DDS触发器数据如下。

```
{
  "instance_id": "string",
  "collection_name": "string",
  "db_name": "string",
  "db_password": string,
  "batch_size": int,
}
```

表 2-18 DDS 触发器数据说明

字段名称	字段说明
instance_id	文档数据库实例ID
collection_name	集合名称
db_name	文档数据库名称
db_password	文档数据库用户密码
batch_size	批次大小

- Kafka触发器数据如下。

```
{
  "instance_id": "string",
  "topic_id": "[]string",
  "kafka_user": "string",
  "kafka_password": "string",
  "kafka_ssl_enable": string,
  "batch_size": int,
}
```

表 2-19 Kafka 触发器数据说明

字段名称	字段说明
instance_id	Kafka实例ID
topic_id	主题ID

字段名称	字段说明
kafka_user	用户名
kafka_password	用户密码
kafka_ssl_enable	是否开启ssl验证(若开启则用户名和密码字段是为必选)
batch_size	批次大小

3 API 概览

FunctionGraph API为开发者、合作伙伴提供开发、部署、托管、运维的开放接口，帮助用户快速、低成本地实现业务创新，缩短应用上线周期。

FunctionGraph API提供的接口有如下几种类型。

表 3-1 接口简介

类型	说明
函数调用	包括同步执行函数、异步执行函数、函数异步执行并返回预留实例ID接口。
函数配额	查询租户配额。
函数依赖包	包括创建、删除、获取、更新依赖包等接口。
函数测试事件	包括创建、删除、获取、更新测试事件等接口。
函数调用链	获取、修改调用链配置。
函数生命周期管理	包括创建、删除函数等接口。
函数版本别名	包括创建、删除函数版本别名等接口。
函数指标	包括获取按指定指标排序的函数列表、租户函数统计信息、获取指定时间段的函数运行指标接口。
函数日志	获取指定函数的lts日志组日志流配置、开通lts日志上报功能。
函数预留实例	获取函数预留实例配置列表、修改函数预留实例数量。
函数导入导出	导入、导出函数。
函数触发器	包括创建、删除、更新触发器等接口。
函数流	查询、删除、创建、停止函数流等接口。
函数异步配置	获取、删除、设置函数异步配置信息等接口。
函数应用中心	包括查询、创建、删除应用程序等接口。

4 如何调用 API

4.1 构造请求

本节介绍REST API请求的组成，并以调用IAM服务的**获取用户Token**接口说明如何调用API，该API获取用户的Token，Token可以用于调用其他API时鉴权。

您还可以通过这个视频教程了解如何构造请求调用API：<https://bbs.huaweicloud.com/videos/102987>。

请求 URI

请求URI由如下部分组成。

{URI-scheme} :// {Endpoint} / {resource-path} ? {query-string}

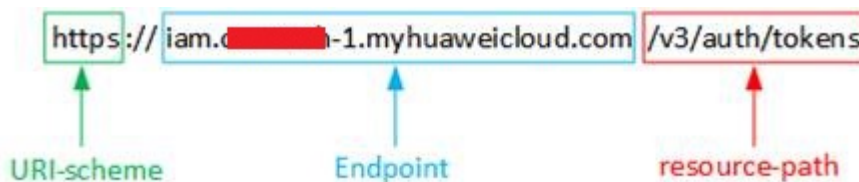
尽管请求URI包含在请求消息头中，但大多数语言或框架都要求您从请求消息中单独传递它，所以在此单独强调。

- **URI-scheme**：表示用于传输请求的协议，当前所有API均采用HTTPS协议。
- **Endpoint**：指定承载REST服务端点的服务器域名或IP，不同服务不同区域的Endpoint不同，您可以从**地区和终端节点**中获取。例如IAM服务在“华北-北京一”区域的Endpoint为“iam.cn-north-1.myhuaweicloud.com”。
- **resource-path**：资源路径，也即API访问路径。从具体API的URI模块获取，例如“获取用户Token”API的resource-path为“/v3/auth/tokens”。
- **query-string**：查询参数，是可选部分，并不是每个API都有查询参数。查询参数前面需要带一个“？”，形式为“参数名=参数取值”，例如“limit=10”，表示查询不超过10条数据。

例如您需要获取IAM在“华北-北京一”区域的Token，则需使用“华北-北京一”区域的Endpoint（iam.cn-north-1.myhuaweicloud.com），并在**获取用户Token**的URI部分找到resource-path（/v3/auth/tokens），拼接起来如下所示。

```
https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
```

图 4-1 URI 示意图



说明

为查看方便，在每个具体API的URI部分，只给出resource-path部分，并将请求方法写在一起。这是因为URI-scheme都是HTTPS，而Endpoint在同一个区域也相同，所以简洁起见将这两部分省略。

请求方法

HTTP请求方法（也称为操作或动词），它告诉服务你正在请求什么类型的操作。

- **GET**：请求服务器返回指定资源。
- **PUT**：请求服务器更新指定资源。
- **POST**：请求服务器新增资源或执行特殊操作。
- **DELETE**：请求服务器删除指定资源，如删除对象等。
- **HEAD**：请求服务器资源头部。
- **PATCH**：请求服务器更新资源的部分内容。当资源不存在的时候，PATCH可能会去创建一个新的资源。

在[获取用户Token](#)的URI部分，您可以看到其请求方法为“POST”，则其请求为：

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
```

请求消息头

附加请求头字段，如指定的URI和HTTP方法所要求的字段。例如定义消息体类型的请求头“Content-Type”，请求鉴权信息等。

如下公共消息头需要添加到请求中。

- **Content-Type**：消息体的类型（格式），必选，默认取值为“application/json”，有其他取值时会在具体接口中专门说明。
- **X-Auth-Token**：用户Token，可选，当使用Token方式认证时，必须填充该字段。用户Token也就是调用[获取用户Token](#)接口的响应值，该接口是唯一不需要认证的接口。

说明

FunctionGraph API同时支持使用AK/SK认证，AK/SK认证是使用SDK对请求进行签名，签名过程会自动往请求中添加Authorization（签名认证信息）和X-Sdk-Date（请求发送的时间）请求头。

AK/SK认证的详细说明请参见[AK/SK认证](#)。

- **X-Project-ID**：子项目ID，可选，在多项目场景中使用。
- **X-Domain-ID**：账号ID。

对于**获取用户Token**接口，由于不需要认证，所以只添加“Content-Type”即可，添加消息头后的请求如下所示。

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

请求消息体

请求消息体通常以结构化格式发出，与请求消息头中Content-type对应，传递除请求消息头之外的内容。若请求消息体中参数支持中文，则中文字符必须为UTF-8编码。

每个接口的请求消息体内容不同，也并不是每个接口都需要有请求消息体（或者说消息体为空），GET、DELETE操作类型的接口就不需要消息体，消息体具体内容需要根据具体接口而定。

对于**获取用户Token**接口，您可以从接口的请求部分看到所需的请求参数及参数说明。将消息体加入后的请求如下所示，加粗的斜体字段需要根据实际值填写，其中***username***为用户名，***domainname***为用户所属的账号名称，***********为用户登录密码，***xxxxxxxxxxxxxxxxxxxx***为project的名称，可以从**地区和终端节点**中获取。

说明

scope参数定义了Token的作用域，上面示例中获取的Token仅能访问project下的资源。您还可以设置Token作用域为某个账号下所有资源或账号的某个project下的资源，详细定义请参见**获取用户Token**。

```
POST https://iam.cn-north-1.myhuaweicloud.com/v3/auth/tokens
Content-Type: application/json
```

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

到这里为止这个请求需要的内容就具备齐全了，您可以使用**curl**、**Postman**或直接编写代码等方式发送请求调用API。对于**获取用户Token**接口，返回的响应消息头中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

4.2 认证鉴权

调用接口有如下两种认证方式，您可以选择其中一种进行认证鉴权。

- Token认证：通过Token认证通用请求。
- AK/SK认证：通过AK（Access Key ID）/SK（Secret Access Key）加密调用请求。

Token 认证

📖 说明

Token的有效期为24小时，需要使用一个Token鉴权时，可以先缓存起来，避免频繁调用。

Token在计算机系统中代表令牌（临时）的意思，拥有Token就代表拥有某种权限。Token认证就是在调用API的时候将Token加到请求消息头，从而通过身份认证，获得操作API的权限。

Token可通过调用**获取用户Token**接口获取，调用本服务API需要project级别的Token，即调用**获取用户Token**接口时，请求body中auth.scope的取值需要选择project，如下所示。

```
{
  "auth": {
    "identity": {
      "methods": [
        "password"
      ],
      "password": {
        "user": {
          "name": "username",
          "password": "*****#",
          "domain": {
            "name": "domainname"
          }
        }
      }
    },
    "scope": {
      "project": {
        "name": "xxxxxxx"
      }
    }
  }
}
```

获取Token后，再调用其他接口时，您需要在请求消息头中添加“X-Auth-Token”，其值即为Token。例如Token值为“ABCDEFJ....”，则调用接口时将“X-Auth-Token: ABCDEFJ....”加到请求消息头即可，如下所示。

```
GET https://iam.cn-north-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

您还可以通过这个视频教程了解如何使用Token认证：<https://bbs.huaweicloud.com/videos/101333>。

AK/SK 认证

说明

- AK/SK签名认证方式仅支持消息体大小12M以内，12M以上的请求请使用Token认证。
- 当AK/SK签名调用函数时，需传入x-project-id请求头（目前利雅得局点和有子项目场景必传，其他局点可不传入）。

AK/SK认证就是使用AK/SK对请求进行签名，在请求时将签名信息添加到消息头，从而通过身份认证。

- AK（Access Key ID）：访问密钥ID。与私有访问密钥关联的唯一标识符；访问密钥ID和私有访问密钥一起使用，对请求进行加密签名。
- SK（Secret Access Key）：与访问密钥ID结合使用的密钥，对请求进行加密签名，可标识发送方，并防止请求被修改。

使用AK/SK认证时，您可以基于签名算法使用AK/SK对请求进行签名，也可以使用专门的签名SDK对请求进行签名。详细的签名方法和SDK使用方法请参见[API签名指南](#)。

须知

签名SDK只提供签名功能，与服务提供的SDK不同，使用时请注意。

4.3 返回结果

状态码

请求发送以后，您会收到响应，包含状态码、响应消息头和消息体。

状态码是一组从1xx到5xx的数字代码，状态码表示了请求响应的状态，完整的状态码列表请参见[状态码](#)。

对于[获取用户Token](#)接口，如果调用后返回状态码为“201”，则表示请求成功。

响应消息头

对应请求消息头，响应同样也有消息头，如“Content-type”。

对于[获取用户Token](#)接口，返回如[图4-2](#)所示的消息头，其中“x-subject-token”就是需要获取的用户Token。有了Token之后，您就可以使用Token认证调用其他API。

5 应用示例

5.1 示例 1：创建函数和 Timer 触发器实现定时从 OBS 桶中下载文件

场景描述

本章节以Python 2.7为例，指导您通过API创建FunctionGraph函数和Timer触发器，实现定时从OBS桶中下载文件。

API的调用方法请参见[如何调用API](#)。

前提条件

- 已在OBS服务中上传文件，并记录文件名、文件所在的OBS桶名，以及OBS地址的链接。
- 已配置具有OBS服务访问权限的委托，并记录委托名。

总体思路

创建FunctionGraph函数和Timer触发器，实现定时从OBS桶中下载文件，步骤如下：

1. **创建函数**：创建下载文件的函数。
2. **修改函数的metadata信息**：修改函数配置信息中的OBS地址、OBS桶名和文件名。
3. **同步执行函数**：确认函数能够成功从OBS桶中下载文件。
4. **创建触发器**：创建Timer触发器，实现定时下载文件。

步骤 1：创建下载文件的函数

URI: POST /v2/{project_id}/fgs/functions

API文档详情请参见：[创建函数](#)

- 请求示例
POST https://{Endpoint}/v2/{project_id}/fgs/functions
{

```
"code_filename": "index.zip",
"code_type": "inline",
"func_code": {
  "file":
  "UESDBAoAAAAIABESwIDHSM8cOQYAAJYRAAAIAAAAaW5kZXgucHm9V91v2zYQf9dfcXAeLKeKkrYZNg
  TQQ9JvpG2CJsOwJ4GWajuLRGok5VT963dHUpZsJ82GofVDQh2Pd8e7333wA14Oj6BQpZDLM2jt4ug3ok
  QH8MeKS9CtlgDdiUMGFY3F5dmDIYBKwpuDFxd3CTQqRbq1lgwDS/
  EogMmgS25LDq4F3YFy0rNWQWGG67UoeH+04boWxgglDcRKA7NQcYZS3BkU/
  ADjLI0WWtVoRZ2quUmLSnBpaZn7JYi6UdrC1dy8coRt/hqtr0y65DZX8794YfMVZyXX/
  bF33F45+ntHfuqw5n+33Ni90188PQp005l+qUwUfa3Wafm+EBXPG4axZCe0iothZas5nH/
  rTmraBHnnj2fzTA652VpMCoYDS8KnBSMiuFMFyv3aeg7aMUA6vKoYdp2UIm5Zlpwk0Z0l9fCmobLmT4zB
  BZFT28+Xedfrq5u8+vz2/do6eTY1s3xBPzvAF6re1kphgYxolPgFehhQ/
  00OXgn9InSXRppvsQwopRpsRKSTWH7dwC3dAANRR9YDEupuAGpLEjOSxl4xzuvmfziF5MIQqWqiNYaJ
  NwTZt+2srCo451mzSogiDuc4n053pCj9lvdctj7/WjtYimZ9QZM16c/+e40Bxmncnr6Eh34/
  WjuCKTe2qx72/g/
  WfkD1gZWlpmqCzneVQx5JdMnq6Ne07lb3RaXaEqmqng7ArpmQQOklauHSjVAdqlgKvxxuiCVfsLaysGZV
  y9MI0/M9mljhLckpRTAtQk5Yhb2YrZHVeiwu0vKvndYwkdZKLZe4mfVkJYfHS2ewaO/
  Ayx3Fm2lFEQJgCvLJNroZlBqO/
  KBI2/7ZKQP76VfM8vi6YhlOnOHjC4u2uKO28eObBims7F9byRFLZwxdufci8L4ae9KVtgWTVLx9DNUFw
  0w9CKsIL9PeFCy2n0ncd4wJLBrHjPle5oTWGB74l9dA0wcT2pRV
  +TsEluRV9DGz0qiVD22L1B9iicwp1xrpePjNBU9jsCywOVaaCVRqulrrNjSXiGIB/mE/UfWopltRGqOpUJC/
  5QWeR/9iQ2dYUxYURbGJ2Ab0D7GFzRarQXFZt7tCCeKpM8XlnGi3QgBpRhEOBcVoptUdCsS6KNQ
  +Mfbqko4xIhVyoGCaHh4eD/
  WT0Guzg2UCa7Z3pj4RLngX28lk3jnYzyHJqmkx3cH7pJL253B54UjjfzDcYWhoCsMOVI6YP55+89exuG5vn
  bv
  +Sd3HlP539G15gIP1+jy6SgW134kAV1o9B6S1DcJbb5iKYC0aelmrrbgleLcGhVt1hAV0xg5y4UM4TmGYwg
  tmonsQcz9HbbQ3cEhom1AdtkaPUCEstlsqbpu5LB5vtGRATTJCRIL+xu+7We/
  +uBeTwYuTExeEEeH5bAiFqNHW3Ihv1OQKVuUDIR5DrudXrW1aaqSTW8wBP1KOC55pnTMXbVv1m6sID
  pKUNE4R4s8h2up40DDZyqQuLya7KHEZ4dXuRdGTHRWN3EeLVW/fzqy3cAh4CBtPHaCG
  +d0ZknoaenM8Rlj3QoQDsQ6Cf3ER1HthPPMw+cWx3J8G6DTNy+E2GM2v
  +NdLsrM8bs0zleNzNwl0U7XFSb3A2Xm/yVuNOA629izmfiyzSqBYRTKhmUCfjLO/L/
  EmCpHUfiWyt4yjAYew2Eloz97dtTsKz5ErO7yQrXSzr8kYEXNMZ7ZI5OQNR/V
  +F0yToIaILOrWtnfdg3PpvggqETBqAwffxPNNMwpuF/hENYi0DLqaAl2xQbDYvxXBP/
  qV7BixXMSqFUV5PTiS2EaZQq3tnBKuDeqf9JFfqQ3o0fSneq4Z/821VBv/
  eizWNmNkU7ysWOB73xELwEnMKm10wB+mtfGynZrZYDZ8SFyNa32wI6TcWx3b1B/
  mJ14hotMSF2W1TAhB0r/
  3LcOcB9qxHduSzzDTE41FNHSTkxJAK2U5W4C72l2yTH31+krpsrPsJHy93EJRtYyyB4OWsd3fywMWzfIik
  psF4il1702Q3HXPYJq4RWDsCw2xEzn15XlG3qY6sVOI+xuGuv2Us/
  v90Htesapok2a5GUowTpChauFr4fli9B7fBfnpDPprSzupm8ExPD95cTq+BTFF/
  wBQSwEChgMKAACAAREsJQx0jPHDKGAACWEQAACAAAAA8wIAAAAAaW5kZXgucHl
  QSwUGAAAAAAEAQA2AAAAXwYAAAAA"
},
"func_name": "download_file_from_obs",
"handler": "index.handler",
"memory_size": 256,
"package": "default",
"runtime": "Python2.7",
"timeout": 30
}
```

● 响应示例

```
{
  "func_urn": "urn:fss:{project_name}:{project_id}:function:default:download_file_from_obs:latest",
  "func_name": "download_file_from_obs",
  "domain_id": "89fexxd636",
  "namespace": "{project_id}",
  "project_name": "xxx",
  "package": "default",
  "runtime": "Python2.7",
  "timeout": 30,
  "handler": "index.handler",
  "memory_size": 256,
  "cpu": 400,
  "code_type": "inline",
  "code_filename": "index.zip",
  "code_size": 1707,
  "digest":
  "68891a6778848a78bd37a8c0798c91d75a5c87aee6e901303047a52edf05bf2170aac4149d79b3f6a40ef
  e78406a83bf6d8683e7b25da4f0c07e7493aa4ccdc",
  "version": "latest",
}
```

```
"image_name": "latest-200603162219@zr2ym",
"last_modified": "2020-06-03T16:22:19+08:00",
"strategy_config": {
  "concurrency": -1
},
"StrategyConfig": {},
"enterprise_project_id": "0"
}
```

记录函数的URN，即响应示例中的“func_urn”信息。

步骤 2：修改函数配置信息中的 OBS 地址、OBS 桶名和文件名

URI: PUT /v2/{project_id}/fgs/functions/{function_urn}/config

API文档详情请参见：[修改函数的metadata信息](#)

- 请求示例

```
PUT https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/config
{
  "func_name": "download_file_from_obs",
  "handler": "index.handler",
  "memory_size": 256,
  "runtime": "Python2.7",
  "timeout": 30,
  "user_data": "{\"obs_address\":\"obs.example.example.com\",\"srcBucket\":\"xxx\",\"srcObjName\":\"xxx\"}",
  "xrole": "xxx"
}
```

function_urn为[步骤1：创建下载文件的函数](#)中记录的函数URN，obs_address为OBS地址，srcBucket为OBS桶名，srcObjName为文件名，xrole为委托名。

- 响应示例

```
{
  "func_urn": "urn:fss:{project_name}:{project_id}:function:default:download_file_from_obs:latest",
  "func_name": "download_file_from_obs",
  "domain_id": "89fexxd636",
  "namespace": "{project_id}",
  "project_name": "xxx",
  "package": "default",
  "runtime": "Python2.7",
  "timeout": 30,
  "handler": "index.handler",
  "memory_size": 256,
  "cpu": 400,
  "code_type": "inline",
  "code_filename": "index.zip",
  "code_size": 1707,
  "user_data": "{\"obs_address\":\"obs.example.example.com\",\"srcBucket\":\"xxx\",\"srcObjName\":\"xxx\"}",
  "digest":
    "68891a6778848a78bd37a8c0798c91d75a5c87aee6e901303047a52edf05bf2170aac4149d79b3f6a40ef
    e78406a83bf6d8683e7b25da4f0c07e7493aa4ccdc",
  "version": "latest",
  "image_name": "latest-200603165355@varrp",
  "xrole": "xxx",
  "app_xrole": "xxx",
  "last_modified": "2020-06-03T17:25:03+08:00",
  "strategy_config": {
    "concurrency": -1
  },
  "StrategyConfig": {},
  "enterprise_project_id": "0"
}
```

步骤 3：确认函数能够成功从 OBS 桶中下载文件

URI: POST /v2/{project_id}/fgs/functions/{function_urn}/invocations

API文档详情请参见：[同步执行函数](#)

- 请求示例

```
POST https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/invocations
{
  "message": "download file"
}
```

function_urn为[步骤1：创建下载文件的函数](#)中记录的函数URN。

- 响应示例

```
"The object downloaded successfully from OBS, and the size is 14 KB"
```

步骤 4：创建 Timer 触发器，实现定时下载文件

URI: POST /v2/{project_id}/fgs/triggers/{function_urn}

API文档详情请参见：[创建触发器](#)

- 请求示例

```
POST https://{Endpoint}/v2/{project_id}/fgs/triggers/{function_urn}
{
  "event_data": {
    "name": "Timer-download",
    "schedule_type": "Rate",
    "schedule": "1d"
  },
  "event_type_code": "MessageCreated",
  "trigger_status": "ACTIVE",
  "trigger_type_code": "TIMER"
}
```

function_urn为[步骤1：创建下载文件的函数](#)中记录的函数URN。

上述请求示例表示每天从OBS桶中下载文件。

- 响应示例

```
{
  "trigger_id": "461bbe95-c85b-4dc9-a306-9701e77f1d66",
  "trigger_type_code": "TIMER",
  "trigger_status": "ACTIVE",
  "event_data": {
    "name": "Timer-download",
    "schedule": "1d",
    "schedule_type": "Rate"
  },
  "last_updated_time": "2020-06-04T10:33:30+08:00",
  "created_time": "2020-06-04T10:33:30+08:00"
}
```

5.2 示例 2：创建函数和 APIG 触发器获取静态页面

场景描述

本章节以Python 2.7为例，指导您通过API创建FunctionGraph函数和APIG触发器，并且通过APIG触发器调用API，获取静态页面。

API的调用方法请参见[如何调用API](#)。

前提条件

已在API网关服务中创建API分组，并记录API分组的ID和分配的子域名。

总体思路

创建FunctionGraph函数和APIG触发器，并且通过APIG触发器触发API获取静态页面，步骤如下：

1. **创建函数**：创建一个返回的静态页面函数。
2. **创建触发器**：创建APIG触发器。
3. 通过APIG触发器调用API，获取静态页面。

步骤 1：创建一个返回的静态页面函数

URI: POST /v2/{project_id}/fgs/functions

API文档详情请参见：[创建函数](#)

- 请求示例

```
POST https://{Endpoint}/v2/{project_id}/fgs/functions
{
  "code_filename": "index.zip",
  "code_type": "inline",
  "func_code": {
    "file":
      "UESDBAoAAAAIABY7vFD7lxPkAgMAALoHAAAIAAAAaW5kZXgucHndVdtu00AQfc9XrMKDExQ7zqW50
      VYqFZRKIFUQhFBVobU9iU1tr9mdbRKqSHwNH8aXMLtxrgoS8lSlog7c
      +bszNnjzRPmPnVZKKIkn440TtyBCVQcx6mMY2ATkaZiRjmWKMZzdNfzfcXgAXJkYsI4UzwrUoLpPMRE5B
      77KDQLCYigkGFMVescCxZMQQq0IDokcl4kruVYeiG15TU66VXqTxWGKvGiMUbwFhE1RGrXr0YVxsmXH
      CMTaBpNmVgQK1uEzdc8gwQpCKIYTFxKeYLU7HCUnBp8V80yMU7INTYsbKclqaKfjclMfBoDzR3FfKpxZl2
      bCMUfiXomWlz7GU5tNXw3oM5hBrBpfE9rdwZod22xzP
      +VeR8prxQZOvq9wqkezElTQxHqrkrCb1wjsBN32v5rPZa8zrLp/NPIG22fN8fbJsMRGTH/
      fHtO31X2kigaRveihxhjtV+j3Zfgq+t7IHU7XT6nV479NthP
      +r328Mh9Af9QSuAoBu01kU016qg2wq6J8A7bt8PTtxWC3pu0A8itzcMTjiHgEigZbNL67gkK4QkZ6n102cl
      8vVzWBX0upVKBBMW8zxKQdasXRpkXztMfWQbKOgckTI02CEoxcoZ2ESkzFiOvCeT6RSkU7FwlltVnmQvI
      XIFyWBbCzLR1vnaP+cO5unV24c2dZYB5Cgb9kdKh9rIMsN08mBwBCxjilzpbALN
      +WGCJ43CTMx6FjQ60uRQTOiLV9v7GfTtRzK9uLnF5xiAgzLhoOQKWxndEBuT1NaxTziU7K3vM6V2jnp6xM
      OZSAZ7Za6O6V7o82MI4kfhXZ+gFvS7YhmrOqaE6P41b5x8gJecDQ8E0TfgyvDauJC/
      iOyYBTu225059y73ckRm1zk1hvEhnhart6lZfqQ3pMb1NzY7eZvkP6c2Llk1CbqRomtb+UufH6j3Yu
      +CBpxqy99VcVenjYoKtnKVfv7fHXqTAmXMTaNQ6hDtH5iWkZslGDN7QbBiczfsmezMEDT2Q0bWP/
      TzgdD1n1BLAQleAwoAAAAIABY7vFD7lxPkAgMAALoHAAAIAAAAAAAAAAAAAAAAADzAgAAAABpbmRleC5
      weVBLBQYAAAAAAQABADYAAAAoAwAAAAA="
  },
  "func_name": "get_html",
  "handler": "index.handler",
  "memory_size": 256,
  "package": "default",
  "runtime": "Python2.7",
  "timeout": 5
}
```

- 响应示例

```
{
  "func_urn": "urn:fss:{project_name}:{project_id}:function:default:get_html:latest",
  "func_name": "get_html",
  "domain_id": "89fexxxd636",
  "namespace": "{project_id}",
  "project_name": "xxx",
  "package": "default",
  "runtime": "Python2.7",
  "timeout": 5,
  "handler": "index.handler",
  "memory_size": 256,
  "cpu": 400,
}
```

```
"code_type": "inline",
"code_filename": "index.zip",
"code_size": 884,
"digest":
"b08fef5e97dd130037978db07f0e9109aa43a191517cd1196bcab822f17dddcf37f7506a15691177962f98
03ba6d170a1c87aafb4fa1b9f0d07f9415642b26d2",
"version": "latest",
"image_name": "latest-200604105808@we0qo",
"last_modified": "2020-06-04T10:58:08+08:00",
"strategy_config": {
"concurrency": -1
},
"StrategyConfig": {},
"enterprise_project_id": "0"
}
```

记录函数的URN，即响应示例中的“func_urn”信息。

步骤 2：创建 APIG 触发器

URI: POST /v2/{project_id}/fgs/triggers/{function_urn}

API文档详情请参见：[创建触发器](#)

- 请求示例

POST https://{Endpoint}/v2/{project_id}/fgs/triggers/{function_urn}

```
{
"event_data": {
"group_id": "a9ad0d5df4d7475c9bc35a7c17d89304",
"env_id": "DEFAULT_ENVIRONMENT_RELEASE_ID",
"auth": "NONE",
"protocol": "HTTP",
"name": "API_GetHtml",
"path": "/test",
"match_mode": "SWA",
"req_method": "ANY",
"backend_type": "FUNCTION",
"sl_domain": "a9ad0d5df4d7475c9bc35a7c17d89304.apig.xxx.xxxapis.com",
"type": 1,
"env_name": "RELEASE"
},
"event_type_code": "APICreated",
"trigger_status": "ACTIVE",
"trigger_type_code": "APIG"
}
```

function_urn为[步骤1：创建一个返回的静态页面函数](#)中记录的函数URN，group_id为API分组的ID，sl_domain为APIG给API分组分配的子域名。

- 响应示例

```
{
"trigger_id": "1b3ec74b86454aa39001a9f89cc70ee2",
"trigger_type_code": "APIG",
"trigger_status": "ACTIVE",
"event_data": {
"api_id": "cbc698153d1f4265bdd8384b5cf6e581",
"api_name": "API_GetHtml",
"auth": "NONE",
"env_id": "",
"env_name": "",
"func_info": {
"function_urn": "urn:fss:{project_name}:{project_id}:function:default:get_html",
"invocation_type": "sync",
"timeout": 5000,
"version": "latest"
},
"group_id": "a9ad0d5df4d7475c9bc35a7c17d89304",
"group_name": "APIGroup_gethtml",
"invoke_url": "http://a9ad0d5df4d7475c9bc35a7c17d89304.apig.xxx.xxxapis.com/test",
}
```

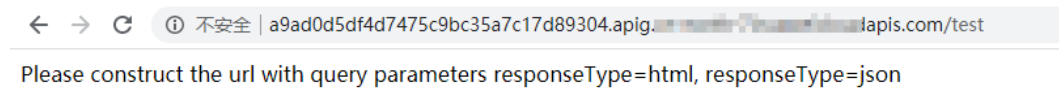
```
"match_mode": "SWA",
"name": "API_GetHtml",
"path": "/test",
"protocol": "HTTP",
"req_method": "ANY",
"triggerid": "1b3ec74b86454aa39001a9f89cc70ee2",
"type": 1
},
"last_updated_time": "2020-06-04T17:14:32+08:00",
"created_time": "2020-06-04T17:14:32+08:00"
}
```

记录“invoke_url”的值。

步骤 3: 通过 APIG 触发器调用 API，获取静态页面

在浏览器输入“invoke_url”的值，调用API，获取静态页面。

图 5-1 调用 API



5.3 示例 3: 上传代码到 OBS 桶，并使用 OBS 链接创建函数

场景描述

本章节以Python 2.7为例，指导您在本地编码，然后将代码文件上传到OBS桶，实现通过OBS链接创建函数。

API的调用方法请参见[如何调用API](#)。

前提条件

已在OBS服务中创建OBS桶。

约束与限制

请确保函数与OBS桶创建在同一区域下。

总体思路

在本地进行函数编码后，将代码文件上传到OBS桶，并使用OBS链接创建函数。步骤如下：

1. 创建函数工程：在本地进行编码实现函数功能。
2. 上传工程到OBS桶：将代码文件压缩为zip格式并上传到已创建的OBS桶中，记录代码文件的OBS链接。
3. [创建函数](#)：调用API通过OBS链接创建函数。

步骤 1: 创建函数工程

1. 编写打印helloworld的代码。

打开文本编辑器，编写helloworld函数，代码如下，文件命名为“helloworld.py”，保存文件。

```
def printhello():  
    print 'Hello world!'
```

2. 定义FunctionGraph函数。

打开文本编辑器，定义函数，代码如下，文件命名为“index.py”，保存文件（与“helloworld.py”保存在同一文件夹下）。

```
import json  
import helloworld  
  
def handler (event, context):  
    output =json.dumps(event)  
    helloworld.printhello()  
    return output
```

步骤 2：上传工程到 OBS 桶

1. 在函数工程中，选中“helloworld.py”和“index.py”文件，打包压缩为“fss_examples_python2.7.zip”。
2. 上传“fss_examples_python2.7.zip”到OBS桶，记录OBS链接。

步骤 3：调用创建函数 API，使用 OBS 链接创建函数

URI: POST /v2/{project_id}/fgs/functions

API文档详情请参见：[创建函数](#)

- 请求示例

```
POST https://{Endpoint}/v2/{project_id}/fgs/functions  
{  
  "code_type": "obs",  
  "code_url": "https://test.obs.example.example.com/fss_examples_python2.7.zip",  
  "func_name": "create_function_from_obs",  
  "handler": "index.handler",  
  "memory_size": 256,  
  "package": "default",  
  "runtime": "Python2.7",  
  "timeout": 30  
}
```

“code_url”为2中记录的OBS链接。

- 响应示例

```
{  
  "func_urn": "urn:fss:{project_name}:{project_id}:function:default:create_function_from_obs:latest",  
  "func_name": "create_function_from_obs",  
  "domain_id": "0503xxxa960",  
  "namespace": "{project_id}",  
  "project_name": "xxx",  
  "package": "default",  
  "runtime": "Python2.7",  
  "timeout": 30,  
  "handler": "index.handler",  
  "memory_size": 256,  
  "cpu": 400,  
  "code_type": "obs",  
  "code_url": "https://test.obs.example.example.com/fss_examples_python2.7.zip",  
  "code_filename": "fss_examples_python2.7.zip",  
  "code_size": 436,  
  "digest":  
  "3af770ada27514564b1a20d964cba4b35f432fa40f9fc4f4f7c1f0d2f42eac6cb4db1358c195235966b05f6  
  6b4664e7bf31c3f384a9066b3d1fcc3e96b4c3f65",  
  "version": "latest",  
}
```

```
"image_name": "latest-200619100734@gjf4p",
"last_modified": "2020-06-19T10:07:34+08:00",
"strategy_config": {
  "concurrency": -1
},
"StrategyConfig": {},
"enterprise_project_id": "0"
}
```

5.4 示例 4：函数互调

场景描述

本章节以python语言举例，指导您在一个函数里调用另外一个函数。

API的调用方法请参见[如何调用API](#)。

前提条件

已创建委托，包含“FunctionGraph Administrator”或“Tenant Administrator”权限的委托。

约束与限制

请确保子函数与父函数创建在同一区域下。

总体思路

1. 创建子函数。
2. 创建父函数。
3. 调用函数。

步骤一：创建子函数

在函数控制台创建子函数，本例所选参数如[图5-2](#)所示，其他参数保持默认。

图 5-2 配置参数

The screenshot shows a configuration form for a function. The 'Function Name' field is highlighted with a red box and contains 'SubFunction'. The 'Runtime' field is also highlighted with a red box and contains 'Python 2.7'. Other fields include 'Function Type' (Event Function), 'Region', 'Project', 'Enterprise Project' (default), 'Delegation' (None), and 'Runtime' (Python 2.7). There are also links for 'View Enterprise Project' and 'View Python Function Development Guide'.

步骤二：创建父函数

1. 在函数控制台创建父函数。委托名称选择包含“FunctionGraph Administrator”或“Tenant Administrator”权限的委托，本例所选参数如图5-3所示，可自行选择函数所在区域，其他参数保持默认。

图 5-3 配置参数

基本信息

函数类型 事件函数 HTTP函数
处理事件请求的函数。您可以通过云服务平台触发函数并执行。

区域
不同区域的资源之间内网不互通。请就近选择靠近您业务的区域，可以降低网络时延、提高访问速度。

项目

函数名称
可包含字母、数字、下划线和中划线，以大小写字母开头，以字母或数字结尾，长度不超过60个字符。

企业项目 [查看企业项目](#)
企业项目是一种云资源管理方式，企业项目管理服务提供统一的云资源按项目管理，以及项目内的资源管理、成员管理。

委托名称 [创建委托](#)
用户委托函数工作流去访问其他的云服务，举例：如果用户函数需要访问LTS、VPC等服务，则需要提供权限委托名称，如果用户函数不访问任何云服务，则不用提供委托名称。

委托权限策略 [查看策略](#)
展示您选择委托 fg-test 中的权限策略。您可以在 IAM 控制台增加或者删除策略。

运行时 [查看Python函数开发指南](#)
选择用来编写函数的语言，请注意，控制台代码编辑器仅支持Node.js、Python和PHP。

2. 在“代码”页签，复制以下示例代码到代码编辑区，替换字段后单击“部署代码”。字段说明请参考表5-1。

```
import json
import requests
def handler (event, context):
    token = context.getToken()
    project_id = context.getProjectID()
    urn = 'urn:fss:xx-xxxx-x:xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx:function:xxx:subfunction:latest' #子函数的urn，在子函数详情页拓扑图的右侧可查询
    url = 'https://functiongraph.{region_id}.{external_global_domain_name}/v2/%s/fgs/functions/%s/inocations' % (project_id,urn)
    headers = {
        "x-auth-token": token,
        "content-type": 'application/json'
    }
    body = {"k":"v"}
    resp = requests.post(url, json=body, headers=headers, verify=False)
    print(resp.text)
    return {
        "statusCode": 200,
        "isBase64Encoded": False,
        "body": json.dumps(event),
        "headers": {
            "Content-Type": "application/json"
        }
    }
```

表 5-1 示例代码字段获取说明

字段	说明
region_id	区域ID。 请参考 区域和终端节点 中“区域”列，根据函数所在区域填写。以“华东-上海一”为例，则填写“cn-east-3”。
external_global_domain_name	外部全局域名。 请填写“myhuaweicloud.com”。

步骤三：调用函数

1. 在父函数的函数详情页，单击“测试”，弹出“配置测试事件”弹窗，单击“创建”使用默认空白模板创建测试事件。
2. 创建成功后，单击代码编辑区的“测试”，进行父函数调用。
3. 查看执行结果。
 - 父函数的执行结果显示“执行成功”，返回“200”。

图 5-4 父函数执行历史

 执行成功

函数返回

```
{
  "body": "{\"key\": \"value\"}",
  "headers": {
    "Content-Type": "application/json"
  },
  "isBase64Encoded": false,
  "statusCode": 200
}
```

- 子函数的日志中，显示被调用的日志。

图 5-5 子函数执行历史

只显示错误日志

```
2022/02/25 14:48:07 GMT+08:00 Start invoke request 'd994086-951553', version: latest
2022/02/25 14:48:07 GMT+08:00 Finish invoke request 'd994086-951553', duration: 1.197ms, billing duration: 100ms, memory used: 58.629MB, rss memory used: 21.402MB.
```

6 扩展及遥测 API

6.1 扩展 API

6.1.1 概述

您可以使用扩展API将函数与监控、可观察性、安全性和监管等工具集成，控制并参与FunctionGraph生命周期的所有阶段。例如在函数调用之前、期间和之后捕获诊断信息；无需更改代码即可自动检测您的代码；在函数调用之前获取配置设置；通过安全代理检测功能发出警报；将遥测数据直接从函数异步发送到自定义目的地。

扩展作为独立进程在函数执行环境中运行。由于扩展作为进程运行，您可以使用不同于函数的语言来编写它们，建议您使用已编译的语言实现扩展。在这种情况下，扩展是一个独立的二进制文件，与支持的运行时兼容。如果使用非编译的语言，请确保和函数的运行时兼容。

说明

- 当前http和事件类型函数支持扩展的使用，自定义镜像不支持。
- 您可以通过将扩展制作为依赖包的方式添加到函数。

6.1.2 操作步骤

步骤1 创建函数，可参考开发指南中的各个语言函数创建流程，此处不再赘述。

步骤2 创建依赖包，将扩展依赖包按照依赖包的添加规则添加到要执行的函数。

步骤3 调用函数，则内部已经启动扩展进程。

----结束

6.1.3 制作扩展依赖包

准备一个zip包，zip包中用于扩展进程的编写，可以拉起一个扩展进程，zip包中限制如下：

zip包中必须含有extension命名的文件夹，extension的文件夹中必须包含以extension为前缀，.sh文件类型的脚本文件，用于后续扩展启动扩展进程，例如zip包的文件结构如下：

extension -

extensionDemo.sh

test.py

代码内容参考[extension.zip](#)。

其中*extensionDemo.sh* 中的语言所在路径参考下表。

表 6-1 多语言路径说明

语言	路径
Java8	/opt/function/runtime/java8/rtsp/jre/bin/java
Java11	/opt/function/runtime/java11/rtsp/jre/bin/java
Node.js6	/opt/function/runtime/nodejs6.10/rtsp/nodejs/bin/node
Node.js8	/opt/function/runtime/nodejs8.10/rtsp/nodejs/bin/node
Node.js10	/opt/function/runtime/nodejs10.16/rtsp/nodejs/bin/node
Node.js12	/opt/function/runtime/nodejs12.13/rtsp/nodejs/bin/node
Node.js14	/opt/function/runtime/nodejs14.18/rtsp/nodejs/bin/node
Python2.7	/opt/function/runtime/python2.7/rtsp/python/bin/python
Python3.6	/opt/function/runtime/python3.6/rtsp/python/bin/python3
Python3.9	/opt/function/runtime/python3.9/rtsp/python/bin/python3
PHP7.3	/opt/function/runtime/php7.3/rtsp/php/bin/php

6.1.4 扩展 API 参考

整个扩展进程由用户自己编写，扩展进程中必须完成以下工作才能成功启用扩展功能：

1. 启动一个httpClient，向FunctionGraph函数发送注册请求。
2. 启动一个httpServer，接收FunctionGraph函数发送过来的调用请求。

6.1.4.1 API 方法

注册

http://\${EXTENSION_API_ADDR}/extension/register

进程部署的扩展都需要在FunctionGraph注册成功后才能接收事件。您可以通过EXTENSION_API_ADDR 环境变量得到FunctionGraph平台开放的提供注册的扩展API地址，例如对于Python，EXTENSION API 的获取方式为os.getenv("EXTENSION_API_ADDR ")。

路径

- /extension/register

方法

- POST

请求标头

FunctionGraph-Extension-Name – 扩展名的完整文件名。必需：是。类型：字符串。

FunctionGraph-Extension-Address – 该扩展用于接收调用事件的服务端地址。必需：是。类型：字符串。例如：127.0.0.1:8081。

说明

- FunctionGraph需要使用扩展的完整文件名来验证扩展是否已完成引导启动，且扩展注册的请求头必须包含本扩展启动的httpServer的地址和端口。例如，请求头的格式为：

```
headers = {
  'Content-type': 'application/json',
  'FunctionGraph-Extension-Name': "extensionDemo.sh"
  'FunctionGraph-Extension-Address': "127.0.0.1:8011"
}
```
- 当前规定最多支持十个扩展，扩展启动的Server地址和端口限制为127.0.0.1:8081 ~ 127.0.0.1:8090，且每个扩展的端口不可重复。

主体参数

不做解析

响应代码

200 - 响应正文包含函数名称、函数版本和处理程序名称。

其他为错误请求

示例响应正文

```
{
  "functionName": "helloWorld",
  "functionVersion": "$LATEST",
  "errorInfo": {
    "errorType": "",
    "errorMessage": ""
  }
}
```

调用

http://\${FunctionGraph-Extension-Address}/extension/invoke

在注册完成后，创建的httpServer的可以监听此路径，用于接收FunctionGraph函数请求调用事件。

路径

- /extension/register

方法

- POST

主体参数

tracelId - 请求的requestID

invokedFunctionUrn - 请求调用的URN

响应代码

200 - 响应正文包含函数名称、函数版本和处理程序名称。

其他为错误请求

示例响应正文

```
{
  "tracelId": "3da1f2dc-3222-475e-9205-e2e6c6318895",
  "invokedFunctionUrn": "sn:cn:yrk:1111:function:0@test@hello:latest"
}
```

6.1.5 如何通过扩展完成 SkyWalking 的对接

通过以下对接SkyWalking的示例帮助您理解如何通过扩展API将函数与监控、可观察性、安全性和监管等工具集成。

说明

当用已编译的语言实现扩展时，在本地编译生成二进制文件；使用非编译的语言实现扩展时，需要把对接第三方平台需要的依赖一起打包上传，且需要函数的运行时兼容。

将skywalking agent直接嵌入到扩展代码中，即可通过扩展完成对函数请求的监控。

示例代码：[skywalking-go.zip](#)

```
func startSkywalkingAgent() {
    r, err := reporter.NewGRPCReporter("10.174.242.117:11800", reporter.WithCDS(100))
    if err != nil {
        log.Fatalf("new reporter error %v\n", err)
        return
    }
    defer r.Close()
    tracer, err := go2sky.NewTracer("testservice", go2sky.WithReporter(r))
    if err != nil {
        log.Fatalf("create tracer error %v\n", err)
        return
    }
    go2sky.SetGlobalTracer(tracer)
    sm, err := httpPlugin.NewServerMiddleware(go2sky.GetGlobalTracer())
    if err != nil {
        log.Fatalf("create server middleware error %v\n", err)
        return
    }
    http.HandleFunc(invokeURL, sm(RequestHandler{}).ServeHTTP)
    http.ListenAndServe(currentExtensionAddr, nil)
}
```

其中10.174.242.117:11800 用户可以根据自己部署的是sykwalking collector的地址即可，testservice 用户可以根据自己项目服务进行命名。

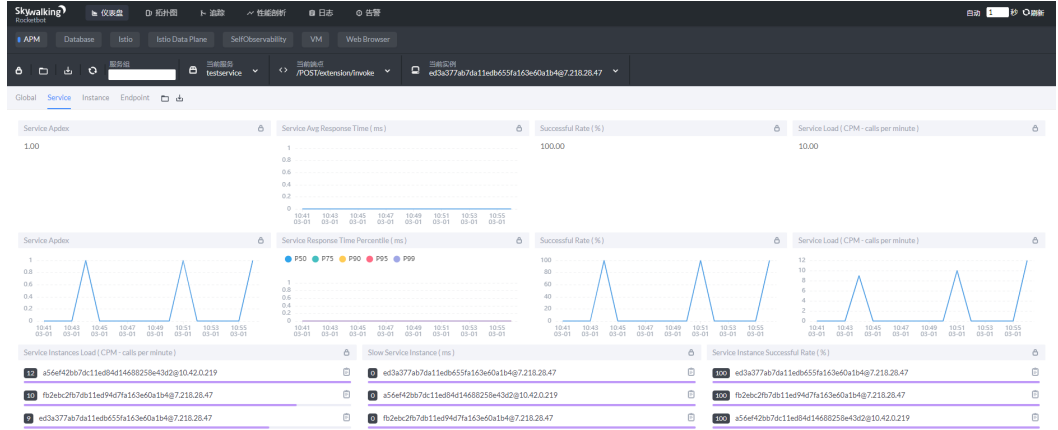
在本地编译生成二进制文件，将二进制文件和以extension为前缀的.sh文件类型的脚本文件打包上传依赖。扩展依赖包的文件结构如下：

extension -

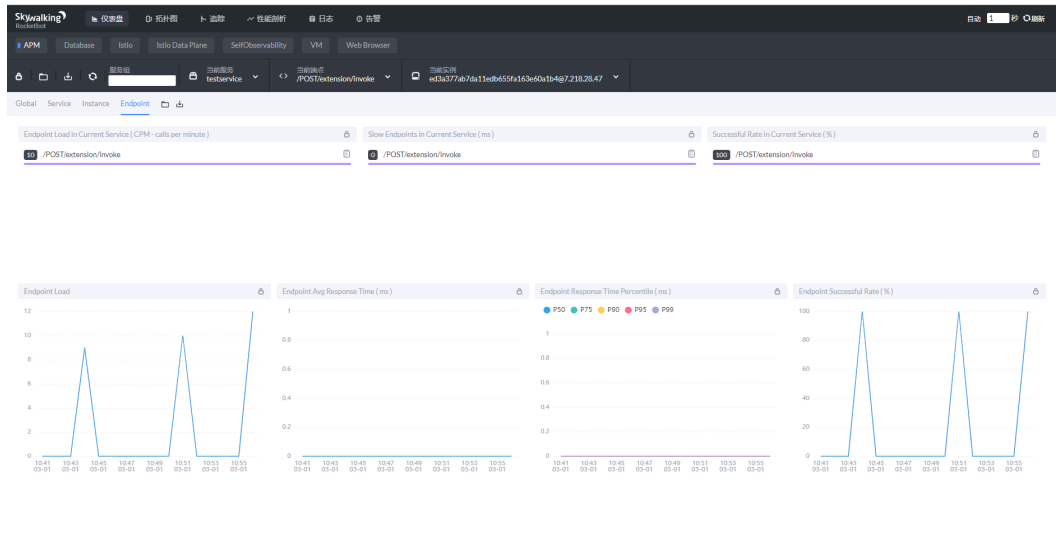
`extensionDemo.sh`

`Extension # 编译后的二进制文件`

成功对接后Service界面如下:



Endpoint界面如下:



6.2 遥测 API

6.2.1 概述

通过FunctionGraph的OpenTelemetry API, 扩展可以直接从FunctionGraph接收到函数执行期间初始化、调用期间的遥测数据, 也可以收集函数执行期间和扩展执行期间的用户打印日志。

在FunctionGraph执行时, 您可以从扩展向OpenTelemetry API发送订阅请求, 订阅后, FunctionGraph会自动将所有遥测数据发送到您的扩展, 您可以对数据进行处理, 例如存储到OBS、写入文件中或转发给其他可观测性工具。

扩展可以使用遥测 API 订阅三种不同的遥测流：

1. 平台遥测：日志、指标和跟踪，描述与执行环境运行时生命周期、扩展生命周期和函数调用相关的事件和错误。
2. 函数日志：FunctionGraph函数代码生成的自定义日志。
3. 扩展日志：FunctionGraph扩展代码生成的自定义日志。

6.2.1.1 如何在扩展中使用 OpenTelemetry API

遥测是一种特殊的扩展，扩展进程中必须完成以下工作才能成功启用遥测功能：

1. 在使用OpenTelemetry API之前，需要先注册拓展，具体内容可参考扩展API使用。
2. 在拓展启动一个httpClient，向FunctionGraph发送注册Telemetry API的请求。
3. 创建一个httpServer遥测侦听器，接收FunctionGraph发送的遥测数据。用户可以对此数据进行任何自定义处理。

📖 说明

最多只能创建一个遥测扩展，遥测的端口限制和扩展的端口限制一致，127.0.0.1:8081 ~ 127.0.0.1:8090。

遥测扩展示例：[Telemetry.zip](#)

6.2.1.2 注册 OpenTelemetry API

`http://${EXTENSION_API_ADDR}/telemetry`

您可以通过 `EXTENSION_API_ADDR` 环境变量得到平台开放的提供注册的API地址，例如对于Python，`EXTENSION_API_ADDR`的获取方式为 `os.getenv("EXTENSION_API_ADDR")`。

路径

- /telemetry

方法

- POST

主体参数

```
{
  "subscriber": {
    "protocol": "POST",
    "uri": "http://127.0.0.1:{PORT}"
  },
  "types": ["platform", "function", "extension"],
}
```

📖 说明

- 请求体中必须包括上述JSON字段。
- protocol为请求类型，限制为POST。
- uri为扩展进程中遥测侦听器URI，用于接收FunctionGraph发送的遥测数据，其中PORT可选8081~8090。
- Types为您希望扩展订阅的事件类型的信息，可选platform、function、extension。

响应代码

200 - 订阅请求成功

400 - 请求错误，并返回错误信息

500 - 服务错误

6.2.1.3 处理遥测

`http://localhost:{PORT}`，`localhost`是本地主机IP: `127.0.0.1`。

使用遥测 API 订阅后，扩展会自动开始通过遥测侦听器接收来自FunctionGraph的遥测数据。每个 POST 请求正文都包含 Event 对象数组。

订阅事件类型包括三大类：platform、function、extension。

类别	Event Type	描述
平台事件	platform	函数的执行情况
函数日志	function	来自函数代码的日志
扩展日志	extension	来自扩展代码的日志

platform类型事件包括以下几类：

类别	子类型	含义	数据结构	示例
开始调用事件	platform.initStart	表示函数初始化阶段已开始	Event: Object - name: String = platform.initStart -- 事件类型 - startTime: String -- 开始时间 - traceId: String -- 调用ID	<pre>{ "name": "platform.initStart", "startTime": "2023-02-01T00:03:50.000Z", "traceId": "69be64a7-0233-11e8-82e4-e5d37911193e" }</pre>
	platform.start	表示函数调用阶段已开始		
	platform.restoreStart	表示函数环境还原事件已启动		

类别	子类型	含义	数据结构	示例
调用完成事件	platform.initRuntimeDone,	表示函数初始化阶段已完成	Event: Object - name: String = platform.initRuntimeDone --事件类型	<pre>{ "name": "platform.initRuntimeDone", "startTime": "2023-02-01T00:03:50.000Z", "tracelId": "69be64a7-0233-11e8-82e4-e5d37911193e", "status": "success", }</pre>
	platform.RuntimeDone	表示函数调用阶段已完成	- startTime: String --开始时间	
	platform.restoreRuntimeDone	表示函数环境还原事件已完成	- tracelId: String --调用ID - status: String --调用状态	
调用报告	platform.initReport	包含函数初始化阶段的总体报告	Event: Object - name: String = platform.initReport platform.report platform.restoreReport --事件类型	<pre>{ "name": "platform.initReport", "startTime": "2023-02-01T00:03:50.000Z", "tracelId": "69be64a7-0233-11e8-82e4-e5d37911193e", "endTime": "2023-02-02T00:03:50.000Z", "status": "success", "reportMetrics": { "durationMs": 10.6, "billingDurationMS": 11, "memoryUsed": 23.4, "memorySize": 128, "podName": "pool22-300-128-fusion-64774779-4g77v", } }</pre>
	platform.report	包含函数调用阶段的总体报告	- startTime: String --开始时间	
	platform.restoreReport	包含函数还原事件的总体报告	- tracelId: String --调用ID - endTime: String --结束时间 - status: String --调用状态 - metrics: reportMetrics --调用数据 reportMetrics: Object - durationMs: float64 --调用时长 - billingDurationMS: float64 --计费时长 - memoryUsed: float64 --内存使用大小 - memorySize: int --内存大小 - podName: String --pod名	

Function类型事件:

类型	数据结构	示例
Function	Event: Object - name: String --事件类型 - startTime: String --开始时间 - logs: string --日志内容	对于需要采集的日志内容需要在日志中添加 [Function], 例如: [Function]this is a simple log. <pre>{ "name": "function", "startTime": "2023-02-01T00:03:50.000Z", "logs": "[Function]this is a simple log." }</pre>

Extension类型事件:

类型	数据结构	示例
Extension	Event: Object - name: String --事件类型 - startTime: String --开始时间 - logs: string --日志内容	对于需要采集的日志内容需要在日志中添加 [Extension], 例如: [Extension]this is a simple log. <pre>{ "name": "extension", "startTime": "2023-02-01T00:03:50.000Z", "logs": "[Extension]this is a simple log." }</pre>

7 API

7.1 函数调用

7.1.1 同步执行函数

功能介绍

同步调用指的是客户端请求需要明确等到响应结果，也就是说这样的请求必须得调用到用户的函数，并且等到调用完成才返回。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/fgs/functions/{function_urn}/invocations

表 7-1 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见 FunctionGraph函数模型的描述。

请求参数

表 7-2 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）。
X-Cff-Log-Type	否	String	是否返回函数执行日志。取值：tail（返回函数执行后的4K日志），或者为空（不返回日志）。
X-CFF-Request-Version	否	String	返回体格式。 v0：默认返回文本格式。 v1：默认返回json格式，sdk需要使用此值。
X-Cff-Instance-Memory	否	String	设置本次执行函数使用的内存规格。 取值：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096、8192、10240。

表 7-3 请求 Body 参数

参数	是否必选	参数类型	描述
{自定义key}	是	Object	执行函数请求体

响应参数

状态码： 200

表 7-4 响应 Header 参数

参数	参数类型	描述
X-Cff-Invoke-Summary	String	本次同步调用执行信息摘要。
X-Cff-Request-Id	String	本次同步调用请求id。

参数	参数类型	描述
X-Cff-Function-Log	String	本次同步调用用户日志，需在请求头中设置X-Cff-Log-Type:tail。截取日志的后2000个字节并且做base编码。
X-CFF-Billing-Duration	String	本次同步调用计费信息。
X-Cff-Response-Version	String	返回体格式。 v0：默认返回文本格式。 v1：默认返回json格式。
X-Func-Err-Code	String	本次同步调用错误码，若执行成功则为0。
X-Is-Func-Err	String	是否为用户函数错误。

表 7-5 响应 Body 参数

参数	参数类型	描述
request_id	String	请求ID
result	String	函数执行结果
log	String	函数执行返回日志
status	Integer	函数执行返回状态

状态码： 400

表 7-6 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-7 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-8 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-9 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-10 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 503

表 7-11 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

同步执行函数，请求参数为键值对“k”：“v”。

```
POST /v2/{project_id}/fgs/functions/{function_urn}/invocations
{
  "k": "v"
}
```

响应示例

状态码： 200

OK

```
{
  "result" : "{\"statusCode\": 200, \"isBase64Encoded\": false, \"body\": \"{{{\"key\": \"value\"}}}\",
  \"headers\": {\"Content-Type\": \"application/json\"}}",
  "log" : "2022-09-20T11:43:57Z Start invoke request '1cbe80f3-3c65-475e-ad88-76ac518d386a', version:
v1\nHello, World!\n\n2022-09-20T11:43:58Z Finish invoke request '1cbe80f3-3c65-475e-
ad88-76ac518d386a', duration: 65.828ms, billing duration: 66ms, memory used: 21.473MB, billing memory:
128MB",
  "status" : 200,
  "request_id" : "1cbe80f3-3c65-475e-ad88-76ac518d386x"
}
```

SDK 代码示例

SDK代码示例如下。

Java

同步执行函数，请求参数为键值对“k”：“v”。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.Map;
import java.util.HashMap;

public class InvokeFunctionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        InvokeFunctionRequest request = new InvokeFunctionRequest();
        Map<String, Object> listbodyInvokeFunctionRequestBody = new HashMap<>();
        listbodyInvokeFunctionRequestBody.put("k", "v");
        request.withBody(listbodyInvokeFunctionRequestBody);
        try {
            InvokeFunctionResponse response = client.invokeFunction(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        }
    }
}
```

```
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

同步执行函数，请求参数为键值对“k”：“v”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = InvokeFunctionRequest()
        listInvokeFunctionRequestBodybody = {
            "k": "v"
        }
        request.body = listInvokeFunctionRequestBodybody
        response = client.invoke_function(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

同步执行函数，请求参数为键值对“k”：“v”。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
    WithAk(ak).
    WithSk(sk).
    Build()

client := functiongraph.NewFunctionGraphClient(
    functiongraph.FunctionGraphClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.InvokeFunctionRequest{}
var listInvokeFunctionRequestBodybody = map[string]interface{}{
    "k": "v",
}
request.Body = listInvokeFunctionRequestBodybody
response, err := client.InvokeFunction(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。
503	Service Unavailable 服务不可用。

错误码

请参见[错误码](#)。

7.1.2 异步执行函数

功能介绍

异步执行函数。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/fgs/functions/{function_urn}/invocations-async

表 7-12 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见 FunctionGraph函数模型的描述。

请求参数

表 7-13 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

表 7-14 请求 Body 参数

参数	是否必选	参数类型	描述
{自定义key}	是	Object	异步执行函数请求体

响应参数

状态码： 202

表 7-15 响应 Body 参数

参数	参数类型	描述
request_id	String	请求ID。

状态码： 400

表 7-16 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-17 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-18 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-19 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-20 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 503

表 7-21 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

异步执行函数，请求参数为键值对“k”：“v”。

```
POST /v2/{project_id}/fgs/functions/{function_urn}/invocations-async
{
  "body": {
    "k": "v"
  }
}
```

响应示例

状态码： 202

Accepted

```
{
  "request_id": "1167bf8c-87b0-43ab-8f5f-26b16c64f252"
}
```

SDK 代码示例

SDK代码示例如下。

Java

异步执行函数，请求参数为键值对“k”：“v”。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```

```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.Map;
import java.util.HashMap;

public class AsyncInvokeFunctionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        AsyncInvokeFunctionRequest request = new AsyncInvokeFunctionRequest();
        Map<String, Object> listbodyAsyncInvokeFunctionRequestBody = new HashMap<>();
        listbodyAsyncInvokeFunctionRequestBody.put("body", "{\"k\":\"v\"}");
        request.withBody(listbodyAsyncInvokeFunctionRequestBody);
        try {
            AsyncInvokeFunctionResponse response = client.asyncInvokeFunction(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

异步执行函数，请求参数为键值对“k”：“v”。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
```

```
credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = AsyncInvokeFunctionRequest()
    listAsyncInvokeFunctionRequestBodybody = {
        "body": "{\"k\":\"v\"}"
    }
    request.body = listAsyncInvokeFunctionRequestBodybody
    response = client.async_invoke_function(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

异步执行函数，请求参数为键值对“k”：“v”。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.AsyncInvokeFunctionRequest{}
    var listAsyncInvokeFunctionRequestBodybody = map[string]interface{}{
        "body": "{\"k\":\"v\"}",
    }
    request.Body = listAsyncInvokeFunctionRequestBodybody
    response, err := client.AsyncInvokeFunction(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
202	Accepted
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。
503	Service Unavailable 服务不可用。

错误码

请参见[错误码](#)。

7.1.3 函数异步执行并返回预留实例 ID（已废弃）

功能介绍

函数异步执行并返回预留实例ID用于场景指客户端请求执行比较费时任务，不需要同步等待执行完成返回结果，该方法提前返回任务执行对应的预留实例ID, 如果预留实例有异常，可以通过该实例ID把对应实例删除（该接口主要针对白名单用户）。目前仅支持广州

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/fgs/functions/{function_urn}/reserved-invocations

表 7-22 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。

参数	是否必选	参数类型	描述
function_urn	是	String	函数的URN，详细解释见FunctionGraph函数模型的描述。

请求参数

表 7-23 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 7-24 请求 Body 参数

参数	是否必选	参数类型	描述
{自定义key}	是	Object	函数异步执行并返回预留实例请求body体。

响应参数

状态码： 202

表 7-25 响应 Header 参数

参数	参数类型	描述
Content-Type	String	application/json

表 7-26 响应 Body 参数

参数	参数类型	描述
instance_id	String	预留实例id

状态码： 404

表 7-27 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

函数异步执行并返回预留实例ID。

```
POST /v2/{project_id}/fgs/functions/{function_urn}/reserved-invocations
{
  "body": {
    "k": "v"
  }
}
```

响应示例

状态码： 202

Accepted

```
{
  "instance_id": "048bad9b-96ae-45ad-8f12-4061a744ef9d"
}
```

状态码： 404

Not Found

```
{
  "error_code": "FSS.1051",
  "error_msg": "Not found the function"
}
```

状态码

状态码	描述
202	Accepted
404	Not Found

错误码

请参见[错误码](#)。

7.2 函数配额

7.2.1 查询租户配额

功能介绍

查询租户配额

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/quotas

表 7-28 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。

请求参数

表 7-29 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 200

表 7-30 响应 Body 参数

参数	参数类型	描述
quotas	ListQuotasResult object	配额信息

表 7-31 ListQuotasResult

参数	参数类型	描述
resources	Array of Resources objects	配额列表

表 7-32 Resources

参数	参数类型	描述
quota	Integer	函数配额限制。
used	Integer	已使用的配额。
type	String	“资源类型” 枚举值： <ul style="list-style-type: none"> • fgs_func_scale_down_timeout: v1版本函数的实例闲置释放时间 • fgs_func_occurs: v1版本函数为实例数配额, v2版本函数为预留实例配额 • fgs_func_pat_idle_time: v1版本vpc函数的pat容器闲置释放时间 • fgs_func_num: 用户函数数量配额 • fgs_func_code_size: 用户函数总代码大小配额 • fgs_workflow_num: 用户函数流数量配额 • fgs_on_demand_instance_limit: v2版本函数单函数最大实例数配额 • fgs_func_qos_limit: 用户函数实例数配额
unit	String	资源的计数单位。fgs_func_code_size,单位为MB,其他场景无单位

状态码： 400

表 7-33 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-34 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-35 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-36 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询租户配额。

```
GET /v2/{project_id}/fgs/quotas
```

响应示例

状态码： 200

查询成功

```
{
  "quotas": {
    "resources": [ {
      "quota": 60,
      "used": 3,
      "type": "fgs_func_scale_down_timeout"
    }, {
      "quota": 100,
      "used": 22,
      "type": "fgs_func_occurs"
    }, {
      "quota": 100,
      "used": 22,
      "type": "fgs_func_pat_idle_time"
    }
  ]
}
```

```
    }, {  
      "quota" : 100,  
      "used" : 22,  
      "type" : "fgs_func_num"  
    }, {  
      "quota" : 10240,  
      "used" : 22,  
      "type" : "fgs_func_code_size",  
      "unit" : "MB"  
    }, {  
      "quota" : 512,  
      "used" : 22,  
      "type" : "fgs_workflow_num"  
    }  
  ]  
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;  
  
import com.huaweicloud.sdk.core.auth.ICredential;  
import com.huaweicloud.sdk.core.auth.BasicCredentials;  
import com.huaweicloud.sdk.core.exception.ConnectionException;  
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;  
import com.huaweicloud.sdk.core.exception.ServiceResponseException;  
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;  
import com.huaweicloud.sdk.functiongraph.v2.*;  
import com.huaweicloud.sdk.functiongraph.v2.model.*;  
  
public class ListQuotasSolution {  
  
    public static void main(String[] args) {  
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great  
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or  
        // environment variables and decrypted during use to ensure security.  
        // In this example, AK and SK are stored in environment variables for authentication. Before running  
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
        String ak = System.getenv("CLOUD_SDK_AK");  
        String sk = System.getenv("CLOUD_SDK_SK");  
  
        ICredential auth = new BasicCredentials()  
            .withAk(ak)  
            .withSk(sk);  
  
        FunctionGraphClient client = FunctionGraphClient.newBuilder()  
            .withCredential(auth)  
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))  
            .build();  
        ListQuotasRequest request = new ListQuotasRequest();  
        try {  
            ListQuotasResponse response = client.listQuotas(request);  
            System.out.println(response.toString());  
        } catch (ConnectionException e) {  
            e.printStackTrace();  
        } catch (RequestTimeoutException e) {  
            e.printStackTrace();  
        } catch (ServiceResponseException e) {  
            e.printStackTrace();  
            System.out.println(e.getHttpStatusCode());  
            System.out.println(e.getRequestId());  
            System.out.println(e.getErrorCode());  
            System.out.println(e.getErrorMsg());  
        }  
    }  
}
```

```
}  
}  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkfunctiongraph.v2 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = FunctionGraphClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListQuotasRequest()  
        response = client.list_quotas(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := functiongraph.NewFunctionGraphClient(  
        functiongraph.FunctionGraphClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).
```

```
Build()  
  
request := &model.ListQuotasRequest{}  
response, err := client.ListQuotas(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	查询成功
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.3 函数依赖包

7.3.1 获取依赖包列表

功能介绍

获取依赖包列表

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/dependencies

表 7-37 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。

表 7-38 Query 参数

参数	是否必选	参数类型	描述
marker	否	String	上一次查询依赖包的最后记录位置，默认为"0"。
maxitems	否	String	单次查询最大条数
ispublic	否	String	是否为公共依赖包
dependency_type	否	String	依赖包类型public: 公开,private:私有, all: 全部。缺省时查询全量
runtime	否	String	FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。
name	否	String	依赖包名称。
limit	否	String	本次查询可获取的依赖包的最大数目，默认为"400"。

请求参数

表 7-39 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 200

表 7-40 响应 Body 参数

参数	参数类型	描述
dependencies	Array of ListDependenciesResult objects	依赖包列表
next_marker	Long	下次读取位置
count	Long	依赖包总数

表 7-41 ListDependenciesResult

参数	参数类型	描述
id	String	依赖包ID
owner	String	依赖包拥有者，public标识为公共依赖包
link	String	依赖包在obs的存储地址

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> • Java8 • Java11 • Node.js6.10 • Node.js8.10 • Node.js10.16 • Node.js12.13 • Node.js14.18 • Node.js16.17 • Node.js18.15 • Python2.7 • Python3.6 • Python3.10 • Go1.x • C#(.NET Core 2.1) • C#(.NET Core 3.1) • Custom • PHP7.3 • Python3.9 • http • Custom Image • Cangjie1.0
etag	String	依赖包唯一标志
size	Long	依赖包大小

参数	参数类型	描述
name	String	依赖包名
file_name	String	依赖包文件名
description	String	依赖包描述。
version	Long	依赖包版本号
last_modified	Long	依赖包更新时间

状态码： 401

表 7-42 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-43 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-44 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-45 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

当前租户依赖包列表。

```
GET https://{Endpoint}/v2/{project_id}/fgs/ListDependencies?marker=0&maxitems=400
```

响应示例

状态码： 200

ok

```
{
  "dependencies": [ {
    "id": "6578de50-9a18-48fa-b84e-b2d36636e433",
    "owner": "public",
    "link": "https://{obs_endpoint}/functionstage-examples/fg_test_pandas_1.1.3.zip",
    "runtime": "Python3.6",
    "etag": "35a893d9495b3e22259b2773996bc94d-6",
    "size": 31009808,
    "name": "panda1.1.3"
  } ],
  "next_marker": 43,
  "count": 43
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListDependenciesSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```

```
ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
    .withCredential(auth)
    .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
    .build();
ListDependenciesRequest request = new ListDependenciesRequest();
request.withMarker("<marker>");
request.withMaxitems("<maxitems>");
request.withIspublic("<ispublic>");
request.withDependencyType("<dependency_type>");
request.withRuntime("<runtime>");
request.withName("<name>");
request.withLimit("<limit>");
try {
    ListDependenciesResponse response = client.listDependencies(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListDependenciesRequest()
        request.marker = "<marker>"
        request.maxitems = "<maxitems>"
        request.ispublic = "<ispublic>"
        request.dependency_type = "<dependency_type>"
        request.runtime = "<runtime>"
        request.name = "<name>"
        request.limit = "<limit>"
        response = client.list_dependencies(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDependenciesRequest{}
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    maxitemsRequest := "<maxitems>"
    request.Maxitems = &maxitemsRequest
    ispublicRequest := "<ispublic>"
    request.Ispublic = &ispublicRequest
    dependencyTypeRequest := "<dependency_type>"
    request.DependencyType = &dependencyTypeRequest
    runtimeRequest := "<runtime>"
    request.Runtime = &runtimeRequest
    nameRequest := "<name>"
    request.Name = &nameRequest
    limitRequest := "<limit>"
    request.Limit = &limitRequest
    response, err := client.ListDependencies(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	ok
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.3.2 创建依赖包（待废弃）

功能介绍

创建依赖包

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/fgs/dependencies

表 7-46 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。

请求参数

表 7-47 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 7-48 请求 Body 参数

参数	是否必选	参数类型	描述
depend_file	否	String	depend_type为zip类型时必填，为文件流格式，需要base64编码zip文件。上传的文件大小限制为40M，如超过40M，请通过OBS上传。
depend_link	否	String	depend_type为obs类型时，依赖包在obs的存储地址。
depend_type	是	String	导入类型，目前支持obs和zip。

参数	是否必选	参数类型	描述
runtime	是	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none">• http• Custom Image• Cangjie1.0
name	是	String	依赖包名称。必须以大、小写字母开头，以字母或数字结尾，只能由字母、数字、下划线、点和中划线组成，长度不超过96个字符。
description	否	String	依赖包描述，不超过512个字符。

响应参数

状态码： 200

表 7-49 响应 Body 参数

参数	参数类型	描述
id	String	依赖包版本ID。
owner	String	依赖包拥有者。
link	String	依赖包在obs的存储地址。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0
etag	String	依赖包唯一标志。
size	Long	依赖包大小。

参数	参数类型	描述
name	String	依赖包名。
description	String	依赖包描述。
file_name	String	依赖包文件名。

状态码： 400

表 7-50 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-51 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-52 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-53 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-54 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 503

表 7-55 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

- 创建依赖包使用zip方式。

POST https://{Endpoint}/v2/{project_id}/fgs/dependencies

```
{
  "depend_file": "xxx",
  "depend_type": "zip",
  "runtime": "Node.js8.10",
  "name": "xxx"
}
```

- 创建依赖包使用obs方式。

POST https://{Endpoint}/v2/{project_id}/fgs/dependencies

```
{
  "depend_link": "https://{bucket}.{obs_endpoint}/depends/****/4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3.zip",
  "depend_type": "obs",
  "runtime": "Node.js8.10",
  "name": "xxx"
}
```

响应示例

状态码： 200

ok

```
{
  "id": "4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3",
  "owner": "*****",
  "link": "https://{bucket}.{obs_endpoint}/depends/****/4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3.zip",
  "runtime": "Python3.6",
  "etag": "83863be4b6c3a86aef995dbc83aae68f",
}
```

```
"size" : 577118,  
"name" : "python-kafka",  
"description" : "python支持kafka操作的库",  
"file_name" : "python-kafka.zip"  
}
```

状态码

状态码	描述
200	ok
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。
503	Service Unavailable 服务不可用

错误码

请参见[错误码](#)。

7.3.3 创建依赖包版本

功能介绍

创建依赖包版本

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/fgs/dependencies/version

表 7-56 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。

请求参数

表 7-57 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

表 7-58 请求 Body 参数

参数	是否必选	参数类型	描述
depend_file	否	String	depend_type为zip类型时必填，为文件流格式，需要base64编码zip文件。上传的文件大小限制为40M，如超过40M，请通过OBS上传。
depend_link	否	String	depend_type为obs类型时，依赖包在obs的存储地址。
depend_type	是	String	导入类型，目前支持obs和zip。

参数	是否必选	参数类型	描述
runtime	是	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> • http • Custom Image • Cangjie1.0
name	是	String	依赖包名称。必须以大、小写字母开头，以字母或数字结尾，只能由字母、数字、下划线、点和中划线组成，长度不超过96个字符。
description	否	String	依赖包描述，不超过512个字符。

响应参数

状态码： 200

表 7-59 响应 Body 参数

参数	参数类型	描述
id	String	依赖包版本ID。
owner	String	依赖包拥有者。
link	String	依赖包在obs的存储地址。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> • Java8 • Java11 • Node.js6.10 • Node.js8.10 • Node.js10.16 • Node.js12.13 • Node.js14.18 • Node.js16.17 • Node.js18.15 • Python2.7 • Python3.6 • Python3.10 • Go1.x • C#(.NET Core 2.1) • C#(.NET Core 3.1) • Custom • PHP7.3 • Python3.9 • http • Custom Image • Cangjie1.0
etag	String	依赖包唯一标志。
size	Long	依赖包大小。

参数	参数类型	描述
name	String	依赖包名。
description	String	依赖包描述。
file_name	String	依赖包文件名。
version	Long	依赖包版本号
dep_id	String	依赖包ID
last_modified	Long	依赖包更新时间

状态码： 401

表 7-60 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-61 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-62 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-63 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

- 创建依赖包版本使用zip方式。
POST https://{Endpoint}/v2/{project_id}/fgs/dependencies/version

```
{
  "depend_file": "xxx",
  "depend_type": "zip",
  "runtime": "Node.js8.10",
  "name": "xxx"
}
```
- 创建依赖包使用obs方式。
POST https://{Endpoint}/v2/{project_id}/fgs/dependencies/version

```
{
  "depend_link": "https://{bucket}.{obs_endpoint}/depends/****/4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3.zip",
  "depend_type": "obs",
  "runtime": "Node.js8.10",
  "name": "xxx"
}
```

响应示例

状态码： 200

ok

```
{
  "id": "4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3",
  "owner": "*****",
  "link": "https://{bucket}.{obs_endpoint}/depends/****/4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3.zip",
  "runtime": "Python3.6",
  "etag": "83863be4b6c3a86aef995dbc83aae68f",
  "size": 577118,
  "name": "python-kafka",
  "description": "python支持kafka操作的库",
  "file_name": "python-kafka.zip",
  "version": 0,
  "dep_id": "edbd67fa-f107-40b3-af75-a85f0577ad61",
  "last_modified": 1660029887
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 创建依赖包版本使用zip方式。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateDependencyVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateDependencyVersionRequest request = new CreateDependencyVersionRequest();
        CreateDependencyRequestBody body = new CreateDependencyRequestBody();
        body.withName("xxx");
        body.withRuntime(CreateDependencyRequestBody.RuntimeEnum.fromValue("Node.js8.10"));
        body.withDependType("zip");
        body.withDependFile("xxx");
        request.withBody(body);
        try {
            CreateDependencyVersionResponse response = client.createDependencyVersion(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

- 创建依赖包使用obs方式。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateDependencyVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
    .withCredential(auth)
    .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
    .build();

CreateDependencyVersionRequest request = new CreateDependencyVersionRequest();
CreateDependencyRequestBody body = new CreateDependencyRequestBody();
body.setName("xxx");
body.withRuntime(CreateDependencyRequestBody.RuntimeEnum.fromValue("Node.js8.10"));
body.withDependType("obs");
body.withDependLink("https://{bucket}-{obs_endpoint}/depends/****/4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3.zip");
request.withBody(body);
try {
    CreateDependencyVersionResponse response = client.createDependencyVersion(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- 创建依赖包版本使用zip方式。

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *
```

```
if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
```

```
    credentials = BasicCredentials(ak, sk) \
```

```
    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
    try:
```

```
request = CreateDependencyVersionRequest()
request.body = CreateDependencyRequestBody(
    name="xxx",
    runtime="Node.js8.10",
    depend_type="zip",
    depend_file="xxx"
)
response = client.create_dependency_version(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 创建依赖包使用obs方式。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateDependencyVersionRequest()
        request.body = CreateDependencyRequestBody(
            name="xxx",
            runtime="Node.js8.10",
            depend_type="obs",
            depend_link="https://{bucket}.{obs_endpoint}/depends/****/4f4ae4eb-dcdc-4dd3-
            bffd-79600bd972b3.zip"
        )
        response = client.create_dependency_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 创建依赖包版本使用zip方式。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateDependencyVersionRequest{}
    dependFileCreateDependencyRequestBody := "xxx"
    request.Body = &model.CreateDependencyRequestBody{
        Name: "xxx",
        Runtime: model.GetCreateDependencyRequestBodyRuntimeEnum().NODE_JS8_10,
        DependType: "zip",
        DependFile: &dependFileCreateDependencyRequestBody,
    }
    response, err := client.CreateDependencyVersion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

- 创建依赖包使用obs方式。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```

```

request := &model.CreateDependencyVersionRequest{}
dependLinkCreateDependencyRequestBody:= "https://{bucket}.{obs_endpoint}/depends/****/
4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3.zip"
request.Body = &model.CreateDependencyRequestBody{
    Name: "xxx",
    Runtime: model.GetCreateDependencyRequestBodyRuntimeEnum().NODE_JS8_10,
    DependType: "obs",
    DependLink: &dependLinkCreateDependencyRequestBody,
}
response, err := client.CreateDependencyVersion(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
}

```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	ok
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.3.4 获取依赖包版本列表

功能介绍

获取依赖包版本列表

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/dependencies/{depend_id}/version

表 7-64 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
depend_id	是	String	依赖包的ID。

表 7-65 Query 参数

参数	是否必选	参数类型	描述
marker	否	String	上一次查询依赖包的最后记录位置，默认为"0"。
maxitems	否	String	单次查询最大条数

请求参数

表 7-66 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 200

表 7-67 响应 Body 参数

参数	参数类型	描述
dependencies	Array of ListDependencyVersionsResult objects	依赖包列表
next_marker	Long	下次读取位置
count	Long	依赖包总数

表 7-68 ListDependencyVersionsResult

参数	参数类型	描述
id	String	依赖包版本ID
owner	String	依赖包拥有者，public标识为公共依赖包
link	String	依赖包在obs的存储地址

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0
etag	String	依赖包唯一标志 (MD5校验值)
size	Long	依赖包大小

参数	参数类型	描述
name	String	依赖包名
file_name	String	依赖包文件名
description	String	依赖包描述。
version	Long	依赖包版本号
last_modified	Long	依赖包更新时间
dep_id	String	依赖包ID
is_shared	Boolean	是否共享（已废弃）

状态码： 401

表 7-69 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-70 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-71 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-72 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

当前租户依赖包列表。

```
GET /v2/{project_id}/fgs/dependencies/{depend_id}/version?marker=0&maxitems=400
```

响应示例

状态码： 200

ok

```
{
  "dependencies": [ {
    "id": "6578de50-9a18-48fa-b84e-b2d36636e433",
    "owner": "public",
    "link": "https://{obs_endpoint}/functionstage-examples/fg_test_pandas_1.1.3.zip",
    "runtime": "Python3.6",
    "etag": "35a893d9495b3e22259b2773996bc94d-6",
    "size": 31009808,
    "name": "panda1.1.3"
  } ],
  "next_marker": 43,
  "count": 43
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListDependencyVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
    }
}
```

```
ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
    .withCredential(auth)
    .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
    .build();
ListDependencyVersionRequest request = new ListDependencyVersionRequest();
request.withMarker("<marker>");
request.withMaxitems("<maxitems>");
try {
    ListDependencyVersionResponse response = client.listDependencyVersion(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListDependencyVersionRequest()
        request.marker = "<marker>"
        request.maxitems = "<maxitems>"
        response = client.list_dependency_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListDependencyVersionRequest{}
    markerRequest := "<marker>"
    request.Marker = &markerRequest
    maxItemsRequest := "<maxitems>"
    request.Maxitems = &maxItemsRequest
    response, err := client.ListDependencyVersion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	ok
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.3.5 获取依赖包版本详情

功能介绍

获取依赖包版本详情

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/dependencies/{depend_id}/version/{version}

表 7-73 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
depend_id	是	String	依赖包的ID。
version	是	String	依赖包版本号。

请求参数

表 7-74 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 7-75 响应 Body 参数

参数	参数类型	描述
id	String	依赖包版本ID。

参数	参数类型	描述
owner	String	依赖包拥有者。
link	String	依赖包在obs的存储地址。
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0

参数	参数类型	描述
etag	String	依赖包唯一标志（MD5校验值）。
size	Long	依赖包大小。
name	String	依赖包名。
description	String	依赖包描述。
file_name	String	依赖包文件名。
version	Long	依赖包版本号
last_modified	Long	依赖包更新时间
dep_id	String	依赖包ID
download_link	String	依赖包文件临时下载链接
is_shared	Boolean	是否共享（已废弃）

状态码： 401

表 7-76 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-77 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-78 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误信息

状态码： 500

表 7-79 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

当前租户依赖包列表。

```
GET https://{Endpoint}/v2/{project_id}/fgs/dependencies/{depend_id}/version/{version}
```

响应示例

状态码： 200

OK

```
{
  "id": "4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3",
  "owner": "*****",
  "link": "https://{bucket}.{obs_endpoint}/depends/****/4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3.zip",
  "runtime": "Python3.6",
  "etag": "83863be4b6c3a86aef995dbc83aae68f",
  "size": 577118,
  "name": "python-kafka",
  "description": "python支持kafka操作的库",
  "file_name": "python-kafka.zip",
  "version": 0,
  "last_modified": 1660029887
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;
```

```
public class ShowDependencyVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowDependencyVersionRequest request = new ShowDependencyVersionRequest();
        try {
            ShowDependencyVersionResponse response = client.showDependencyVersion(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowDependencyVersionRequest()
        response = client.show_dependency_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowDependencyVersionRequest{}
    response, err := client.ShowDependencyVersion(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.3.6 删除依赖包版本

功能介绍

删除依赖包版本

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/fgs/dependencies/{depend_id}/version/{version}

表 7-80 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
depend_id	是	String	依赖包的ID。
version	是	String	依赖包版本号。

请求参数

表 7-81 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 400

表 7-82 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-83 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-84 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-85 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-86 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

删除依赖包版本。

```
DELETE https://{Endpoint}/v2/{project_id}/fgs/dependencies/{depend_id}/version/{version}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class DeleteDependencyVersionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteDependencyVersionRequest request = new DeleteDependencyVersionRequest();
        try {
            DeleteDependencyVersionResponse response = client.deleteDependencyVersion(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteDependencyVersionRequest()
        response = client.delete_dependency_version(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteDependencyVersionRequest{}
    response, err := client.DeleteDependencyVersion(request)
```



```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	No Content
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.3.7 删除指定的依赖包（待废弃）

功能介绍

删除指定的依赖包

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/fgs/dependencies/{depend_id}

表 7-87 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
depend_id	是	String	依赖包的ID。

请求参数

表 7-88 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 400

表 7-89 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-90 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-91 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-92 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-93 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

删除指定依赖包。

```
DELETE https://{Endpoint}/v2/{project_id}/fgs/dependencies/{depend_id}
```

响应示例

无

状态码

状态码	描述
204	No Content
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.3.8 获取指定依赖包（待废弃）

功能介绍

获取指定依赖包

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/dependencies/{depend_id}

表 7-94 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
depend_id	是	String	依赖包的ID。

请求参数

表 7-95 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

响应参数

状态码： 200

表 7-96 响应 Body 参数

参数	参数类型	描述
id	String	依赖包ID。
owner	String	依赖包拥有者。
link	String	依赖包在obs的存储地址。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0
etag	String	依赖包唯一标志。
size	Long	依赖包大小。

参数	参数类型	描述
name	String	依赖包名。
description	String	依赖包描述。
file_name	String	依赖包文件名。

状态码： 401

表 7-97 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-98 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-99 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-100 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询指定的依赖包信息。

```
GET https://{Endpoint}/v2/{project_id}/fgs/dependencies/{depend_id}
```

响应示例

状态码： 200

OK

```
{
  "id": "4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3",
  "owner": "*****",
  "link": "https://{bucket}.{obs_endpoint}/depends/****/4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3.zip",
  "runtime": "Python3.6",
  "etag": "83863be4b6c3a86aef995dbc83aae68f",
  "size": 577118,
  "name": "python-kafka",
  "description": "python支持kafka操作的库",
  "file_name": "python-kafka.zip"
}
```

状态码

状态码	描述
200	OK
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.3.9 更新指定依赖包

功能介绍

更新指定依赖包

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/fgs/dependencies/{depend_id}

表 7-101 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
depend_id	是	String	依赖包的ID。

请求参数

表 7-102 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。

表 7-103 请求 Body 参数

参数	是否必选	参数类型	描述
depend_file	否	String	depend_type为zip类型时必填，为文件流格式，需要base64编码zip文件。上传的文件大小限制为40M，如超过40M，请通过OBS上传。
depend_link	否	String	depend_type为obs类型时，依赖包在obs的存储地址。
depend_type	是	String	导入类型，目前支持obs和zip。

参数	是否必选	参数类型	描述
runtime	是	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none">• http• Custom Image• Cangjie1.0
name	是	String	依赖包名称。必须以大、小写字母开头，以字母或数字结尾，只能由字母、数字、下划线、点和中划线组成，长度不超过96个字符。
description	否	String	依赖包描述，不超过512个字符。

响应参数

状态码： 200

表 7-104 响应 Body 参数

参数	参数类型	描述
id	String	依赖包ID。
owner	String	依赖包拥有者。
link	String	依赖包在obs的存储地址。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> • Java8 • Java11 • Node.js6.10 • Node.js8.10 • Node.js10.16 • Node.js12.13 • Node.js14.18 • Node.js16.17 • Node.js18.15 • Python2.7 • Python3.6 • Python3.10 • Go1.x • C#(.NET Core 2.1) • C#(.NET Core 3.1) • Custom • PHP7.3 • Python3.9 • http • Custom Image • Cangjie1.0
etag	String	依赖包唯一标志。
size	Long	依赖包大小。

参数	参数类型	描述
name	String	依赖包名。
description	String	依赖包描述。
file_name	String	依赖包文件名。

状态码： 400

表 7-105 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-106 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-107 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-108 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-109 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

更新指定的依赖包。

```
PUT https://{Endpoint}/v2/{project_id}/fgs/dependencies/{depend_id}
{
  "depend_file": "xxx",
  "depend_type": "zip",
  "runtime": "Node.js8.10",
  "name": "xxx"
}
```

响应示例

状态码： 200

ok

- 示例 1

null

- 示例 2

```
{
  "id": "4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3",
  "owner": "*****",
  "link": "https://{bukcet}.{obs_endpoint}/depends/****/4f4ae4eb-dcdc-4dd3-bffd-79600bd972b3.zip",
  "runtime": "Python3.6",
  "etag": "83863be4b6c3a86aef995dbc83aae68f",
  "size": 577118,
  "name": "python-kafka",
  "description": "python支持kafka操作的库",
  "file_name": "python-kafka.zip"
}
```

状态码

状态码	描述
200	ok
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。

状态码	描述
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.4 函数测试事件

7.4.1 获取指定函数的测试事件列表

功能介绍

获取指定函数的测试事件列表

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/functions/{function_urn}/events

表 7-110 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见 FunctionGraph函数模型的描述。

请求参数

表 7-111 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 200

表 7-112 响应 Body 参数

参数	参数类型	描述
count	Integer	测试事件总数。
events	Array of ListEventsResult objects	测试事件列表。
next_marker	Long	下次读取位置。

表 7-113 ListEventsResult

参数	参数类型	描述
id	String	测试事件ID。
last_modified	Number	上次修改的时间。
name	String	测试事件名称。

状态码： 400

表 7-114 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-115 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-116 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-117 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-118 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

查询测试事件列表。

```
GET https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/events
```


响应示例

状态码： 200

OK

```
{
  "events": [ {
    "id": "3b659dc0-12fc-40dc-aa05-a321d9424cb3",
    "name": "event-k9r3",
    "last_modified": 1597374286
  } ],
  "next_marker": 1,
  "count": 1
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListEventsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        ListEventsRequest request = new ListEventsRequest();
        try {
            ListEventsResponse response = client.listEvents(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

```
}  
}  
}
```

Python

```
# coding: utf-8  
  
from huaweicloudsdkcore.auth.credentials import BasicCredentials  
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion  
from huaweicloudsdkcore.exceptions import exceptions  
from huaweicloudsdkfunctiongraph.v2 import *  
  
if __name__ == "__main__":  
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    # In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak = __import__('os').getenv("CLOUD_SDK_AK")  
    sk = __import__('os').getenv("CLOUD_SDK_SK")  
  
    credentials = BasicCredentials(ak, sk) \  
  
    client = FunctionGraphClient.new_builder() \  
        .with_credentials(credentials) \  
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \  
        .build()  
  
    try:  
        request = ListEventsRequest()  
        response = client.list_events(request)  
        print(response)  
    except exceptions.ClientRequestException as e:  
        print(e.status_code)  
        print(e.request_id)  
        print(e.error_code)  
        print(e.error_msg)
```

Go

```
package main  
  
import (  
    "fmt"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"  
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"  
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"  
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"  
)  
  
func main() {  
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security  
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment  
    variables and decrypted during use to ensure security.  
    // In this example, AK and SK are stored in environment variables for authentication. Before running this  
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment  
    ak := os.Getenv("CLOUD_SDK_AK")  
    sk := os.Getenv("CLOUD_SDK_SK")  
  
    auth := basic.NewCredentialsBuilder().  
        WithAk(ak).  
        WithSk(sk).  
        Build()  
  
    client := functiongraph.NewFunctionGraphClient(  
        functiongraph.FunctionGraphClientBuilder().  
            WithRegion(region.ValueOf("<YOUR REGION>")).  
            WithCredential(auth).
```

```
Build()  
  
request := &model.ListEventsRequest{}  
response, err := client.ListEvents(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.4.2 创建测试事件

功能介绍

创建测试事件

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/fgs/functions/{function_urn}/events

表 7-119 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见FunctionGraph函数模型的描述。

请求参数

表 7-120 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

表 7-121 请求 Body 参数

参数	是否必选	参数类型	描述
name	是	String	测试事件名称。只能由字母、数字、中划线和下划线组成，且必须以大写或小写字母开头
content	是	String	测试事件content，为json字符串的base64编码

响应参数

状态码： 200

表 7-122 响应 Body 参数

参数	参数类型	描述
id	String	测试事件ID。
name	String	测试事件名称。

状态码： 400

表 7-123 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-124 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-125 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-126 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-127 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

创建名为event-xx，内容为{"k":"v"}的测试事件。

```
POST https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/events
{
  "name" : "event-xx",
  "content" : "eyJrljoidi9"
}
```

响应示例

状态码： 200

OK

```
{
  "id" : "3b659dc0-12fc-40dc-aa05-a321d9424cb3",
  "name" : "event-k9r3"
}
```

SDK 代码示例

SDK代码示例如下。

Java

创建名为event-xx，内容为{"k":"v"}的测试事件。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateEventSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateEventRequest request = new CreateEventRequest();
        CreateEventRequestBody body = new CreateEventRequestBody();
        body.withContent("eyJrljoidi9");
        body.withName("event-xx");
        request.withBody(body);
    }
}
```

```
try {
    CreateEventResponse response = client.createEvent(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

创建名为event-xx，内容为{"k":"v"}的测试事件。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateEventRequest()
        request.body = CreateEventRequestBody(
            content="eyJrIjoiaV",
            name="event-xx"
        )
        response = client.create_event(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

创建名为event-xx，内容为{"k":"v"}的测试事件。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateEventRequest{}
    request.Body = &model.CreateEventRequestBody{
        Content: "eyJrIjoidiJ9",
        Name: "event-xx",
    }
    response, err := client.CreateEvent(request)
    if err == nil {
        fmt.Printf("%v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.4.3 删除指定测试事件

功能介绍

删除指定测试事件

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/fgs/functions/{function_urn}/events/{event_id}

表 7-128 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见FunctionGraph函数模型的描述。
event_id	是	String	测试事件ID

请求参数

表 7-129 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 400

表 7-130 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误信息

状态码： 401

表 7-131 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-132 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-133 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-134 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

删除指定测试事件。

```
DELETE https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/events/{event_id}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class DeleteEventSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteEventRequest request = new DeleteEventRequest();
        try {
            DeleteEventResponse response = client.deleteEvent(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteEventRequest()
        response = client.delete_event(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteEventRequest{}
    response, err := client.DeleteEvent(request)
```

```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	No Content
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.4.4 获取测试事件详细信息

功能介绍

获取测试事件详细信息

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/functions/{function_urn}/events/{event_id}

表 7-135 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。

参数	是否必选	参数类型	描述
function_urn	是	String	函数的URN，详细解释见FunctionGraph函数模型的描述。
event_id	是	String	测试事件ID

请求参数

表 7-136 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 200

表 7-137 响应 Body 参数

参数	参数类型	描述
id	String	测试事件ID。
name	String	测试事件名称。
content	String	测试事件content。
last_modified	Number	上次修改的时间。

状态码： 400

表 7-138 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-139 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-140 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-141 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-142 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

获取指定测试事件。

```
GET https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/events/{event_id}
```


Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowEventRequest()
        response = client.show_event(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowEventRequest{}
    response, err := client.ShowEvent(request)
```

```
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.4.5 更新测试事件详细信息

功能介绍

更新测试事件详细信息

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/fgs/functions/{function_urn}/events/{event_id}

表 7-143 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。

参数	是否必选	参数类型	描述
function_urn	是	String	函数的URN，详细解释见FunctionGraph函数模型的描述。
event_id	是	String	测试事件ID

请求参数

表 7-144 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

表 7-145 请求 Body 参数

参数	是否必选	参数类型	描述
content	是	String	测试事件content，为json字符串的base64编码

响应参数

状态码： 200

表 7-146 响应 Body 参数

参数	参数类型	描述
id	String	测试事件ID。
name	String	测试事件名称。

状态码： 400

表 7-147 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-148 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-149 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-150 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-151 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

更新指定测试事件内容为{"k":"v"}。

```
PUT https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/events/{event_id}
{
  "content" : "eyJrljoidiJ9"
}
```

响应示例

状态码： 200

OK

```
{
  "id" : "3b659dc0-12fc-40dc-aa05-a321d9424cb3",
  "name" : "event-k9r3"
}
```

SDK 代码示例

SDK代码示例如下。

Java

更新指定测试事件内容为{"k":"v"}。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateEventSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateEventRequest request = new UpdateEventRequest();
        UpdateEventRequestBody body = new UpdateEventRequestBody();
        body.withContent("eyJrljoidiJ9");
        request.withBody(body);
        try {
            UpdateEventResponse response = client.updateEvent(request);
        }
    }
}
```

```
        System.out.println(response.toString());
    } catch (ConnectionException e) {
        e.printStackTrace();
    } catch (RequestTimeoutException e) {
        e.printStackTrace();
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

更新指定测试事件内容为{"k":"v"}。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateEventRequest()
        request.body = UpdateEventRequestBody(
            content="eyJrIjoia3V"
        )
        response = client.update_event(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

更新指定测试事件内容为{"k":"v"}。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)
```

```
func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateEventRequest{}
    request.Body = &model.UpdateEventRequestBody{
        Content: "eyJrIjoidiI9",
    }
    response, err := client.UpdateEvent(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.5 函数调用链

7.5.1 获取函数调用链配置

功能介绍

获取函数调用链配置

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/functions/{function_urn}/tracing

表 7-152 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见 FunctionGraph函数模型的描述。

请求参数

表 7-153 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 200

表 7-154 响应 Body 参数

参数	参数类型	描述
tracing_ak	String	apm的ak
tracing_sk	String	apm的sk

状态码： 403

表 7-155 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-156 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-157 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

获取函数调用链配置。

```
GET /v2/{project_id}/fgs/functions/{function_urn}/tracing
```

响应示例

状态码： 200

ok

```
{  
  "tracing_ak": "xxx",  
  "tracing_sk": "*****"  
}
```

状态码： 403

FORBIDDEN

```
{
  "error_code": "FSS.0403",
  "error_msg": "invalid token"
}
```

状态码： 404

Not Found

```
{
  "error_code": "FSS.0404",
  "error_msg": "can not find tracing aksk"
}
```

状态码： 500

内部错误

```
{
  "error_code": "FSS.0500",
  "error_msg": "xxx"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowTracingSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowTracingRequest request = new ShowTracingRequest();
        try {
            ShowTracingResponse response = client.showTracing(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        }
    }
}
```

```
    } catch (ServiceResponseException e) {
        e.printStackTrace();
        System.out.println(e.getHttpStatusCode());
        System.out.println(e.getRequestId());
        System.out.println(e.getErrorCode());
        System.out.println(e.getErrorMsg());
    }
}
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ShowTracingRequest()
        response = client.show_tracing(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
```

```
Build()

client := functiongraph.NewFunctionGraphClient(
    functiongraph.FunctionGraphClientBuilder().
        WithRegion(region.ValueOf("<YOUR REGION>")).
        WithCredential(auth).
        Build())

request := &model.ShowTracingRequest{}
response, err := client.ShowTracing(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	ok
403	FORBIDDEN
404	Not Found
500	内部错误

错误码

请参见[错误码](#)。

7.5.2 修改函数调用链配置

功能介绍

修改函数调用链配置,开通/修改传入aksk, 关闭aksk传空

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/fgs/functions/{function_urn}/tracing

表 7-158 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见FunctionGraph函数模型的描述。

请求参数

表 7-159 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

表 7-160 请求 Body 参数

参数	是否必选	参数类型	描述
tracing_ak	否	String	apm的ak
tracing_sk	否	String	apm的sk

响应参数

状态码： 400

表 7-161 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-162 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-163 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

修改函数调用链的ak，sk配置。

```
PUT /v2/{project_id}/fgs/functions/{function_urn}/tracing
{
  "tracing_ak": "xxx",
  "tracing_sk": "yyy"
}
```

响应示例

状态码： 400

Bad Request

```
{
  "error_code": "FSS.1128",
  "error_msg": "function[xxx] not existed"
}
```

状态码： 403

FORBIDDEN

```
{
  "error_code": "FSS.0403",
  "error_msg": "invalid token"
}
```

状态码： 500

内部错误

```
{
  "error_code": "FSS.0500",
  "error_msg": "xxx"
}
```

SDK 代码示例

SDK代码示例如下。

Java

修改函数调用链的ak，sk配置。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateTracingSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        UpdateTracingRequest request = new UpdateTracingRequest();
        UpdateTracingRequestBody body = new UpdateTracingRequestBody();
        body.withTracingSk("yyy");
        body.withTracingAk("xxx");
        request.withBody(body);
        try {
            UpdateTracingResponse response = client.updateTracing(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

修改函数调用链的ak，sk配置。

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateTracingRequest()
        request.body = UpdateTracingRequestBody(
            tracing_sk="yyy",
            tracing_ak="xxx"
        )
        response = client.update_tracing(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

修改函数调用链的ak，sk配置。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())
```



```
request := &model.UpdateTracingRequest{}
tracingSkUpdateTracingRequestBody:= "yyy"
tracingAkUpdateTracingRequestBody:= "xxx"
request.Body = &model.UpdateTracingRequestBody{
    TracingSk: &tracingSkUpdateTracingRequestBody,
    TracingAk: &tracingAkUpdateTracingRequestBody,
}
response, err := client.UpdateTracing(request)
if err == nil {
    fmt.Printf("%v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Bad Request
403	FORBIDDEN
500	内部错误

错误码

请参见[错误码](#)。

7.6 函数生命周期管理

7.6.1 获取函数列表

功能介绍

获取函数列表

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/functions

表 7-164 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。

表 7-165 Query 参数

参数	是否必选	参数类型	描述
marker	否	String	上一次查询到的最后的记录位置。
maxitems	否	String	每次查询获取的最大函数记录数量 最大值：400 如果不提供该值或者提供的值大于400或等于0，则使用默认值：400 如果该值小于0，则返回参数错误。
package_name	否	String	自定义分组名称。

请求参数

表 7-166 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 200

表 7-167 响应 Body 参数

参数	参数类型	描述
functions	Array of ListFunctionResult objects	函数列表。
next_marker	Long	函数下次记录读取位置。

参数	参数类型	描述
count	Long	满足查询条件的函数总数。

表 7-168 ListFunctionResult

参数	参数类型	描述
resource_id	String	资源id。
func_urn	String	函数的URN (Uniform Resource Name) , 唯一标识函数。
func_name	String	函数名称。
domain_id	String	域名id。
namespace	String	租户的project id。
project_name	String	租户的project name。
package	String	函数所属的分组Package, 用于用户针对函数的自定义分组。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> • Java8 • Java11 • Node.js6.10 • Node.js8.10 • Node.js10.16 • Node.js12.13 • Node.js14.18 • Node.js16.17 • Node.js18.15 • Python2.7 • Python3.6 • Python3.10 • Go1.x • C#(.NET Core 2.1) • C#(.NET Core 3.1) • Custom • PHP7.3 • Python3.9 • http • Custom Image • Cangjie1.0
timeout	Integer	<p>函数执行超时时间，超时函数将被强行停止，范围3~259200秒。</p>

参数	参数类型	描述
handler	String	函数执行入口 规则：xx.xx，必须包含“.” 举例：对于node.js函数：myfunction.handler，则表示函数的文件名为myfunction.js，执行的入口函数名为handler。长度不超过128个字符。 最小长度：3 最大长度：128
memory_size	Integer	函数消耗的内存。单位M。取值范围为：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128，最大值为4096。
gpu_memory	Integer	函数消耗的显存，只支持自定义运行时与自定义镜像函数配置GPU。单位MB。取值范围为：1024、2048、3072、4096、5120、6144、7168、8192、9216、10240、11264、12288、13312、14336、15360、16384。最小值为1024，最大值为16384。
cpu	Integer	函数占用的cpu资源。单位为millicore（1 core=1000 millicores）。取值与MemorySize成比例，默认是128M内存占0.1个核（100 millicores）。
code_type	String	函数代码类型，取值有5种。inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包，主要针对Java函数。Custom-Image-Swr: 函数代码来源于SWR自定义镜像。 枚举值： <ul style="list-style-type: none"> • inline • zip • obs • jar • Custom-Image-Swr
code_url	String	函数代码包在OBS上的地址。
code_filename	String	函数的文件名。
code_size	Long	函数大小，单位：字节。
user_data	String	用户自定义的name/value信息。在函数中使用的参数。举例：如函数要访问某个主机，可以设置自定义参数：Host={host_ip}，最多定义20个，总长度不超过4KB。
encrypted_user_data	String	用户自定义的name/value信息，用于需要加密的配置。

参数	参数类型	描述
digest	String	函数代码SHA512 hash值，用于判断函数是否变化。
version	String	函数版本号。
image_name	String	函数版本的内部标识。
xrole	String	函数使用的权限委托名称，需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。
app_xrole	String	函数app使用的权限委托名称，需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。
description	String	函数描述。
last_modified	String	函数最后一次更新时间。
func_vpc_id	String	用户的vpcid。
strategy_config	StrategyConfig object	函数策略配置。
extend_config	String	函数扩展配置。
initializer_handler	String	函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。举例：对于node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。
initializer_timeout	Integer	初始化超时时间，超时函数将被强行停止，范围1~300秒。当配置初始化函数时，此参数必填。
pre_stop_handler	String	函数预停止函数的入口，规则：xx.xx，必须包含“.”。举例：对于node.js函数：myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。
pre_stop_timeout	Integer	初始化超时时间，超时函数将被强行停止，范围1~90秒。
enterprise_project_id	String	企业项目ID，在企业用户创建函数时必填。
long_time	Boolean	是否允许进行长时间超时设置。
log_group_id	String	自定义日志查询组id
log_stream_id	String	自定义日志查询流id

参数	参数类型	描述
type	String	v2表示为正式版本,v1为废弃版本。 枚举值: <ul style="list-style-type: none"> • v1 • v2
fail_count	Integer	函数最近1天内执行失败的次数。
is_bridge_func tion	Boolean	是否为bridge函数
bind_bridge_f uncUrns	Array of strings	绑定bridge函数的urn列表

表 7-169 StrategyConfig

参数	参数类型	描述
concurrency	Integer	单函数最大实例数, v1取值0和-1, v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用
concurrent_nu m	Integer	单实例并发数, v2版本才支持, 取值1到1000

状态码: 401

表 7-170 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码: 403

表 7-171 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码: 404

表 7-172 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-173 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

当前租户函数列表。

```
GET https://{Endpoint}/v2/{project_id}/fgs/functions?marker=0&maxitems=400
```

响应示例

状态码： 200

ok

```
{
  "functions": [ {
    "func_urn": "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",
    "resource_id": "0d075978-5a54-4ee1-8e24-ff5bd070xxxx:test",
    "func_name": "test",
    "domain_id": "cff01_hk",
    "namespace": "7aad83af3e8d42e99ac194e8419e2c9b",
    "project_name": "xxxxxxx",
    "package": "default",
    "runtime": "Node.js6.10",
    "timeout": 3,
    "handler": "test.handler",
    "memory_size": 128,
    "cpu": 300,
    "code_type": "inline",
    "code_filename": "index.js",
    "code_size": 272,
    "digest":
      "decfce6939297b0b5ec6d1a23bf9c725870f5e69fc338a89a6a4029264688dc26338f56d08b6535de47f15ad53
      8e22ca66613b9a46f807d50b687bb53fdded1c6",
    "version": "latest",
    "xrole": "cff",
    "image_name": "latest-191025153727@zehht",
    "last_modified": "2019-10-25 15:37:27",
    "strategy_config": {
      "concurrency": 0
    },
    "initializer_handler": "index.initializer",
    "initializer_timeout": 3,
  }
]
```



```
"pre_stop_handler" : "index.pre_stop_handler",
  "pre_stop_timeout" : 3
}],
"next_marker" : 4
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListFunctionsSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        ListFunctionsRequest request = new ListFunctionsRequest();
        request.withMarker("<marker>");
        request.withMaxitems("<maxitems>");
        request.withPackageName("<package_name>");
        try {
            ListFunctionsResponse response = client.listFunctions(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = ListFunctionsRequest()
        request.marker = "<marker>"
        request.maxitems = "<maxitems>"
        request.package_name = "<package_name>"
        response = client.list_functions(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ListFunctionsRequest{}
    markerRequest := "<marker>"
    request.Marker = &markerRequest
```

```
maxitemsRequest:= "<maxitems>"
request.Maxitems = &maxitemsRequest
packageNameRequest:= "<package_name>"
request.PackageName = &packageNameRequest
response, err := client.ListFunctions(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	ok
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.6.2 创建函数

功能介绍

创建指定的函数。

调用方法

请参见[如何调用API](#)。

URI

POST /v2/{project_id}/fgs/functions

表 7-174 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。

请求参数

表 7-175 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

表 7-176 请求 Body 参数

参数	是否必选	参数类型	描述
func_name	是	String	函数名称。 可包含字母、数字、下划线和中划线，以大/小写字母开头，以字母或数字结尾，长度不超过60个字符。 最小长度：1 最大长度：60
package	是	String	函数所属的分组Package，用于用户针对函数的自定义分组。

参数	是否必选	参数类型	描述
runtime	是	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> • http • Custom Image • Cangjie1.0
timeout	是	Integer	函数执行超时时间，超时函数将被强行停止，范围3~259200秒。
handler	是	String	<p>函数执行入口规则：xx.xx，必须包含“.”；自定义镜像函数handler为“-”</p> <p>举例：对于node.js函数：myfunction.handler，则表示函数的文件名为myfunction.js，执行的入口函数名为handler。长度不超过128个字符。</p> <p>最小长度：3 最大长度：128</p>
depend_version_list	否	Array of strings	依赖版本id列表
func_vpc	否	FuncVpc object	函数vpc配置。需同时为函数配置具有vpc权限的委托。
memory_size	是	Integer	函数消耗的内存。单位M。取值范围为：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128，最大值为4096。
gpu_memory	否	Integer	<p>函数消耗的显存，只支持自定义运行时与自定义镜像函数配置GPU。单位MB。</p> <p>取值范围为：1024、2048、3072、4096、5120、6144、7168、8192、9216、10240、11264、12288、13312、14336、15360、16384。最小值为1024，最大值为16384。（目前只支持华东-上海一）</p>
gpu_type	否	String	显卡类型。（目前只支持华东-上海一）

参数	是否必选	参数类型	描述
code_type	否	String	<p>函数代码类型，取值有5种。</p> <p>inline: UI在线编辑代码。 zip: 函数代码为zip包。 obs: 函数代码来源于obs存储。 jar: 函数代码为jar包，主要针对Java函数。 Custom-Image-Swr: 函数代码来源于SWR自定义镜像。</p> <p>创建自定义镜像函数此参数非必填，其他类型函数此参数必填。</p> <p>枚举值：</p> <ul style="list-style-type: none"> • inline • zip • obs • jar • Custom-Image-Swr
code_url	否	String	<p>当code_type为obs时，该值为函数代码包在OBS上的地址，code_type为其他值时，该字段为空。</p>
code_filename	否	String	<p>函数的文件名，当code_type为jar/zip时必须提供该字段，code_type为其他值时不需要提供。</p>
custom_image	否	CustomImage object	<p>用户容器镜像，当code_type为Custom-Image-Swr时必须提供该字段，code_type为其他值时不需要提供。</p>
user_data	否	String	<p>用户自定义的name/value信息。在函数中使用的参数。</p> <p>举例：如函数要访问某个主机，可以设置自定义参数： Host={host_ip}，最多定义20个，总长度不超过4KB。</p>
encrypted_user_data	否	String	<p>用户自定义的name/value信息，用于需要加密的配置。</p> <p>举例：如配置加密密码，可以设置自定义参数： password={1234}，最多定义20个，总长度不超过4KB。</p>

参数	是否必选	参数类型	描述
xrole	否	String	函数配置委托。需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。 配置后用户可以通过函数执行入口方法中的context参数获取具有委托中权限的token、ak、sk，用于访问其他云服务。如果用户函数不访问任何云服务，则不用提供委托名称。
app_xrole	否	String	函数执行委托。可为函数执行单独配置执行委托，这将减小不必要的性能损耗；不单独配置执行委托时，函数执行和函数配置将使用同一委托。
description	否	String	函数描述。
func_code	否	FuncCode object	函数代码。
mount_config	否	MountConfig object	函数挂载配置。
initializer_handler	否	String	函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。 举例：对于node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。
initializer_timeout	否	Integer	初始化超时时间，超时函数将被强行停止，范围1~300秒。 当配置初始化函数时，此参数必填。
pre_stop_handler	否	String	函数预停止函数的入口，规则：xx.xx，必须包含“.”。 举例：对于node.js函数：myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。
pre_stop_timeout	否	Integer	初始化超时时间，超时函数将被强行停止，范围1~90秒。
enterprise_project_id	否	String	企业项目ID，在企业用户创建函数时必填。

参数	是否必选	参数类型	描述
type	否	String	函数版本；部分局点只支持v1函数，缺省值则为v1。 缺省值： v2 枚举值： <ul style="list-style-type: none"> • v1 • v2
log_config	否	FuncLogConfig object	函数绑定日志配置。
network_controller	否	NetworkControllerConfig object	函数网络配置。
is_stateful_function	否	Boolean	是否支持有状态，如果需要支持，需要固定传参为true，v2版本支持。
enable_dynamic_memory	否	Boolean	是否启动动态内存配置。
lts_custom_tag	否	Map<String,String>	自定义日志标签。 函数执行时，可以按照自定义标签配置上报标签到云日志服务(LTS)，用户可以通过标签对日志进行过滤筛选。

表 7-177 FuncVpc

参数	是否必选	参数类型	描述
domain_id	否	String	域名id。
namespace	否	String	租户的project id。
vpc_name	否	String	虚拟私有云名称。
vpc_id	是	String	虚拟私有云唯一标识。
subnet_name	否	String	子网名称。
subnet_id	是	String	子网编号。
cidr	否	String	子网掩码。
gateway	否	String	网关。
security_groups	否	Array of strings	安全组。

表 7-178 CustomImage

参数	是否必选	参数类型	描述
enabled	否	Boolean	是否启用
image	否	String	镜像地址
command	否	String	启动容器镜像的命令
args	否	String	启动容器镜像的命令行参数
working_dir	否	String	镜像容器工作目录
uid	否	String	镜像容器的用户id
gid	否	String	镜像容器的用户组id

表 7-179 FuncCode

参数	是否必选	参数类型	描述
file	否	String	函数代码，如果不为空必须进行base64编码，为空时使用默认的代码。
link	否	String	函数代码链接。

表 7-180 MountConfig

参数	是否必选	参数类型	描述
mount_user	是	MountUser object	挂载用户信息。
func_mounts	是	Array of FuncMount objects	函数挂载列表。

表 7-181 MountUser

参数	是否必选	参数类型	描述
user_id	是	String	用户ID(-1~65534的非0整数)
user_group_id	是	String	用户组ID(-1~65534的非0整数)

表 7-182 FuncMount

参数	是否必选	参数类型	描述
mount_type	是	String	挂载类型(sfs/sfsTurbo/ecs), func_mounts非空时必选。
mount_resource	是	String	挂载资源ID (对应云服务ID), func_mounts非空时必选。
mount_share_path	否	String	远端挂载路径 (例如 192.168.0.12:/data), 如果 mount_type为ecs, 必选。
local_mount_path	是	String	函数访问路径, func_mounts非空时必选。

表 7-183 FuncLogConfig

参数	是否必选	参数类型	描述
group_name	否	String	函数绑定日志组名。
group_id	否	String	函数绑定日志组ID。
stream_name	否	String	函数绑定日志流名。
stream_id	否	String	函数绑定日志流ID。

表 7-184 NetworkControlConfig

参数	是否必选	参数类型	描述
disable_public_network	否	Boolean	禁止公网访问开关。
trigger_access_vpcs	否	Array of VpcConfig objects	指定触发函数vpc配置。

表 7-185 VpcConfig

参数	是否必选	参数类型	描述
vpc_name	否	String	vpc名称。
vpc_id	否	String	vpc ID。

响应参数

状态码： 200

表 7-186 响应 Body 参数

参数	参数类型	描述
func_id	String	函数id, 唯一标识函数。
func_urn	String	函数的URN (Uniform Resource Name), 唯一标识函数。
func_name	String	函数名称。
domain_id	String	域名id。
namespace	String	租户的project id。
project_name	String	租户的project name。
package	String	函数所属的分组Package, 用于用户针对函数的自定义分组。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0
timeout	Integer	<p>函数执行超时时间，超时函数将被强行停止，范围3~259200秒。</p>

参数	参数类型	描述
handler	String	函数执行入口 规则：xx.xx，必须包含“.” 举例：对于node.js函数：myfunction.handler，则表示函数的文件名为myfunction.js，执行的入口函数名为handler。长度不超过128个字符。 最小长度：3 最大长度：128
memory_size	Integer	函数消耗的内存。单位M。 取值范围为：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128，最大值为4096。
gpu_memory	Integer	函数消耗的显存，只支持自定义运行时与自定义镜像函数配置GPU。单位MB。 取值范围为：1024、2048、3072、4096、5120、6144、7168、8192、9216、10240、11264、12288、13312、14336、15360、16384。最小值为1024，最大值为16384。
cpu	Integer	函数占用的cpu资源。单位为millicore（1 core=1000 millicores）。 取值与MemorySize成比例，默认是128M内存占0.1个核（100 millicores）。
code_type	String	函数代码类型，取值有5种。 inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包，主要针对Java函数。Custom-Image-Swr: 函数代码来源于SWR自定义镜像。 枚举值： <ul style="list-style-type: none"> ● inline ● zip ● obs ● jar ● Custom-Image-Swr
code_url	String	当code_type为obs时，该值为函数代码包在OBS上的地址，code_type为其他值时，该字段为空。
code_filename	String	函数的文件名，当code_type为jar/zip时必须提供该字段，inline和obs不需要提供。
code_size	Long	函数大小，单位：字节。
domain_names	String	函数配置的需要支持域名解析的内网域名。

参数	参数类型	描述
user_data	String	用户自定义的name/value信息。在函数中使用的参数。 举例：如函数要访问某个主机，可以设置自定义参数：Host={host_ip}，最多定义20个，总长度不超过4KB。
encrypted_user_data	String	用户自定义的name/value信息，用于需要加密的配置。
digest	String	函数代码SHA512 hash值，用于判断函数是否变化。
version	String	函数版本号。
image_name	String	函数版本的内部标识。
xrole	String	函数使用的权限委托名称，需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。
app_xrole	String	函数app使用的权限委托名称，需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。
description	String	函数描述。
last_modified	String	函数最后一次更新时间。
func_vpc	FuncVpc object	函数vpc配置。需同时为函数配置具有vpc权限的委托。
mount_config	MountConfig object	函数挂载配置。
reserved_instance_count	Integer	函数预留实例数量。
depend_list	Array of strings	依赖id列表
depend_version_list	Array of strings	依赖版本id列表
strategy_config	StrategyConfig object	函数策略配置。
extend_config	String	函数扩展配置。
dependencies	Array of Dependency objects	函数依赖代码包列表。

参数	参数类型	描述
initializer_handler	String	函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。 举例：对于node.js函数： myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。
initializer_timeout	Integer	初始化超时时间，超时函数将被强行停止，范围1~300秒。当配置初始化函数时，此参数必填。
pre_stop_handler	String	函数预停止函数的入口，规则：xx.xx，必须包含“.”。 举例：对于node.js函数： myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。
pre_stop_timeout	Integer	初始化超时时间，超时函数将被强行停止，范围1~90秒。
enterprise_project_id	String	企业项目ID，在企业用户创建函数时必填。
long_time	Boolean	是否允许进行长时间超时设置。
log_group_id	String	自定义日志查询组id
log_stream_id	String	自定义日志查询流id
type	String	v2表示为正式版本,v1为废弃版本。 枚举值： <ul style="list-style-type: none"> • v1 • v2
enable_cloud_debug	String	适配CloudDebug场景，是否开启云调试（已废弃）
enable_dynamic_memory	Boolean	是否启动动态内存配置
is_stateful_function	Boolean	是否支持有状态，v2版本支持
custom_image	CustomImage object	用户容器镜像。
is_bridge_function	Boolean	是否为bridge函数
apig_route_enabled	Boolean	是否配置下沉apig路由规则。

参数	参数类型	描述
heartbeat_handler	String	心跳函数函数的入口，规则：xx.xx，必须包含“.”，只支持JAVA运行时配置。 心跳函数入口需要与函数执行入口在同一文件下。在开启心跳函数配置时，此参数必填。
enable_class_isolation	Boolean	类隔离开关，只支持JAVA运行时配置。 开启类隔离后可以支持Kafka转储并提升类加载效率，但也可能会导致某些兼容性问题，请谨慎开启。
gpu_type	String	显卡类型。
allow_ephemeral_storage	Boolean	是否支持配置临时存储。
ephemeral_storage	Integer	临时存储大小。 默认情况下会为函数的/tmp目录分配512MB的空间。您可以通过临时存储设置将函数的/tmp目录大小调整为10G。 枚举值： <ul style="list-style-type: none"> • 512 • 10240
network_controller	NetworkControlConfig object	函数网络配置。
resource_id	String	资源id。
is_return_stream	Boolean	是否返回流式数据（已废弃）。
enable_auth_in_header	Boolean	是否允许在请求头中添加鉴权信息，只支持自定义镜像函数（创建函数时不支持修改）。
lts_custom_tag	Map<String,String>	自定义日志标签。 函数执行时，可以按照自定义标签配置上报标签到云日志服务(LTS)，用户可以通过标签对日志进行过滤筛选。

表 7-187 FuncVpc

参数	参数类型	描述
domain_id	String	域名id。
namespace	String	租户的project id。
vpc_name	String	虚拟私有云名称。

参数	参数类型	描述
vpc_id	String	虚拟私有云唯一标识。
subnet_name	String	子网名称。
subnet_id	String	子网编号。
cidr	String	子网掩码。
gateway	String	网关。
security_groups	Array of strings	安全组。

表 7-188 MountConfig

参数	参数类型	描述
mount_user	MountUser object	挂载用户信息。
func_mounts	Array of FuncMount objects	函数挂载列表。

表 7-189 MountUser

参数	参数类型	描述
user_id	String	用户ID(-1~65534的非0整数)
user_group_id	String	用户组ID(-1~65534的非0整数)

表 7-190 FuncMount

参数	参数类型	描述
mount_type	String	挂载类型(sfs/sfsTurbo/ecs), func_mounts非空时必选。
mount_resource	String	挂载资源ID (对应云服务ID), func_mounts非空时必选。
mount_share_path	String	远端挂载路径 (例如192.168.0.12:/data), 如果 mount_type为ecs, 必选。
local_mount_path	String	函数访问路径, func_mounts非空时必选。

表 7-191 StrategyConfig

参数	参数类型	描述
concurrency	Integer	单函数最大实例数，v1取值0和-1，v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用
concurrent_num	Integer	单实例并发数，v2版本才支持，取值1到1000

表 7-192 Dependency

参数	参数类型	描述
id	String	依赖包版本ID。
owner	String	依赖包属主的domainId。
link	String	依赖包在OBS上的链接。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0
etag	String	依赖包的md5值
size	Long	依赖包大小。

参数	参数类型	描述
name	String	依赖包名称。
description	String	依赖包描述。
file_name	String	依赖包文件名，如果创建方式为zip时。
version	Long	依赖包版本编号。
dep_id	String	依赖包ID
last_modified	Integer	函数最后一次更新时间。

表 7-193 CustomImage

参数	参数类型	描述
enabled	Boolean	是否启用
image	String	镜像地址
command	String	启动容器镜像的命令
args	String	启动容器镜像的命令行参数
working_dir	String	镜像容器工作目录
uid	String	镜像容器的用户id
gid	String	镜像容器的用户组id

表 7-194 NetworkControlConfig

参数	参数类型	描述
disable_public_network	Boolean	禁止公网访问开关。
trigger_access_vpcs	Array of VpcConfig objects	指定触发函数vpc配置。

表 7-195 VpcConfig

参数	参数类型	描述
vpc_name	String	vpc名称。
vpc_id	String	vpc ID。

状态码：400**表 7-196** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码：401**表 7-197** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码：403**表 7-198** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码：404**表 7-199** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码：500**表 7-200** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误信息

状态码： 503

表 7-201 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

- 创建名为xxx的python2.7函数，内存大小为128MB，超时时间3s，添加加密环境变量password，添加磁盘挂载。

POST https://{Endpoint}/v2/{project_id}/fgs/functions

```
{
  "func_name": "xxx",
  "package": "xxx",
  "runtime": "Python2.7",
  "handler": "index.py",
  "type": "v2",
  "memory_size": 128,
  "timeout": 3,
  "code_type": "inline",
  "mount_config": {
    "mount_user": {
      "user_id": -1,
      "user_group_id": -1
    }
  },
  "func_mounts": [ {
    "mount_resource": "268xxx-d3xx-4cxx-98xx-d1dbxxx",
    "local_mount_path": "/mnt",
    "mount_type": "sfsTurbo"
  } ]
},
"encrypted_user_data": "{\\"password\\":\\"1234\\"}",
"func_code": {
  "file": "xxxxx",
  "link": ""
}
}
```

- 创建名为xxx的python3.9函数，内存大小为128MB，超时时间3s，代码由OBS上传。

POST https://{Endpoint}/v2/{project_id}/fgs/functions

```
{
  "func_name": "xxx",
  "package": "xxx",
  "runtime": "Python3.9",
  "handler": "index.py",
  "type": "v2",
  "memory_size": 128,
  "timeout": 3,
  "code_type": "obs",
}
```

- ```
"code_url" : "https:xxx"
}
```
- 创建名为xxx的自定义镜像函数，内存大小为128MB，超时时间3s，代码由SWR上传。  
POST https://{Endpoint}/v2/{project\_id}/fgs/functions
- ```
{
  "func_name" : "xxx",
  "package" : "xxx",
  "runtime" : "Custom Image",
  "handler" : "-",
  "type" : "v2",
  "memory_size" : 128,
  "timeout" : 3,
  "code_type" : "obs",
  "code_url" : "https:xxx",
  "xrole" : "xxx",
  "custom_image" : {
    "enabled" : true,
    "image" : "swr.xxx"
  }
}
```

响应示例

状态码： 200

ok

```
{
  "func_urn" :
  "urn:fss:xxxxxxxx:c3b2459a6d5e4b548e6777e57852692d:function:default:TestCreateFunctionInPythonSdk:latest",
  "func_name" : "TestCreateFunctionInPythonSdk",
  "domain_id" : "14ee2e3501124efcbca7998baa24xxxx",
  "namespace" : "46b6f338fc3445b8846c71dfb1fbxxxx",
  "project_name" : "xxxxxxxx",
  "package" : "default",
  "runtime" : "Python3.6",
  "timeout" : 3,
  "handler" : "index.handler",
  "memory_size" : 128,
  "encrypted_user_data" : "{\"password\":\"\"}",
  "mount_config" : {
    "mount_user" : {
      "user_id" : -1,
      "user_group_id" : -1
    }
  },
  "func_mounts" : [ {
    "mount_resource" : "268xxx-d3xx-4cxx-98xx-d1dbxxx",
    "local_mount_path" : "/mnt",
    "mount_type" : "sfsTurbo"
  } ]
},
"cpu" : 300,
"code_type" : "inline",
"code_filename" : "index.py",
"code_size" : 100,
"digest" :
"1c8610d1870731a818a037f1d2adf3223e8ac351aeb293fb1f8eabd2e9820069a61ed8b5d38182e760adc33a307d0e957afc357f415cd8c9c3ff6f0426fd85cd",
"version" : "latest",
"image_name" : "latest-191025153727@zehht",
"last_modified" : "2019-10-25 15:37:27",
"strategy_config" : {
  "concurrency" : 0
}
}
```


状态码： 400

Bad Request 请求错误。

```
{
  "error_code": "FSS.0400",
  "error_msg": "invalid parameter"
}
```

状态码： 401

Unauthorized 鉴权失败。

```
{
  "error_code": "FSS.0401",
  "error_msg": "not permit"
}
```

状态码： 403

Forbidden 没有操作权限。

```
{
  "error_code": "FSS.0403",
  "error_msg": "invalid token"
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 创建名为xxx的python2.7函数，内存大小为128MB，超时时间3s，添加加密环境变量password，添加磁盘挂载。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateFunctionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
```

```
CreateFunctionRequest request = new CreateFunctionRequest();
CreateFunctionRequestBody body = new CreateFunctionRequestBody();
FuncCode funcCodebody = new FuncCode();
funcCodebody.withFile("xxxx")
    .withLink("");
body.withType(CreateFunctionRequestBody.TypeEnum.fromValue("v2"));
body.withFuncCode(funcCodebody);
body.withCodeType(CreateFunctionRequestBody.CodeTypeEnum.fromValue("inline"));
body.withMemorySize(128);
body.withHandler("index.py");
body.withTimeout(3);
body.withRuntime(CreateFunctionRequestBody.RuntimeEnum.fromValue("Python2.7"));
body.withPackage("xxx");
body.withFuncName("xxx");
request.withBody(body);
try {
    CreateFunctionResponse response = client.createFunction(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- 创建名为xxx的python3.9函数，内存大小为128MB，超时时间3s，代码由OBS上传。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateFunctionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        CreateFunctionRequest request = new CreateFunctionRequest();
        CreateFunctionRequestBody body = new CreateFunctionRequestBody();
        body.withType(CreateFunctionRequestBody.TypeEnum.fromValue("v2"));
        body.withCodeUrl("https:xxx");
    }
}
```

```
body.withCodeType(CreateFunctionRequestBody.CodeTypeEnum.fromValue("obs"));
body.withMemorySize(128);
body.withHandler("index.py");
body.withTimeout(3);
body.withRuntime(CreateFunctionRequestBody.RuntimeEnum.fromValue("Python3.9"));
body.withPackage("xxx");
body.withFuncName("xxx");
request.withBody(body);
try {
    CreateFunctionResponse response = client.createFunction(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- 创建名为xxx的自定义镜像函数，内存大小为128MB，超时时间3s，代码由SWR上传。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateFunctionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();

        CreateFunctionRequest request = new CreateFunctionRequest();
        CreateFunctionRequestBody body = new CreateFunctionRequestBody();
        CustomImage customImagebody = new CustomImage();
        customImagebody.setEnabled(true)
            .withImage("swr.xxx");
        body.withType(CreateFunctionRequestBody.TypeEnum.fromValue("v2"));
        body.withXrole("xxx");
        body.withCustomImage(customImagebody);
        body.withCodeUrl("https:xxx");
        body.withCodeType(CreateFunctionRequestBody.CodeTypeEnum.fromValue("obs"));
        body.withMemorySize(128);
    }
}
```

```
body.withHandler("-");
body.withTimeout(3);
body.withRuntime(CreateFunctionRequestBody.RuntimeEnum.fromValue("Custom Image"));
body.withPackage("xxx");
body.withFuncName("xxx");
request.withBody(body);
try {
    CreateFunctionResponse response = client.createFunction(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- 创建名为xxx的python2.7函数，内存大小为128MB，超时时间3s，添加加密环境变量password，添加磁盘挂载。

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    # environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    # environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = CreateFunctionRequest()
        funcCodebody = FuncCode(
            file="xxxxx",
            link=""
        )
        request.body = CreateFunctionRequestBody(
            type="v2",
            func_code=funcCodebody,
            code_type="inline",
            memory_size=128,
            handler="index.py",
            timeout=3,
            runtime="Python2.7",
            package="xxx",
            func_name="xxx"
        )
        response = client.create_function(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 创建名为xxx的python3.9函数，内存大小为128MB，超时时间3s，代码由OBS上传。

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *
```

```
if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")
```

```
    credentials = BasicCredentials(ak, sk) \
```

```
    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()
```

```
try:
```

```
    request = CreateFunctionRequest()
    request.body = CreateFunctionRequestBody(
        type="v2",
        code_url="https:xxx",
        code_type="obs",
        memory_size=128,
        handler="index.py",
        timeout=3,
        runtime="Python3.9",
        package="xxx",
        func_name="xxx"
    )
```

```
    response = client.create_function(request)
    print(response)
```

```
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 创建名为xxx的自定义镜像函数，内存大小为128MB，超时时间3s，代码由SWR上传。

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *
```

```
if __name__ == "__main__":
```

```
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
    # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = CreateFunctionRequest()
    customImagebody = CustomImage(
        enabled=True,
        image="swr.xxx"
    )
    request.body = CreateFunctionRequestBody(
        type="v2",
        xrole="xxx",
        custom_image=customImagebody,
        code_url="https:xxx",
        code_type="obs",
        memory_size=128,
        handler="-",
        timeout=3,
        runtime="Custom Image",
        package="xxx",
        func_name="xxx"
    )
    response = client.create_function(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

- 创建名为xxx的python2.7函数，内存大小为128MB，超时时间3s，添加加密环境变量password，添加磁盘挂载。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).
Build()

request := &model.CreateFunctionRequest{}
fileFuncCode:= "xxxx"
linkFuncCode:= ""
funcCodebody := &model.FuncCode{
    File: &fileFuncCode,
    Link: &linkFuncCode,
}
typeCreateFunctionRequestBody:= model.GetCreateFunctionRequestBodyTypeEnum().V2
request.Body = &model.CreateFunctionRequestBody{
    Type: &typeCreateFunctionRequestBody,
    FuncCode: funcCodebody,
    CodeType: model.GetCreateFunctionRequestBodyCodeTypeEnum().INLINE,
    MemorySize: int32(128),
    Handler: "index.py",
    Timeout: int32(3),
    Runtime: model.GetCreateFunctionRequestBodyRuntimeEnum().PYTHON2_7,
    Package: "xxx",
    FuncName: "xxx",
}
response, err := client.CreateFunction(request)
if err == nil {
    fmt.Printf("%+v\n", response)
} else {
    fmt.Println(err)
}
}
```

- 创建名为xxx的python3.9函数，内存大小为128MB，超时时间3s，代码由OBS上传。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build()

    request := &model.CreateFunctionRequest{}
    typeCreateFunctionRequestBody:= model.GetCreateFunctionRequestBodyTypeEnum().V2
    codeUrlCreateFunctionRequestBody:= "https:xxx"
    request.Body = &model.CreateFunctionRequestBody{
        Type: &typeCreateFunctionRequestBody,
        CodeUrl: &codeUrlCreateFunctionRequestBody,
        CodeType: model.GetCreateFunctionRequestBodyCodeTypeEnum().OBS,
    }
}
```

```
    MemorySize: int32(128),
    Handler: "index.py",
    Timeout: int32(3),
    Runtime: model.GetCreateFunctionRequestBodyRuntimeEnum().PYTHON3_9,
    Package: "xxx",
    FuncName: "xxx",
  }
  response, err := client.CreateFunction(request)
  if err == nil {
    fmt.Printf("%+v\n", response)
  } else {
    fmt.Println(err)
  }
}
```

- 创建名为xxx的自定义镜像函数，内存大小为128MB，超时时间3s，代码由SWR上传。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.CreateFunctionRequest{
        enabledCustomImage:= true
        imageCustomImage:= "swr.xxx"
        customImagebody := &model.CustomImage{
            Enabled: &enabledCustomImage,
            Image: &imageCustomImage,
        }
    }
    typeCreateFunctionRequestBody:= model.GetCreateFunctionRequestBodyTypeEnum().V2
    xroleCreateFunctionRequestBody:= "xxx"
    codeUrlCreateFunctionRequestBody:= "https:xxx"
    request.Body = &model.CreateFunctionRequestBody{
        Type: &typeCreateFunctionRequestBody,
        Xrole: &xroleCreateFunctionRequestBody,
        CustomImage: customImagebody,
        CodeUrl: &codeUrlCreateFunctionRequestBody,
        CodeType: model.GetCreateFunctionRequestBodyCodeTypeEnum().OBS,
        MemorySize: int32(128),
        Handler: "-",
        Timeout: int32(3),
        Runtime: model.GetCreateFunctionRequestBodyRuntimeEnum().CUSTOM_IMAGE,
        Package: "xxx",
        FuncName: "xxx",
    }
```



```
}  
response, err := client.CreateFunction(request)  
if err == nil {  
    fmt.Printf("%+v\n", response)  
} else {  
    fmt.Println(err)  
}  
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	ok
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。
503	Service Unavailable 服务不可用。

错误码

请参见[错误码](#)。

7.6.3 删除函数/版本

功能介绍

删除指定的函数或者特定的版本（不允许删除latest版本）。

如果URN中包含函数版本或者别名，则删除特定的函数版本或者别名指向的版本以及该版本关联的trigger。如果URN中不包含版本或者别名，则删除整个函数，包含所有版本以及别名，触发器。

调用方法

请参见[如何调用API](#)。

URI

DELETE /v2/{project_id}/fgs/functions/{function_urn}

表 7-202 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见FunctionGraph函数模型的描述。 不允许删除函数的latest版本，如要删除整个函数（包含所有版本），提供不带任何版本号/别名的urn，如： urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test

请求参数

表 7-203 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 401

表 7-204 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-205 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-206 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-207 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

删除租户指定函数/版本。

```
DELETE https://{Endpoint}/v2/{project_id}/fgs/functions/{func_urn}
```

响应示例

无

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
```

```
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class DeleteFunctionSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        DeleteFunctionRequest request = new DeleteFunctionRequest();
        try {
            DeleteFunctionResponse response = client.deleteFunction(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = DeleteFunctionRequest()
        response = client.delete_function(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.DeleteFunctionRequest{}
    response, err := client.DeleteFunction(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
204	No Content
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。

状态码	描述
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.6.4 获取指定函数代码信息

功能介绍

获取指定函数代码信息。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/functions/{function_urn}/code

表 7-208 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见 FunctionGraph函数模型的描述。

请求参数

表 7-209 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码: 200

表 7-210 响应 Body 参数

参数	参数类型	描述
func_urn	String	函数的URN (Uniform Resource Name), 唯一标识函数。
func_name	String	函数名称。
domain_id	String	域名id。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0

参数	参数类型	描述
code_type	String	函数代码类型，取值有5种。inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包，主要针对Java函数。Custom-Image-Swr: 函数代码来源于SWR自定义镜像。 枚举值： <ul style="list-style-type: none"> • inline • zip • obs • jar • Custom-Image-Swr
code_url	String	当code_type为obs时，该值为函数代码包在OBS上的地址，code_type为其他值时，该字段为空。
code_filename	String	函数的文件名，当code_type为jar/zip时必须提供该字段，inline和obs不需要提供。
code_size	Long	函数大小，单位：字节。
digest	String	函数代码SHA512 hash值，用于判断函数是否变化。
last_modified	String	函数最后一次更新时间。
func_code	FuncCode object	函数代码。
depend_list	Array of strings	依赖id列表
depend_version_list	Array of strings	依赖版本id列表
strategy_config	StrategyConfig object	函数策略配置。
dependencies	Array of Dependency objects	函数依赖代码包列表。

表 7-211 FuncCode

参数	参数类型	描述
file	String	函数代码，如果不为空必须进行base64编码，为空时使用默认的代码。
link	String	函数代码链接。

表 7-212 StrategyConfig

参数	参数类型	描述
concurrency	Integer	单函数最大实例数，v1取值0和-1，v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用
concurrent_num	Integer	单实例并发数，v2版本才支持，取值1到1000

表 7-213 Dependency

参数	参数类型	描述
id	String	依赖包版本ID。
owner	String	依赖包属主的domainId。
link	String	依赖包在OBS上的链接。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0
etag	String	依赖包的md5值
size	Long	依赖包大小。

参数	参数类型	描述
name	String	依赖包名称。
description	String	依赖包描述。
file_name	String	依赖包文件名，如果创建方式为zip时。
version	Long	依赖包版本编号。
dep_id	String	依赖包ID
last_modified	Integer	函数最后一次更新时间。

状态码： 400

表 7-214 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-215 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-216 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-217 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-218 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

获取指定函数代码信息。

```
GET https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/code
```

响应示例

状态码： 200

OK

```
{
  "func_urn": "urn:fss:xxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest",
  "func_name": "test",
  "domain_id": "14ee2e3501124efcbca7998baa24xxxx",
  "runtime": "Node.js6.10",
  "code_type": "inline",
  "code_filename": "index.zip",
  "code_size": 245,
  "digest":
    "39d51ae334ffb239037e194159f4cc9cec31086719aa73db7d0a608343121a290f7115693e008bd01cb12b99a803856e83aa96173b26c528c41cfed18267e08f",
  "last_modified": "2019-10-31 11:37:58",
  "func_code": {
    "file": "",
    "link": ""
  },
  "strategy_config": {
    "concurrency": -1
  }
}
```

状态码： 404

Not Found 找不到资源。

```
{
  "error_code": "FSS.1052",
  "error_msg": "Not found the function version"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowFunctionCodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowFunctionCodeRequest request = new ShowFunctionCodeRequest();
        try {
            ShowFunctionCodeResponse response = client.showFunctionCode(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
```

```
# In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
    .with_credentials(credentials) \
    .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
    .build()

try:
    request = ShowFunctionCodeRequest()
    response = client.show_function_code(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowFunctionCodeRequest{}
    response, err := client.ShowFunctionCode(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.6.5 修改函数代码

功能介绍

修改指定的函数的代码。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/fgs/functions/{function_urn}/code

表 7-219 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见 FunctionGraph函数模型的描述。

请求参数

表 7-220 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

表 7-221 请求 Body 参数

参数	是否必选	参数类型	描述
code_type	是	String	函数代码类型，取值有5种。 inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包，主要针对Java函数。修改自定义镜像函数镜像配置请参考： 修改自定义镜像类型函数的镜像配置 。 枚举值： <ul style="list-style-type: none"> • inline • zip • obs • jar
code_url	否	String	当code_type为obs时，该值为函数代码包在OBS上的地址，code_type为其他值时，该字段为空。
code_filename	否	String	函数的文件名，当code_type为jar/zip时必须提供该字段，inline和obs不需要提供。
func_code	是	FuncCode object	函数代码。
depend_version_list	否	Array of strings	依赖版本id列表

表 7-222 FuncCode

参数	是否必选	参数类型	描述
file	否	String	函数代码，如果不为空必须进行 base64 编码，为空时使用默认的代码。
link	否	String	函数代码链接。

响应参数

状态码： 200

表 7-223 响应 Body 参数

参数	参数类型	描述
func_urn	String	函数的 URN (Uniform Resource Name)，唯一标识函数。
func_name	String	函数名称。
domain_id	String	域名 id。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0

参数	参数类型	描述
code_type	String	函数代码类型，取值有5种。inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包，主要针对Java函数。修改自定义镜像函数镜像配置请参考： 修改自定义镜像类型函数的镜像配置 。 枚举值： <ul style="list-style-type: none"> • inline • zip • obs • jar
code_url	String	当code_type为obs时，该值为函数代码包在OBS上的地址，code_type为其他值时，该字段为空。
code_filename	String	函数的文件名，当code_type为jar/zip时必须提供该字段，inline和obs不需要提供。
code_size	Long	函数大小，单位：字节。
digest	String	函数代码SHA512 hash值，用于判断函数是否变化。
last_modified	String	函数最后一次更新时间。
func_code	FuncCode object	函数代码。
depend_list	Array of strings	依赖id列表
depend_version_list	Array of strings	依赖版本id列表
strategy_config	StrategyConfig object	函数策略配置。
dependencies	Array of Dependency objects	函数依赖代码包列表。

表 7-224 FuncCode

参数	参数类型	描述
file	String	函数代码，如果不为空必须进行base64编码，为空时使用默认的代码。
link	String	函数代码链接。

表 7-225 StrategyConfig

参数	参数类型	描述
concurrency	Integer	单函数最大实例数，v1取值0和-1，v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用
concurrent_num	Integer	单实例并发数，v2版本才支持，取值1到1000

表 7-226 Dependency

参数	参数类型	描述
id	String	依赖包版本ID。
owner	String	依赖包属主的domainId。
link	String	依赖包在OBS上的链接。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0
etag	String	依赖包的md5值
size	Long	依赖包大小。

参数	参数类型	描述
name	String	依赖包名称。
description	String	依赖包描述。
file_name	String	依赖包文件名，如果创建方式为zip时。
version	Long	依赖包版本编号。
dep_id	String	依赖包ID
last_modified	Integer	函数最后一次更新时间。

状态码： 400

表 7-227 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-228 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-229 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-230 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-231 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

- 以inline方式修改指定函数代码。

PUT https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/code

```
{
  "code_filename": "index.zip",
  "code_type": "inline",
  "func_code": {
    "file": "UESDBAoAAAAIAPQ1M1gNImPLrAAAAAEBAaIAAAaW5kZXGucHlNjtEOgjAMRd/
5igVfxDALxhjDo0S/wB+YrMgMdMvWGYnh390wEfrU3nVb0xXjG85qLRU
+Sk8NP0UhUb3RltjTaUwkNKwVKDuwbA0vQMrDAhK8KsTFsoCeYvsMw2xUkeCvKu0hLRk
+6LIZ0u5s3BwPFwwUEEG/yo6B4vEXcshyBG
+lb437kfNfpEWWhATrQmqGtKYVH0Pit8fEdCqM6VQtSGncxYppPz5O3fgFUEsBAh4DCgAAAAgA9DUzWA
0iY8usAAAAAQEAAAgAAAAAAAAAAAAAAAAAPMCAAAAAGluZGV4LnB5UEsFBgAAAAABAAEANGAAANIAA
AAAAA=="
  }
}
```

- 以zip方式修改指定函数代码。

PUT https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/code

```
{
  "code_filename": "index.zip",
  "code_type": "zip",
  "func_code": {
    "file": "UESDBAoAAAAIAPQ1M1gNImPLrAAAAAEBAaIAAAaW5kZXGucHlNjtEOgjAMRd/
5igVfxDALxhjDo0S/wB+YrMgMdMvWGYnh390wEfrU3nVb0xXjG85qLRU
+Sk8NP0UhUb3RltjTaUwkNKwVKDuwbA0vQMrDAhK8KsTFsoCeYvsMw2xUkeCvKu0hLRk
+6LIZ0u5s3BwPFwwUEEG/yo6B4vEXcshyBG
+lb437kfNfpEWWhATrQmqGtKYVH0Pit8fEdCqM6VQtSGncxYppPz5O3fgFUEsBAh4DCgAAAAgA9DUzWA
0iY8usAAAAAQEAAAgAAAAAAAAAAAAAAAAAPMCAAAAAGluZGV4LnB5UEsFBgAAAAABAAEANGAAANIAA
AAAAA=="
  }
}
```

响应示例

状态码： 200

OK


```
{
  "func_urn" : "urn:fss:xxxxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest",
  "func_name" : "test",
  "domain_id" : "14ee2e3501124efcbca7998baa24xxxx",
  "runtime" : "Node.js6.10",
  "code_type" : "inline",
  "code_filename" : "index.js",
  "code_size" : 245,
  "digest" :
"39d51ae334ffb239037e194159f4cc9cec31086719aa73db7d0a608343121a290f7115693e008bd01cb12b99a8
03856e83aa96173b26c528c41cfed18267e08f",
  "last_modified" : "2019-10-31 11:37:58",
  "func_code" : {
    "file" :
"aW1wb3J0IGpzb24KZGVmIGhhbmRsZXIoZXZlbnQsIGNvbnRleHQpOgogICAgb3V0cHV0ID0gJ0hlbGxvIE1zZyB
mb3IgbW9kaWZ5OjEiAnIcSganNvbi5kdW1wcyhldmVudCkKICAgIGFrID0gY29udGV4dC5nZXRB2Nlc3NLZXkoK
QogICAgc2sgPSBjb250ZXh0LmdldFNlY3JldEtleSgpCiAgICB0b2t1biA9IGNvbnRleHQpZ2V0VG9rZW4oKQogICAg
cHJpbnQgJ2FrOicgKyBhawogICAgcHJpbnQgJ3NrOicgKyBzawogICAgcHJpbnQgJ3Rva2VuOicgKyB0b2t1bGogIC
AgcmV0dXJlG91dHB1dAo+C0NLtLwKlMnCTgBMSn55C+UbkysuKp1Bj7JQqxs+5p7La/ew4RBTp8lusZ/
k2mqmqvCDdoBY0ywYCl
+RgctnLbTS6hLLj9sNyKoVXiXb1BLAQieAwoAAAAIAAHYOU8ssPQsgwAAALYAAAAIAAAAAAAAAAAAAAADzAgA
AAABpbmRleC5weVBLBQYAAAAAAQABADYAAACpAAAAAA="
    "link" : ""
  },
  "strategy_config" : {
    "concurrency" : -1
  }
}
```

状态码： 404

Not Found 找不到资源。

```
{
  "error_code" : "FSS.1052",
  "error_msg" : "Not found the function version"
}
```

SDK 代码示例

SDK代码示例如下。

Java

- 以inline方式修改指定函数代码。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateFunctionCodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");
```

```
ICredential auth = new BasicCredentials()
    .withAk(ak)
    .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
    .withCredential(auth)
    .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateFunctionCodeRequest request = new UpdateFunctionCodeRequest();
UpdateFunctionCodeRequestBody body = new UpdateFunctionCodeRequestBody();
FuncCode funcCodebody = new FuncCode();

funcCodebody.withFile("UESDBAoAAAAIAPQ1M1gNImPLrAAAAEBAaIAAAaW5kZXgucHlnjEogJA
MRd/SigVfxDALxhjDo0S/wB+YrMgMdMvWGYnh390wEfrU3nvn0xXjG85qLRU
+Sk8NP0UhUb3RltjTaUwkNKwVKDuwbA0vQMrDAhK8KsTFsoCeYvsMw2xUkeCvKu0hLRk
+6LIZ0u5s3BwPFwwUEEG/yo6B4vEXcshyBG
+lb437kfNFpEWWhATrQmqGTkYVH0Pit8FEdCqM6VQtSGncxYPpPz5O3fgFUEsBAh4DCgAAAAgA9DUzWA
0iY8usAAAAAQEAAAgAAAAAAPMCAAAAAGluZGV4LnB5UEsFBgAAAAABAAEANGAAANIAA
AAAA==");
body.withFuncCode(funcCodebody);
body.withCodeFilename("index.zip");
body.withCodeType(UpdateFunctionCodeRequestBody.CodeTypeEnum.fromValue("inline"));
request.withBody(body);
try {
    UpdateFunctionCodeResponse response = client.updateFunctionCode(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

- 以zip方式修改指定函数代码。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateFunctionCodeSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before
        // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
        // environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);
```

```
FunctionGraphClient client = FunctionGraphClient.newBuilder()
    .withCredential(auth)
    .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
    .build();
UpdateFunctionCodeRequest request = new UpdateFunctionCodeRequest();
UpdateFunctionCodeRequestBody body = new UpdateFunctionCodeRequestBody();
FuncCode funcCodebody = new FuncCode();

funcCodebody.withFile("UESDBAoAAAAIAPQ1M1gNImPLrAAAAEBAAAIAAAaW5kZXgucHINjtEOgJA
MRd/5igVfxDAIxhjDo0S/wB+YrMgMdMvWGYnh390wEfrU3nvn0xXjG85qLRU
+Sk8NP0UhUb3RltjTaUwkNKwVKDuwbA0vQMrDAhK8KsTFsoCeYvsMw2xUkeCvKu0hLRk
+6LIZ0u5s3BwPFwwUEEG/yo6B4vEXcshyBG
+lb437kfNFpEWhATrQmqGTkYVH0Pit8FEdCqM6VQtSGncxYppPz5O3fgFUEsBAh4DCgAAAAgA9DUzWA
0iY8usAAAAAQEAAAgAAAAAAPMCAAAAGLUZGV4LnB5UEsFBgAAAAABAAEANGAAANIAA
AAAAA==");
body.withFuncCode(funcCodebody);
body.withCodeFilename("index.zip");
body.withCodeType(UpdateFunctionCodeRequestBody.CodeTypeEnum.fromValue("zip"));
request.withBody(body);
try {
    UpdateFunctionCodeResponse response = client.updateFunctionCode(request);
    System.out.println(response.toString());
} catch (ConnectionException e) {
    e.printStackTrace();
} catch (RequestTimeoutException e) {
    e.printStackTrace();
} catch (ServiceResponseException e) {
    e.printStackTrace();
    System.out.println(e.getHttpStatusCode());
    System.out.println(e.getRequestId());
    System.out.println(e.getErrorCode());
    System.out.println(e.getErrorMsg());
}
}
```

Python

- 以inline方式修改指定函数代码。

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateFunctionCodeRequest()
        funcCodebody = FuncCode(
            file="UESDBAoAAAAIAPQ1M1gNImPLrAAAAEBAAAIAAAaW5kZXgucHINjtEOgJAMRd/
5igVfxDAIxhjDo0S/wB+YrMgMdMvWGYnh390wEfrU3nvn0xXjG85qLRU
+Sk8NP0UhUb3RltjTaUwkNKwVKDuwbA0vQMrDAhK8KsTFsoCeYvsMw2xUkeCvKu0hLRk
```

```
+6LIZ0u5s3BwPFwwUEEG/yo6B4vEXcshyBG
+lb437kfNfPEWhATrQmqGTKYVH0Pit8FEEdCqM6VQtSGncxYPpPz5O3fgFUEsBAh4DCgAAAAgA9DUzWA
0iY8usAAAAAQEAAAgAAAAAAAAAAAAAAAAAPMCAAAAAGluZGV4LnB5UEsFBgAAAAABAAEANGAAANIAA
AAAAA=="
)
request.body = UpdateFunctionCodeRequestBody(
    func_code=funcCodebody,
    code_filename="index.zip",
    code_type="inline"
)
response = client.update_function_code(request)
print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

- 以zip方式修改指定函数代码。

```
# coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    environment variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before
    running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
        .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
        .build()

    try:
        request = UpdateFunctionCodeRequest()
        funcCodebody = FuncCode(
            file="UEsDBAoAAAAIAPQ1M1gNImPLrAAAAEBAAAIAAAaW5kZXZgucHINjtEOgjAMRd/
5igVfxDAIxbjDo0S/wB+YrMgMdMvWGYnh390wEfrU3nxb0xJG85qLRU
+Sk8NP0UUhUb3RltjTaUwkNKwVKDuwbA0vQMrDAhK8KSsTFsoCeYvsMw2xUkeCvKu0hLRk
+6LIZ0u5s3BwPFwwUEEG/yo6B4vEXcshyBG
+lb437kfNfPEWhATrQmqGTKYVH0Pit8FEEdCqM6VQtSGncxYPpPz5O3fgFUEsBAh4DCgAAAAgA9DUzWA
0iY8usAAAAAQEAAAgAAAAAAAAAAAAAAAAAPMCAAAAAGluZGV4LnB5UEsFBgAAAAABAAEANGAAANIAA
AAAAA=="
        )
        request.body = UpdateFunctionCodeRequestBody(
            func_code=funcCodebody,
            code_filename="index.zip",
            code_type="zip"
        )
        response = client.update_function_code(request)
        print(response)
    except exceptions.ClientRequestException as e:
        print(e.status_code)
        print(e.request_id)
        print(e.error_code)
        print(e.error_msg)
```

Go

- 以inline方式修改指定函数代码。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateFunctionCodeRequest{
        fileFuncCode:= "UEsDBAoAAAAIAPQ1M1gNImPLrAAAAEBAAAIAAAAaW5kZXgucHlINjtEOgjAMRd/
5igVfxDALxhjDo0S/wB+YrMgMdMvWGYnh390wEfrU3nxb0xXjG85qLRU
+Sk8NP0UhUb3RltjTaUwkNKwVKDuwbA0vQMrDAhK8KsTFsoCeYvsMw2xUkeCvKu0hLRk
+6LIZ0u5s3BwPFfwUEEG/yo6B4vEXcshyBG
+lb437kfNFpEWhATrQmqGtKYVH0Pit8FEdCqM6VQtSGncxYPpPz5O3fgFUEsBAh4DCgAAAAgA9DUzWA
0iY8usAAAAAQEAAAgAAAAAAMCAAAAGluZGV4LnB5UEsFBgAAAAABAAEANGAAANIAA
AAAAA=="
        funcCodebody := &model.FuncCode{
            File: &fileFuncCode,
        }
        codeFilenameUpdateFunctionCodeRequestBody:= "index.zip"
        request.Body = &model.UpdateFunctionCodeRequestBody{
            FuncCode: funcCodebody,
            CodeFilename: &codeFilenameUpdateFunctionCodeRequestBody,
            CodeType: model.GetUpdateFunctionCodeRequestBodyCodeTypeEnum().INLINE,
        }
        response, err := client.UpdateFunctionCode(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}
```

- 以zip方式修改指定函数代码。

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

```

```

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
    // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
    // environment variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before
    // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
    // environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.UpdateFunctionCodeRequest{
        fileFuncCode:= "UEsDBAoAAAAIAPQ1M1gNImlPLrAAAAAEBAAAAIAAAAw5kZXgucHlnjtEOgjAMRd/
5igVfxDALxhjDo0S/wB+YrMgMdMvWGYnh390wEfrU3nvn0xXjG85qLRU
+Sk8NP0UUhUb3RltjTaUwkNKwVKDuwbA0vQMrDAhK8KSsTFsoCeYvsMw2xUkeCvKu0hLRk
+6LIZ0u5s3BwPFwwUEEG/yo6B4vEXcshyBG
+lb437kfNFpEWhATRQmqGtKYVH0Pit8FEEdCqM6VQtSGncxYPpPz5O3fgFUEsBAh4DCgAAAAgA9DUzWA
0iY8usAAAAAQEAAAgAAAAAAAAAAAAAAAAAPMCAAAAAGluZGV4LnB5UEsFBgAAAAABAAEANGAAANIAA
AAAAA=="
        funcCodebody := &model.FuncCode{
            File: &fileFuncCode,
        }
        codeFilenameUpdateFunctionCodeRequestBody:= "index.zip"
        request.Body = &model.UpdateFunctionCodeRequestBody{
            FuncCode: funcCodebody,
            CodeFilename: &codeFilenameUpdateFunctionCodeRequestBody,
            CodeType: model.GetUpdateFunctionCodeRequestBodyCodeTypeEnum().ZIP,
        }
        response, err := client.UpdateFunctionCode(request)
        if err == nil {
            fmt.Printf("%+v\n", response)
        } else {
            fmt.Println(err)
        }
    }
}

```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。

状态码	描述
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.6.6 获取函数的 metadata

功能介绍

获取指定函数的metadata。

调用方法

请参见[如何调用API](#)。

URI

GET /v2/{project_id}/fgs/functions/{function_urn}/config

表 7-232 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见 FunctionGraph函数模型的描述。

请求参数

表 7-233 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

响应参数

状态码： 200

表 7-234 响应 Body 参数

参数	参数类型	描述
func_id	String	函数id, 唯一标识函数。
resource_id	String	资源id。
func_urn	String	函数的URN (Uniform Resource Name), 唯一标识函数。
func_name	String	函数名称。
domain_id	String	域名id。
namespace	String	租户的project id。
project_name	String	租户的project name。
package	String	函数所属的分组Package, 用于用户针对函数的自定义分组。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0
timeout	Integer	<p>函数执行超时时间，超时函数将被强行停止，范围3~259200秒。</p>

参数	参数类型	描述
handler	String	函数执行入口 规则: xx.xx, 必须包含 “.” 举例: 对于node.js函数: myfunction.handler, 则表示函数的文件名为myfunction.js, 执行的入口函数名为handler。长度不超过128个字符。 最小长度: 3 最大长度: 128
memory_size	Integer	函数消耗的内存。单位M。取值范围为: 128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128, 最大值为4096。
gpu_memory	Integer	函数消耗的显存, 只支持自定义运行时与自定义镜像函数配置GPU。单位MB。取值范围为: 1024、2048、3072、4096、5120、6144、7168、8192、9216、10240、11264、12288、13312、14336、15360、16384。最小值为1024, 最大值为16384。
cpu	Integer	函数占用的cpu资源。单位为millicore (1 core=1000 millicores)。取值与MemorySize成比例, 默认是128M内存占0.1个核 (100 millicores)。
code_type	String	函数代码类型, 取值有5种。inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包, 主要针对Java函数。Custom-Image-Swr: 函数代码来源于SWR自定义镜像。 枚举值: <ul style="list-style-type: none"> • inline • zip • obs • jar • Custom-Image-Swr
code_url	String	当code_type为obs时, 该值为函数代码包在OBS上的地址, code_type为其他值时, 该字段为空。
code_filename	String	函数的文件名, 当code_type为jar/zip时必须提供该字段, inline和obs不需要提供。
code_size	Long	函数大小, 单位: 字节。
user_data	String	用户自定义的name/value信息。在函数中使用的参数。举例: 如函数要访问某个主机, 可以设置自定义参数: Host={host_ip}, 最多定义20个, 总长度不超过4KB。

参数	参数类型	描述
encrypted_user_data	String	用户自定义的name/value信息，用于需要加密的配置。
digest	String	函数代码SHA512 hash值，用于判断函数是否变化。
version	String	函数版本号，由系统自动生成，规则：vYYYYMMDD-HHMMSS（v+年月日-时分秒）。
image_name	String	函数版本的内部标识。
xrole	String	函数配置委托。需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。 配置后用户可以通过函数执行入口方法中的context参数获取具有委托中权限的token、ak、sk，用于访问其他云服务。如果用户函数不访问任何云服务，则不用提供委托名称。
app_xrole	String	函数执行委托。可为函数执行单独配置执行委托，这将减小不必要的性能损耗；不单独配置执行委托时，函数执行和函数配置将使用同一委托。
description	String	函数描述。
last_modified	String	函数最后一次更新时间。
ephemeral_storage	Integer	临时存储大小。默认情况下会为函数的/tmp目录分配512MB的空间。您可以通过临时存储设置将函数的/tmp目录大小调整为10G。 枚举值： <ul style="list-style-type: none"> • 512 • 10240
func_vpc	FuncVpc object	函数vpc配置。需同时为函数配置具有vpc权限的委托。
mount_config	MountConfig object	函数挂载配置。
depend_list	Array of strings	依赖id列表。
depend_version_list	Array of strings	依赖版本id列表。
strategy_config	StrategyConfig object	函数策略配置。
dependencies	Array of Dependency objects	函数依赖代码包列表。

参数	参数类型	描述
initializer_handler	String	函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。举例：对于node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。
initializer_timeout	Integer	初始化超时时间，超时函数将被强行停止，范围1~300秒。当配置初始化函数时，此参数必填。
pre_stop_handler	String	函数预停止函数的入口，规则：xx.xx，必须包含“.”。举例：对于node.js函数：myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。
pre_stop_timeout	Integer	初始化超时时间，超时函数将被强行停止，范围1~90秒。
enterprise_project_id	String	企业项目ID，在企业用户创建函数时必填。
long_time	Boolean	是否允许进行长时间超时设置。
log_group_id	String	自定义日志查询组id。
log_stream_id	String	自定义日志查询流id。
type	String	v2表示为正式版本,v1为废弃版本。 枚举值： <ul style="list-style-type: none"> • v1 • v2
enable_cloud_debug	String	适配CloudDebug场景，是否开启云调试（已废弃）。
enable_dynamic_memory	Boolean	是否允许动态内存配置。
is_stateful_function	Boolean	是否支持有状态，如果支持，需要固定传参为true，v2版本支持。
is_bridge_function	Boolean	是否为bridge函数。
enable_auth_in_header	Boolean	是否允许在请求头中添加鉴权信息，只支持自定义镜像函数。
custom_image	CustomImage object	用户容器镜像。
reserved_instance_idle_mode	Boolean	是否开启预留实例闲置模式。

参数	参数类型	描述
apig_route_enable	Boolean	是否配置下沉apig路由规则。
heartbeat_handler	String	心跳函数函数的入口，规则：xx.xx，必须包含“.”，只支持JAVA运行时配置。心跳函数入口需要与函数执行入口在同一文件下。 在开启心跳函数配置时，此参数必填。
enable_class_isolation	Boolean	类隔离开关，只支持JAVA运行时配置。 开启类隔离后可以支持Kafka转储并提升类加载效率，但也可能会导致某些兼容性问题，请谨慎开启。
gpu_type	String	显卡类型。
allow_ephemeral_storage	Boolean	是否支持配置临时存储。
network_controller	NetworkControlConfig object	函数网络配置。
is_return_stream	Boolean	是否返回流式数据（已废弃）。
peering_cidr	String	VPC对等连接网段。 您可以声明代码中使用到的VPC网段，用以检测是否与服务使用VPC网段冲突。网段间使用分号分隔且不能超过5个。
lts_custom_tag	Map<String,String>	自定义日志标签。 函数执行时，可以按照自定义标签配置上报标签到云日志服务(LTS)，用户可以通过标签对日志进行过滤筛选。

表 7-235 FuncVpc

参数	参数类型	描述
domain_id	String	域名id。
namespace	String	租户的project id。
vpc_name	String	虚拟私有云名称。
vpc_id	String	虚拟私有云唯一标识。
subnet_name	String	子网名称。
subnet_id	String	子网编号。
cidr	String	子网掩码。

参数	参数类型	描述
gateway	String	网关。
security_groups	Array of strings	安全组

表 7-236 MountConfig

参数	参数类型	描述
mount_user	MountUser object	挂载用户信息。
func_mounts	Array of FuncMount objects	函数挂载列表。

表 7-237 MountUser

参数	参数类型	描述
user_id	String	用户ID(-1~65534的非0整数)
user_group_id	String	用户组ID(-1~65534的非0整数)

表 7-238 FuncMount

参数	参数类型	描述
mount_type	String	挂载类型(sfs/sfsTurbo/ecs), func_mounts非空时必选。
mount_resource	String	挂载资源ID(对应云服务ID), func_mounts非空时必选。
mount_share_path	String	远端挂载路径(例如192.168.0.12:/data), 如果mount_type为ecs, 必选。
local_mount_path	String	函数访问路径, func_mounts非空时必选。

表 7-239 StrategyConfig

参数	参数类型	描述
concurrency	Integer	单函数最大实例数，v1取值0和-1，v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用
concurrent_num	Integer	单实例并发数，v2版本才支持，取值1到1000

表 7-240 Dependency

参数	参数类型	描述
id	String	依赖包版本ID。
owner	String	依赖包属主的domainId。
link	String	依赖包在OBS上的链接。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> • Java8 • Java11 • Node.js6.10 • Node.js8.10 • Node.js10.16 • Node.js12.13 • Node.js14.18 • Node.js16.17 • Node.js18.15 • Python2.7 • Python3.6 • Python3.10 • Go1.x • C#(.NET Core 2.1) • C#(.NET Core 3.1) • Custom • PHP7.3 • Python3.9 • http • Custom Image • Cangjie1.0
etag	String	依赖包的md5值
size	Long	依赖包大小。

参数	参数类型	描述
name	String	依赖包名称。
description	String	依赖包描述。
file_name	String	依赖包文件名，如果创建方式为zip时。
version	Long	依赖包版本编号。
dep_id	String	依赖包ID
last_modified	Integer	函数最后一次更新时间。

表 7-241 CustomImage

参数	参数类型	描述
enabled	Boolean	是否启用
image	String	镜像地址
command	String	启动容器镜像的命令
args	String	启动容器镜像的命令行参数
working_dir	String	镜像容器工作目录
uid	String	镜像容器的用户id
gid	String	镜像容器的用户组id

表 7-242 NetworkControlConfig

参数	参数类型	描述
disable_public_network	Boolean	禁止公网访问开关。
trigger_access_vpcs	Array of VpcConfig objects	指定触发函数vpc配置。

表 7-243 VpcConfig

参数	参数类型	描述
vpc_name	String	vpc名称。
vpc_id	String	vpc ID。

状态码： 400**表 7-244** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401**表 7-245** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403**表 7-246** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404**表 7-247** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500**表 7-248** 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码

参数	参数类型	描述
error_msg	String	错误信息

请求示例

查询指定函数metadata。

```
GET https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/config
```

响应示例

状态码： 200

OK

```
{
  "func_id" : "0d075978-5a54-4ee1-8e24-ff5bd070xxxx",
  "resource_id" : "0d075978-5a54-4ee1-8e24-ff5bd070xxxx:test",
  "func_urn" : "urn:fss:xxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest",
  "func_name" : "test",
  "domain_id" : "14ee2e3501124efcbca7998baa24xxxx",
  "namespace" : "46b6f338fc3445b8846c71dfb1fbxxxx",
  "project_name" : "xxxxxxxx",
  "package" : "default",
  "runtime" : "Node.js6.10",
  "timeout" : 3,
  "handler" : "test.handler",
  "memory_size" : 128,
  "cpu" : 300,
  "code_type" : "inline",
  "code_filename" : "index.js",
  "code_size" : 272,
  "digest" :
"faa825575c45437cddd4e369bea69893bcbe195d478178462ad90984fe72993f3f59d15f41c5373f807f3e05fb9af322c55dabeb16565c386e402413458e6068",
  "version" : "latest",
  "ephemeral_storage" : 512,
  "image_name" : "latest-191025153727@zehht",
  "last_modified" : "2019-10-25 15:37:27",
  "strategy_config" : {
    "concurrency" : 0
  }
}
```

状态码： 404

Not Found 找不到资源。

```
{
  "error_code" : "FSS.1051",
  "error_msg" : "Not found the function"
}
```

SDK 代码示例

SDK代码示例如下。

Java

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowFunctionConfigSolution {

    public static void main(String[] args) {
        // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
        // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
        // environment variables and decrypted during use to ensure security.
        // In this example, AK and SK are stored in environment variables for authentication. Before running
        // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
        String ak = System.getenv("CLOUD_SDK_AK");
        String sk = System.getenv("CLOUD_SDK_SK");

        ICredential auth = new BasicCredentials()
            .withAk(ak)
            .withSk(sk);

        FunctionGraphClient client = FunctionGraphClient.newBuilder()
            .withCredential(auth)
            .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
            .build();
        ShowFunctionConfigRequest request = new ShowFunctionConfigRequest();
        try {
            ShowFunctionConfigResponse response = client.showFunctionConfig(request);
            System.out.println(response.toString());
        } catch (ConnectionException e) {
            e.printStackTrace();
        } catch (RequestTimeoutException e) {
            e.printStackTrace();
        } catch (ServiceResponseException e) {
            e.printStackTrace();
            System.out.println(e.getHttpStatusCode());
            System.out.println(e.getRequestId());
            System.out.println(e.getErrorCode());
            System.out.println(e.getErrorMsg());
        }
    }
}
```

Python

```
# coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
    # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    # variables and decrypted during use to ensure security.
    # In this example, AK and SK are stored in environment variables for authentication. Before running this
    # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak = __import__('os').getenv("CLOUD_SDK_AK")
    sk = __import__('os').getenv("CLOUD_SDK_SK")

    credentials = BasicCredentials(ak, sk) \

    client = FunctionGraphClient.new_builder() \
        .with_credentials(credentials) \
```

```
.with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
.build()

try:
    request = ShowFunctionConfigRequest()
    response = client.show_function_config(request)
    print(response)
except exceptions.ClientRequestException as e:
    print(e.status_code)
    print(e.request_id)
    print(e.error_code)
    print(e.error_msg)
```

Go

```
package main

import (
    "fmt"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
    functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
    "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
    region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
    // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
    // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
    // variables and decrypted during use to ensure security.
    // In this example, AK and SK are stored in environment variables for authentication. Before running this
    // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
    ak := os.Getenv("CLOUD_SDK_AK")
    sk := os.Getenv("CLOUD_SDK_SK")

    auth := basic.NewCredentialsBuilder().
        WithAk(ak).
        WithSk(sk).
        Build()

    client := functiongraph.NewFunctionGraphClient(
        functiongraph.FunctionGraphClientBuilder().
            WithRegion(region.ValueOf("<YOUR REGION>")).
            WithCredential(auth).
            Build())

    request := &model.ShowFunctionConfigRequest{}
    response, err := client.ShowFunctionConfig(request)
    if err == nil {
        fmt.Printf("%+v\n", response)
    } else {
        fmt.Println(err)
    }
}
```

更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

状态码

状态码	描述
200	OK

状态码	描述
400	Bad Request 请求错误。
401	Unauthorized 鉴权失败。
403	Forbidden 没有操作权限。
404	Not Found 找不到资源。
500	Internal Server Error 服务内部错误。

错误码

请参见[错误码](#)。

7.6.7 修改函数的 metadata 信息

功能介绍

修改指定的函数的metadata信息。

调用方法

请参见[如何调用API](#)。

URI

PUT /v2/{project_id}/fgs/functions/{function_urn}/config

表 7-249 路径参数

参数	是否必选	参数类型	描述
project_id	是	String	租户项目 ID，获取方式请参见 获取项目ID 。
function_urn	是	String	函数的URN，详细解释见 FunctionGraph函数模型的描述。

请求参数

表 7-250 请求 Header 参数

参数	是否必选	参数类型	描述
X-Auth-Token	是	String	用户Token。 通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。
Content-Type	是	String	消息体的类型（格式）

表 7-251 请求 Body 参数

参数	是否必选	参数类型	描述
func_name	是	String	函数名称。

参数	是否必选	参数类型	描述
runtime	是	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9

参数	是否必选	参数类型	描述
			<ul style="list-style-type: none"> • http • Custom Image • Cangjie1.0
timeout	是	Integer	函数执行超时时间，超时函数将被强行停止，范围3~259200秒。
handler	是	String	函数执行入口规则：xx.xx，必须包含“.” 举例：对于node.js函数：myfunction.handler，则表示函数的文件名为myfunction.js，执行的入口函数名为handler。长度不超过128个字符。 最小长度：3 最大长度：128
memory_size	是	Integer	函数消耗的内存。单位M。取值范围为：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128，最大值为4096。
gpu_memory	否	Integer	函数消耗的显存，只支持自定义运行时与自定义镜像函数配置GPU。单位MB。取值范围为：1024、2048、3072、4096、5120、6144、7168、8192、9216、10240、11264、12288、13312、14336、15360、16384。最小值为1024，最大值为16384。（目前只支持华东-上海一）
gpu_type	否	String	显卡类型。（目前只支持华东-上海一）
user_data	否	String	用户自定义的name/value信息。在函数中使用的参数。举例：如函数要访问某个主机，可以设置自定义参数：Host={host_ip}，最多定义20个，总长度不超过4KB。
encrypted_user_data	否	String	用户自定义的name/value信息，用于需要加密的配置。

参数	是否必选	参数类型	描述
xrole	否	String	函数配置委托。需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。配置后用户可以通过函数执行入口方法中的context参数获取具有委托中权限的token、ak、sk，用于访问其他云服务。如果用户函数不访问任何云服务，则不用提供委托名称。
app_xrole	否	String	函数执行委托。可为函数执行单独配置执行委托，这将减小不必要的性能损耗；不单独配置执行委托时，函数执行和函数配置将使用同一委托。
description	否	String	函数描述。
func_vpc	否	FuncVpc object	函数vpc配置。需同时为函数配置具有vpc权限的委托。
mount_config	否	MountConfig object	函数挂载配置。
strategy_config	否	StrategyConfig object	函数策略配置。
custom_image	否	CustomImage object	用户容器镜像。
extend_config	否	String	函数扩展配置。
initializer_handler	否	String	函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。举例：对于node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。
initializer_timeout	否	Integer	初始化超时时间，超时函数将被强行停止，范围1~300秒。当配置初始化函数时，此参数必填。

参数	是否必选	参数类型	描述
pre_stop_handler	否	String	函数预停止函数的入口，规则：xx.xx，必须包含“.”。举例：对于node.js函数：myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。
pre_stop_timeout	否	Integer	初始化超时时间，超时函数将被强行停止，范围1~90秒。
ephemeral_storage	否	Integer	临时存储大小。默认情况下会为函数的/tmp目录分配512MB的空间。您可以通过临时存储设置将函数的/tmp目录大小调整为10G。 枚举值： <ul style="list-style-type: none"> • 512 • 10240
enterprise_project_id	否	String	企业项目ID，在企业用户创建函数时必填。
log_config	否	FuncLogConfig object	函数绑定日志配置。
network_controller	否	NetworkControlConfig object	函数网络配置。
is_stateful_function	否	Boolean	是否支持有状态，如果需要支持，需要固定传参为true，v2版本支持
enable_dynamic_memory	否	Boolean	是否启动动态内存配置
enable_auth_in_header	否	Boolean	是否允许在请求头中添加鉴权信息
domain_names	否	String	内网域名配置。
restore_hook_handler	否	String	函数快照式冷启动Restore Hook入口，仅支持Java，规则：xx.xx，必须包含“.”。如：com.xxx.demo.Test.restoreHook
restore_hook_timeout	否	Integer	快照冷启动Restore Hook的超时时间，超时函数将被强行停止，范围1~300秒。

参数	是否必选	参数类型	描述
heartbeat_handler	否	String	心跳函数函数的入口，规则：xx.xx，必须包含“.”，只支持JAVA运行时配置。心跳函数入口需要与函数执行入口在同一文件下。 在开启心跳函数配置时，此参数必填。
enable_class_isolation	否	Boolean	类隔离开关，只支持JAVA运行时配置。 开启类隔离后可以支持Kafka转储并提升类加载效率，但也可能会导致某些兼容性问题，请谨慎开启。
peering_cidr	否	String	VPC对等连接网段。 您可以声明代码中使用到的VPC网段，用以检测是否与服务使用VPC网段冲突。网段间使用分号分隔且不能超过5个。
lts_custom_tag	否	Map<String,String>	自定义日志标签。 函数执行时，可以按照自定义标签配置上报标签到云日志服务(LTS)，用户可以通过标签对日志进行过滤筛选。

表 7-252 FuncVpc

参数	是否必选	参数类型	描述
domain_id	否	String	域名id。
namespace	否	String	租户的project id。
vpc_name	否	String	虚拟私有云名称。
vpc_id	是	String	虚拟私有云唯一标识。
subnet_name	否	String	子网名称。
subnet_id	是	String	子网编号。
cidr	否	String	子网掩码。
gateway	否	String	网关。
security_groups	否	Array of strings	安全组

表 7-253 MountConfig

参数	是否必选	参数类型	描述
mount_user	是	MountUser object	挂载用户信息。
func_mounts	是	Array of FuncMount objects	函数挂载列表。

表 7-254 MountUser

参数	是否必选	参数类型	描述
user_id	是	String	用户ID(-1~65534的非0整数)
user_group_id	是	String	用户组ID(-1~65534的非0整数)

表 7-255 FuncMount

参数	是否必选	参数类型	描述
mount_type	是	String	挂载类型(sfs/sfsTurbo/ecs), func_mounts非空时必选。
mount_resource	是	String	挂载资源ID (对应云服务ID), func_mounts非空时必选。
mount_share_path	否	String	远端挂载路径 (例如 192.168.0.12:/data), 如果 mount_type为ecs, 必选。
local_mount_path	是	String	函数访问路径, func_mounts非空时必选。

表 7-256 StrategyConfig

参数	是否必选	参数类型	描述
concurrency	是	Integer	单函数最大实例数, v1取值0和-1, v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用
concurrent_num	是	Integer	单实例并发数, v2版本才支持, 取值1到1000

表 7-257 CustomImage

参数	是否必选	参数类型	描述
enabled	否	Boolean	是否启用
image	否	String	镜像地址
command	否	String	启动容器镜像的命令
args	否	String	启动容器镜像的命令行参数
working_dir	否	String	镜像容器工作目录
uid	否	String	镜像容器的用户id
gid	否	String	镜像容器的用户组id

表 7-258 FuncLogConfig

参数	是否必选	参数类型	描述
group_name	否	String	函数绑定日志组名。
group_id	否	String	函数绑定日志组ID。
stream_name	否	String	函数绑定日志流名。
stream_id	否	String	函数绑定日志流ID。

表 7-259 NetworkControlConfig

参数	是否必选	参数类型	描述
disable_public_network	否	Boolean	禁止公网访问开关。
trigger_access_vpcs	否	Array of VpcConfig objects	指定触发函数vpc配置。

表 7-260 VpcConfig

参数	是否必选	参数类型	描述
vpc_name	否	String	vpc名称。
vpc_id	否	String	vpc ID。

响应参数

状态码： 200

表 7-261 响应 Body 参数

参数	参数类型	描述
func_id	String	函数id, 唯一标识函数。
resource_id	String	函数资源id。
func_urn	String	函数的URN (Uniform Resource Name), 唯一标识函数。
func_name	String	函数名称。
domain_id	String	域名id。
namespace	String	租户的project id。
project_name	String	租户的project name。
package	String	函数所属的分组Package, 用于用户针对函数的自定义分组。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0
timeout	Integer	<p>函数执行超时时间，超时函数将被强行停止，范围3~259200秒。</p>

参数	参数类型	描述
handler	String	函数执行入口 规则：xx.xx，必须包含“.” 举例：对于node.js函数：myfunction.handler，则表示函数的文件名为myfunction.js，执行的入口函数名为handler。长度不超过128个字符。 最小长度：3 最大长度：128
memory_size	Integer	函数消耗的内存。单位M。取值范围为：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128，最大值为4096。
gpu_memory	Integer	函数消耗的显存，只支持自定义运行时与自定义镜像函数配置GPU。单位MB。取值范围为：1024、2048、3072、4096、5120、6144、7168、8192、9216、10240、11264、12288、13312、14336、15360、16384。最小值为1024，最大值为16384。
cpu	Integer	函数占用的cpu资源。单位为millicore（1 core=1000 millicores）。取值与MemorySize成比例，默认是128M内存占0.1个核（100 millicores）。
code_type	String	函数代码类型，取值有5种。inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包，主要针对Java函数。Custom-Image-Swr: 函数代码来源于SWR自定义镜像。 枚举值： <ul style="list-style-type: none"> • inline • zip • obs • jar • Custom-Image-Swr
code_url	String	当code_type为obs时，该值为函数代码包在OBS上的地址，code_type为其他值时，该字段为空。
code_filename	String	函数的文件名，当code_type为jar/zip时必须提供该字段，inline和obs不需要提供。
code_size	Long	函数大小，单位：字节。
user_data	String	用户自定义的name/value信息。在函数中使用的参数。举例：如函数要访问某个主机，可以设置自定义参数：Host={host_ip}，最多定义20个，总长度不超过4KB。

参数	参数类型	描述
encrypted_use_r_data	String	用户自定义的name/value信息，用于需要加密的配置。
digest	String	函数代码SHA512 hash值，用于判断函数是否变化。
version	String	函数版本号，由系统自动生成，规则：vYYYYMMDD-HHMMSS（v+年月日-时分秒）。
image_name	String	函数版本的内部标识。
xrole	String	函数配置委托。需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。配置后用户可以通过函数执行入口方法中的context参数获取具有委托中权限的token、ak、sk，用于访问其他云服务。如果用户函数不访问任何云服务，则不用提供委托名称。
app_xrole	String	函数执行委托。可为函数执行单独配置执行委托，这将减小不必要的性能损耗；不单独配置执行委托时，函数执行和函数配置将使用同一委托。
description	String	函数描述。
last_modified	String	函数最后一次更新时间。
ephemeral_storage	Integer	临时存储大小。默认情况下会为函数的/tmp目录分配512MB的空间。您可以通过临时存储设置将函数的/tmp目录大小调整为10G。 枚举值： <ul style="list-style-type: none"> • 512 • 10240
func_vpc	FuncVpc object	函数vpc配置。需同时为函数配置具有vpc权限的委托。
mount_config	MountConfig object	函数挂载配置。
strategy_config	StrategyConfig object	函数策略配置。
dependencies	Array of Dependency objects	函数依赖代码包列表。
initializer_handler	String	函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。举例：对于node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。

参数	参数类型	描述
initializer_timeout	Integer	初始化超时时间，超时函数将被强行停止，范围1~300秒。当配置初始化函数时，此参数必填。
pre_stop_handler	String	函数预停止函数的入口，规则：xx.xx，必须包含“.”。举例：对于node.js函数：myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。
pre_stop_timeout	Integer	初始化超时时间，超时函数将被强行停止，范围1~90秒。
enterprise_project_id	String	企业项目ID，在企业用户创建函数时必填。
long_time	Boolean	是否允许进行长时间超时设置。
log_group_id	String	自定义日志查询组id。
log_stream_id	String	自定义日志查询流id。
type	String	v2表示为正式版本,v1为废弃版本。 枚举值： <ul style="list-style-type: none"> • v1 • v2
enable_cloud_debug	String	适配CloudDebug场景，是否开启云调试（已废弃）。
enable_dynamic_memory	Boolean	是否启动动态内存配置。
enable_auth_header	Boolean	是否允许在请求头中添加鉴权信息，只支持自定义镜像函数。
is_stateful_function	Boolean	是否支持有状态，v2版本支持。
domain_names	String	函数配置的需要支持域名解析的内网域名。
enable_auth_header	Boolean	是否允许在请求头中添加鉴权信息。
custom_image	CustomImage object	用户容器镜像。
is_return_stream	Boolean	是否返回流式数据（已废弃）。
peering_cidr	String	VPC对等连接网段。 您可以声明代码中使用到的VPC网段，用以检测是否与服务使用VPC网段冲突。网段间使用分号分隔且不能超过5个。

参数	参数类型	描述
lts_custom_tag	Map<String,String>	自定义日志标签。 函数执行时，可以按照自定义标签配置上报标签到云日志服务(LTS)，用户可以通过标签对日志进行过滤筛选。

表 7-262 FuncVpc

参数	参数类型	描述
domain_id	String	域名id。
namespace	String	租户的project id。
vpc_name	String	虚拟私有云名称。
vpc_id	String	虚拟私有云唯一标识。
subnet_name	String	子网名称。
subnet_id	String	子网编号。
cidr	String	子网掩码。
gateway	String	网关。
security_groups	Array of strings	安全组

表 7-263 MountConfig

参数	参数类型	描述
mount_user	MountUser object	挂载用户信息。
func_mounts	Array of FuncMount objects	函数挂载列表。

表 7-264 MountUser

参数	参数类型	描述
user_id	String	用户ID(-1~65534的非0整数)
user_group_id	String	用户组ID(-1~65534的非0整数)

表 7-265 FuncMount

参数	参数类型	描述
mount_type	String	挂载类型(sfs/sfsTurbo/ecs), func_mounts非空时必选。
mount_resource	String	挂载资源ID (对应云服务ID), func_mounts非空时必选。
mount_share_path	String	远端挂载路径 (例如192.168.0.12:/data), 如果mount_type为ecs, 必选。
local_mount_path	String	函数访问路径, func_mounts非空时必选。

表 7-266 StrategyConfig

参数	参数类型	描述
concurrency	Integer	单函数最大实例数, v1取值0和-1, v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用
concurrent_num	Integer	单实例并发数, v2版本才支持, 取值1到1000

表 7-267 Dependency

参数	参数类型	描述
id	String	依赖包版本ID。
owner	String	依赖包属主的domainId。
link	String	依赖包在OBS上的链接。

参数	参数类型	描述
runtime	String	<p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> ● Java8 ● Java11 ● Node.js6.10 ● Node.js8.10 ● Node.js10.16 ● Node.js12.13 ● Node.js14.18 ● Node.js16.17 ● Node.js18.15 ● Python2.7 ● Python3.6 ● Python3.10 ● Go1.x ● C#(.NET Core 2.1) ● C#(.NET Core 3.1) ● Custom ● PHP7.3 ● Python3.9 ● http ● Custom Image ● Cangjie1.0
etag	String	依赖包的md5值
size	Long	依赖包大小。

参数	参数类型	描述
name	String	依赖包名称。
description	String	依赖包描述。
file_name	String	依赖包文件名，如果创建方式为zip时。
version	Long	依赖包版本编号。
dep_id	String	依赖包ID
last_modified	Integer	函数最后一次更新时间。

表 7-268 CustomImage

参数	参数类型	描述
enabled	Boolean	是否启用
image	String	镜像地址
command	String	启动容器镜像的命令
args	String	启动容器镜像的命令行参数
working_dir	String	镜像容器工作目录
uid	String	镜像容器的用户id
gid	String	镜像容器的用户组id

状态码： 400

表 7-269 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 401

表 7-270 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 403

表 7-271 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 404

表 7-272 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

状态码： 500

表 7-273 响应 Body 参数

参数	参数类型	描述
error_code	String	错误码
error_msg	String	错误信息

请求示例

- 修改名为fgsTest的Nodejs函数配置信息，将内存大小改为768MB，超时时间设为40s，添加环境变量host，添加磁盘挂载。

PUT https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/config

```
{
  "func_name": "fgsTest",
  "handler": "index.handler",
  "memory_size": 768,
  "user_data": "{\"host\":\"192.168.0.1\"}",
  "runtime": "Node.js6.10",
  "mount_config": {
    "mount_user": {
      "user_id": -1,
      "user_group_id": -1
    },
    "func_mounts": [ {
      "mount_resource": "268xxx-d3xx-4cxx-98xx-d1dbxxx",
      "local_mount_path": "/mnt",
      "mount_type": "sfsTurbo"
    } ]
  }
}
```


- ```
"timeout" : 40
}
```
- 修改自定义运行时函数GPU配置，将GPU显存设置为1GB，内存大小设置为4GB。  
PUT [https://{Endpoint}/v2/{project\\_id}/fgs/functions/{function\\_urn}/config](https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/config)

```
{
 "func_name" : "test_GPU",
 "handler" : "bootstrap",
 "memory_size" : 4096,
 "runtime" : "Custom",
 "timeout" : 40,
 "gpu_memory" : 1024
}
```
  - 修改自定义镜像类型函数的镜像配置。

```
{
 "func_name": "test_function",
 "handler": "-",
 "memory_size": 128,
 "runtime": "Custom Image",
 "timeout": 3,
 "xrole": "xxx",
 "custom_image": {
 "enabled": true,
 "image": "swr.xxx.example.com/xxx/xxx:xxx",
 "working_dir": "/"
 }
}
```

## 响应示例

状态码： 200

OK

```
{
 "func_id" : "0d075978-5a54-4ee1-8e24-ff5bd070xxxx",
 "resource_id" : "0d075978-5a54-4ee1-8e24-ff5bd070xxxx:test",
 "func_urn" : "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",
 "func_name" : "test",
 "domain_id" : "14ee2e3501124efcbca7998baa24xxxx",
 "namespace" : "46b6f338fc3445b8846c71dfb1fbxxxx",
 "project_name" : "xxxxxx",
 "package" : "default",
 "runtime" : "Node.js6.10",
 "timeout" : 3,
 "handler" : "fssExampleCsharp2.1::fssExampleCsharp21.Program::MyFunc",
 "memory_size" : 128,
 "cpu" : 300,
 "code_type" : "inline",
 "code_filename" : "index.js",
 "code_size" : 272,
 "user_data" : "{\"host\":\"192.168.0.1\"}",
 "mount_config" : {
 "mount_user" : {
 "user_id" : -1,
 "user_group_id" : -1
 }
 },
 "func_mounts" : [[
 "mount_resource" : "268xxx-d3xx-4cxx-98xx-d1dbxxx",
 "local_mount_path" : "/mnt",
 "mount_type" : "sfsTurbo"
]]
},
 "digest" :
"faa825575c45437cddd4e369bea69893bcbe195d478178462ad90984fe72993f3f59d15f41c5373f807f3e05fb9af322c55dabeb16565c386e402413458e6068",
 "version" : "latest",
```

```
"ephemeral_storage" : 512,
"image_name" : "latest-191025153727@zehlht",
"last_modified" : "2019-10-25 15:37:27",
"strategy_config" : {
 "concurrency" : 0
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 修改名为fgsTest的Nodejs函数配置信息，将内存大小改为768MB，超时时间设为40s，添加环境变量host，添加磁盘挂载。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateFunctionConfigSolution {
 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 UpdateFunctionConfigRequest request = new UpdateFunctionConfigRequest();
 UpdateFunctionConfigRequestBody body = new UpdateFunctionConfigRequestBody();
 List<FuncMount> listMountConfigFuncMounts = new ArrayList<>();
 listMountConfigFuncMounts.add(
 new FuncMount()
 .withMountType("sfsTurbo")
 .withMountResource("268xxx-d3xx-4cxx-98xx-d1dbxxx")
 .withLocalMountPath("/mnt")
);
 MountUser mountUserMountConfig = new MountUser();
 mountUserMountConfig.withUserId("-1")
 .withUserGroupid("-1");
 MountConfig mountConfigbody = new MountConfig();
 mountConfigbody.withMountUser(mountUserMountConfig)
 .withFuncMounts(listMountConfigFuncMounts);
 body.withMountConfig(mountConfigbody);
 body.withUserData("{\"host\":\"192.168.0.1\"}");
 }
}
```

```
body.withMemorySize(768);
body.withHandler("index.handler");
body.withTimeout(40);
body.withRuntime(UpdateFunctionConfigRequestBody.RuntimeEnum.fromValue("Node.js6.10"));
body.withFuncName("fgsTest");
request.withBody(body);
try {
 UpdateFunctionConfigResponse response = client.updateFunctionConfig(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

- 修改自定义运行时函数GPU配置，将GPU显存设置为1GB，内存大小设置为4GB。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateFunctionConfigSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 UpdateFunctionConfigRequest request = new UpdateFunctionConfigRequest();
 UpdateFunctionConfigRequestBody body = new UpdateFunctionConfigRequestBody();
 body.withGpuMemory(1024);
 body.withMemorySize(4096);
 body.withHandler("bootstrap");
 body.withTimeout(40);
 body.withRuntime(UpdateFunctionConfigRequestBody.RuntimeEnum.fromValue("Custom"));
 body.withFuncName("test_GPU");
 request.withBody(body);
 try {
 UpdateFunctionConfigResponse response = client.updateFunctionConfig(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 }
 }
}
```

```
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

- 修改名为fgsTest的Nodejs函数配置信息，将内存大小改为768MB，超时时间设为40s，添加环境变量host，添加磁盘挂载。

```
coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *
```

```
if __name__ == "__main__":
```

```
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 # environment variables and decrypted during use to ensure security.
```

```
 # In this example, AK and SK are stored in environment variables for authentication. Before
 # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 # environment
```

```
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")
```

```
 credentials = BasicCredentials(ak, sk) \
```

```
 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()
```

```
try:
```

```
 request = UpdateFunctionConfigRequest()
 listFuncMountsMountConfig = [
 FuncMount(
 mount_type="sfsTurbo",
 mount_resource="268xxx-d3xx-4cxx-98xx-d1dbxxx",
 local_mount_path="/mnt"
)
]
```

```
 mountUserMountConfig = MountUser(
 user_id="-1",
 user_group_id="-1"
)
```

```
 mountConfigbody = MountConfig(
 mount_user=mountUserMountConfig,
 func_mounts=listFuncMountsMountConfig
)
```

```
 request.body = UpdateFunctionConfigRequestBody(
 mount_config=mountConfigbody,
 user_data="{\"host\":\"192.168.0.1\"}",
 memory_size=768,
 handler="index.handler",
 timeout=40,
 runtime="Node.js6.10",
 func_name="fgsTest"
)
```

```
 response = client.update_function_config(request)
 print(response)
```

```
except exceptions.ClientRequestException as e:
```

```
print(e.status_code)
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

- 修改自定义运行时函数GPU配置，将GPU显存设置为1GB，内存大小设置为4GB。  
# coding: utf-8

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 # environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 # environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = UpdateFunctionConfigRequest()
 request.body = UpdateFunctionConfigRequestBody(
 gpu_memory=1024,
 memory_size=4096,
 handler="bootstrap",
 timeout=40,
 runtime="Custom",
 func_name="test_GPU"
)
 response = client.update_function_config(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

- 修改名为fgsTest的Nodejs函数配置信息，将内存大小改为768MB，超时时间设为40s，添加环境变量host，添加磁盘挂载。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
```

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.UpdateFunctionConfigRequest{}
var listFuncMountsMountConfig = []model.FuncMount{
 {
 MountType: "sfsTurbo",
 MountResource: "268xxx-d3xx-4cxx-98xx-d1dbxxx",
 LocalMountPath: "/mnt",
 },
}
mountUserMountConfig := &model.MountUser{
 UserId: "-1",
 UserGroupId: "-1",
}
mountConfigbody := &model.MountConfig{
 MountUser: mountUserMountConfig,
 FuncMounts: listFuncMountsMountConfig,
}
userDataUpdateFunctionConfigRequestBody := "{\"host\":\"192.168.0.1\"}"
request.Body = &model.UpdateFunctionConfigRequestBody{
 MountConfig: mountConfigbody,
 UserData: &userDataUpdateFunctionConfigRequestBody,
 MemorySize: int32(768),
 Handler: "index.handler",
 Timeout: int32(40),
 Runtime: model.GetUpdateFunctionConfigRequestBodyRuntimeEnum().NODE_JS6_10,
 FuncName: "fgsTest",
}
response, err := client.UpdateFunctionConfig(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

- 修改自定义运行时函数GPU配置，将GPU显存设置为1GB，内存大小设置为4GB。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
```

```
WithAk(ak).
WithSk(sk).
Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.UpdateFunctionConfigRequest{
 gpuMemoryUpdateFunctionConfigRequestBody:= int32(1024)
 request.Body = &model.UpdateFunctionConfigRequestBody{
 GpuMemory: &gpuMemoryUpdateFunctionConfigRequestBody,
 MemorySize: int32(4096),
 Handler: "bootstrap",
 Timeout: int32(40),
 Runtime: model.GetUpdateFunctionConfigRequestBodyRuntimeEnum().CUSTOM,
 FuncName: "test_GPU",
 }
}
response, err := client.UpdateFunctionConfig(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.6.8 更新函数最大实例数

### 功能介绍

更新函数最大实例数

## 调用方法

请参见[如何调用API](#)。

## URI

PUT /v2/{project\_id}/fgs/functions/{function\_urn}/config-max-instance

表 7-274 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见 FunctionGraph函数模型的描述。       |

## 请求参数

表 7-275 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                            |
|--------------|------|--------|---------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                    |

表 7-276 请求 Body 参数

| 参数               | 是否必选 | 参数类型    | 描述                            |
|------------------|------|---------|-------------------------------|
| max_instance_num | 否    | Integer | 最大实例数；-1代表该函数实例数无限制，0代表该函数被禁用 |

## 响应参数

状态码： 200



表 7-277 响应 Body 参数

| 参数           | 参数类型   | 描述                                         |
|--------------|--------|--------------------------------------------|
| func_urn     | String | 函数的URN ( Uniform Resource Name ) , 唯一标识函数。 |
| func_name    | String | 函数名称。                                      |
| domain_id    | String | 域名id。                                      |
| namespace    | String | 租户的project id。                             |
| project_name | String | 租户的project name。                           |
| package      | String | 函数所属的分组Package, 用于用户针对函数的自定义分组。            |

| 参数      | 参数类型    | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| runtime | String  | <p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>Java8</b></li> <li>● <b>Java11</b></li> <li>● <b>Node.js6.10</b></li> <li>● <b>Node.js8.10</b></li> <li>● <b>Node.js10.16</b></li> <li>● <b>Node.js12.13</b></li> <li>● <b>Node.js14.18</b></li> <li>● <b>Node.js16.17</b></li> <li>● <b>Node.js18.15</b></li> <li>● <b>Python2.7</b></li> <li>● <b>Python3.6</b></li> <li>● <b>Python3.10</b></li> <li>● <b>Go1.x</b></li> <li>● <b>C#(.NET Core 2.1)</b></li> <li>● <b>C#(.NET Core 3.1)</b></li> <li>● <b>Custom</b></li> <li>● <b>PHP7.3</b></li> <li>● <b>Python3.9</b></li> <li>● <b>http</b></li> <li>● <b>Custom Image</b></li> <li>● <b>Cangjie1.0</b></li> </ul> |
| timeout | Integer | <p>函数执行超时时间，超时函数将被强行停止，范围3~259200秒。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| 参数                  | 参数类型    | 描述                                                                                                                                                                                                                                                                         |
|---------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| handler             | String  | 函数执行入口 规则：xx.xx，必须包含“.” 举例：对于node.js函数：myfunction.handler，则表示函数的文件名为myfunction.js，执行的入口函数名为handler。长度不超过128个字符。<br>最小长度：3<br>最大长度：128                                                                                                                                      |
| memory_size         | Integer | 函数消耗的内存。单位M。取值范围为：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128，最大值为4096。                                                                                                                                                                           |
| cpu                 | Integer | 函数占用的cpu资源。单位为millicore（1 core=1000 millicores）。取值与MemorySize成比例，默认是128M内存占0.1个核（100 millicores）。                                                                                                                                                                          |
| code_type           | String  | 函数代码类型，取值有5种。inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包，主要针对Java函数。Custom-Image-Swr: 函数代码来源于SWR自定义镜像。<br>枚举值：<br><ul style="list-style-type: none"> <li>• inline</li> <li>• zip</li> <li>• obs</li> <li>• jar</li> <li>• Custom-Image-Swr</li> </ul> |
| code_url            | String  | 当code_type为obs时，该值为函数代码包在OBS上的地址，code_type为其他值时，该字段为空。                                                                                                                                                                                                                     |
| code_filename       | String  | 函数的文件名，当code_type为jar/zip时必须提供该字段，inline和obs不需要提供。                                                                                                                                                                                                                         |
| code_size           | Long    | 函数大小，单位：字节。                                                                                                                                                                                                                                                                |
| user_data           | String  | 用户自定义的name/value信息。在函数中使用的参数。举例：如函数要访问某个主机，可以设置自定义参数：Host={host_ip}，最多定义20个，总长度不超过4KB。                                                                                                                                                                                     |
| encrypted_user_data | String  | 用户自定义的name/value信息，用于需要加密的配置。                                                                                                                                                                                                                                              |
| digest              | String  | 函数代码SHA512 hash值，用于判断函数是否变化。                                                                                                                                                                                                                                               |
| version             | String  | 函数版本号，由系统自动生成，规则：vYYYYMMDD-HHMMSS（v+年月日-时分秒）。                                                                                                                                                                                                                              |

| 参数                    | 参数类型                               | 描述                                                                                                                                    |
|-----------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| image_name            | String                             | 函数版本的内部标识。                                                                                                                            |
| xrole                 | String                             | 函数配置委托。需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。配置后用户可以通过函数执行入口方法中的context参数获取具有委托中权限的token、ak、sk，用于访问其他云服务。如果用户函数不访问任何云服务，则不用提供委托名称。 |
| app_xrole             | String                             | 函数执行委托。可为函数执行单独配置执行委托，这将减小不必要的性能损耗；不单独配置执行委托时，函数执行和函数配置将使用同一委托。                                                                       |
| description           | String                             | 函数描述。                                                                                                                                 |
| last_modified         | String                             | 函数最后一次更新时间。                                                                                                                           |
| func_vpc              | <b>FuncVpc</b> object              | 函数vpc配置。需同时为函数配置具有vpc权限的委托。                                                                                                           |
| mount_config          | <b>MountConfig</b> object          | 函数挂载配置。                                                                                                                               |
| strategy_config       | <b>StrategyConfig</b> object       | 函数策略配置。                                                                                                                               |
| dependencies          | Array of <b>Dependency</b> objects | 函数依赖代码包列表。                                                                                                                            |
| initializer_handler   | String                             | 函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。举例：对于node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。         |
| initializer_timeout   | Integer                            | 初始化超时时间，超时函数将被强行停止，范围1~300秒。当配置初始化函数时，此参数必填。                                                                                          |
| pre_stop_handler      | String                             | 函数预停止函数的入口，规则：xx.xx，必须包含“.”。举例：对于node.js函数：myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。            |
| pre_stop_timeout      | Integer                            | 初始化超时时间，超时函数将被强行停止，范围1~90秒。                                                                                                           |
| enterprise_project_id | String                             | 企业项目ID，在企业用户创建函数时必填。                                                                                                                  |
| long_time             | Boolean                            | 是否允许进行长时间超时设置。                                                                                                                        |
| log_group_id          | String                             | 自定义日志查询组id                                                                                                                            |

| 参数                    | 参数类型    | 描述                                                                                                 |
|-----------------------|---------|----------------------------------------------------------------------------------------------------|
| log_stream_id         | String  | 自定义日志查询流id                                                                                         |
| type                  | String  | v2表示为正式版本,v1为废弃版本。<br>枚举值：<br><ul style="list-style-type: none"> <li>• v1</li> <li>• v2</li> </ul> |
| enable_cloud_debug    | String  | 适配CloudDebug场景，是否开启云调试（已废弃）                                                                        |
| enable_dynamic_memory | Boolean | 是否启动动态内存配置                                                                                         |
| is_stateful_function  | Boolean | 是否支持有状态，v2版本支持                                                                                     |
| domain_names          | String  | 函数配置的需要支持域名解析的内网域名。                                                                                |
| is_return_stream      | Boolean | 是否返回流式数据（已废弃）                                                                                      |
| enable_auth_header    | Boolean | 是否允许在请求头中添加鉴权信息，只支持自定义镜像函数                                                                         |

表 7-278 FuncVpc

| 参数              | 参数类型             | 描述             |
|-----------------|------------------|----------------|
| domain_id       | String           | 域名id。          |
| namespace       | String           | 租户的project id。 |
| vpc_name        | String           | 虚拟私有云名称。       |
| vpc_id          | String           | 虚拟私有云唯一标识。     |
| subnet_name     | String           | 子网名称。          |
| subnet_id       | String           | 子网编号。          |
| cidr            | String           | 子网掩码。          |
| gateway         | String           | 网关。            |
| security_groups | Array of strings | 安全组            |

表 7-279 MountConfig

| 参数          | 参数类型                                    | 描述      |
|-------------|-----------------------------------------|---------|
| mount_user  | <b>MountUser</b><br>object              | 挂载用户信息。 |
| func_mounts | Array of<br><b>FuncMount</b><br>objects | 函数挂载列表。 |

表 7-280 MountUser

| 参数            | 参数类型   | 描述                   |
|---------------|--------|----------------------|
| user_id       | String | 用户ID(-1~65534的非0整数)  |
| user_group_id | String | 用户组ID(-1~65534的非0整数) |

表 7-281 FuncMount

| 参数               | 参数类型   | 描述                                                  |
|------------------|--------|-----------------------------------------------------|
| mount_type       | String | 挂载类型(sfs/sfsTurbo/ecs), func_mounts非空时必选。           |
| mount_resource   | String | 挂载资源ID(对应云服务ID), func_mounts非空时必选。                  |
| mount_share_path | String | 远端挂载路径(例如192.168.0.12:/data), 如果mount_type为ecs, 必选。 |
| local_mount_path | String | 函数访问路径, func_mounts非空时必选。                           |

表 7-282 StrategyConfig

| 参数             | 参数类型    | 描述                                                      |
|----------------|---------|---------------------------------------------------------|
| concurrency    | Integer | 单函数最大实例数, v1取值0和-1, v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用 |
| concurrent_num | Integer | 单实例并发数, v2版本才支持, 取值1到1000                               |

表 7-283 Dependency

| 参数    | 参数类型   | 描述              |
|-------|--------|-----------------|
| id    | String | 依赖包版本ID。        |
| owner | String | 依赖包属主的domainId。 |
| link  | String | 依赖包在OBS上的链接。    |

| 参数      | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| runtime | String | <p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>Java8</b></li> <li>● <b>Java11</b></li> <li>● <b>Node.js6.10</b></li> <li>● <b>Node.js8.10</b></li> <li>● <b>Node.js10.16</b></li> <li>● <b>Node.js12.13</b></li> <li>● <b>Node.js14.18</b></li> <li>● <b>Node.js16.17</b></li> <li>● <b>Node.js18.15</b></li> <li>● <b>Python2.7</b></li> <li>● <b>Python3.6</b></li> <li>● <b>Python3.10</b></li> <li>● <b>Go1.x</b></li> <li>● <b>C#(.NET Core 2.1)</b></li> <li>● <b>C#(.NET Core 3.1)</b></li> <li>● <b>Custom</b></li> <li>● <b>PHP7.3</b></li> <li>● <b>Python3.9</b></li> <li>● <b>http</b></li> <li>● <b>Custom Image</b></li> <li>● <b>Cangjie1.0</b></li> </ul> |
| etag    | String | 依赖包的md5值                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| size    | Long   | 依赖包大小。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |



| 参数            | 参数类型    | 描述                  |
|---------------|---------|---------------------|
| name          | String  | 依赖包名称。              |
| description   | String  | 依赖包描述。              |
| file_name     | String  | 依赖包文件名，如果创建方式为zip时。 |
| version       | Long    | 依赖包版本编号。            |
| dep_id        | String  | 依赖包ID               |
| last_modified | Integer | 函数最后一次更新时间。         |

状态码： 400

表 7-284 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-285 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-286 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-287 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-288 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

更新函数最大实例数为300。

```
PUT https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/config-max-instance
{
 "max_instance_num" : 300
}
```

## 响应示例

状态码： 200

OK

```
{
 "func_urn" : "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",
 "func_name" : "test",
 "domain_id" : "14ee2e3501124efcbca7998baa24xxxx",
 "namespace" : "46b6f338fc3445b8846c71dfb1fbxxxx",
 "project_name" : "xxxxxx",
 "package" : "default",
 "runtime" : "Node.js6.10",
 "timeout" : 3,
 "handler" : "fssExampleCsharp2.1::fssExampleCsharp21.Program::MyFunc",
 "memory_size" : 128,
 "cpu" : 300,
 "code_type" : "inline",
 "code_filename" : "index.js",
 "code_size" : 272,
 "digest" :
"faa825575c45437cddd4e369bea69893bcbe195d478178462ad90984fe72993f3f59d15f41c5373f807f3e05fb9af322c55dabeb16565c386e402413458e6068",
 "version" : "latest",
 "image_name" : "latest-191025153727@zehht",
 "last_modified" : "2019-10-25 15:37:27",
 "strategy_config" : {
 "concurrency" : 300,
 "concurrent_num" : 1
 }
}
```

```
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新函数最大实例数为300。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateFunctionMaxInstanceConfigSolution {
 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 UpdateFunctionMaxInstanceConfigRequest request = new
UpdateFunctionMaxInstanceConfigRequest();
 UpdateFunctionMaxInstanceConfigRequestBody body = new
UpdateFunctionMaxInstanceConfigRequestBody();
 body.withMaxInstanceNum(300);
 request.withBody(body);
 try {
 UpdateFunctionMaxInstanceConfigResponse response =
client.updateFunctionMaxInstanceConfig(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

更新函数最大实例数为300。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = UpdateFunctionMaxInstanceConfigRequest()
 request.body = UpdateFunctionMaxInstanceConfigRequestBody(
 max_instance_num=300
)
 response = client.update_function_max_instance_config(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

更新函数最大实例数为300。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()
```

```

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.UpdateFunctionMaxInstanceConfigRequest{
 maxInstanceNumUpdateFunctionMaxInstanceConfigRequestBody:= int32(300)
 request.Body = &model.UpdateFunctionMaxInstanceConfigRequestBody{
 MaxInstanceNum: &maxInstanceNumUpdateFunctionMaxInstanceConfigRequestBody,
 }
}
response, err := client.UpdateFunctionMaxInstanceConfig(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.6.9 查询函数标签列表

### 功能介绍

查询函数标签列表

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/{resource\_type}/{resource\_id}/tags

表 7-289 路径参数

| 参数            | 是否必选 | 参数类型   | 描述                                       |
|---------------|------|--------|------------------------------------------|
| project_id    | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| resource_type | 是    | String | 资源类型，此处请填写 functions                     |
| resource_id   | 是    | String | 资源ID，函数urn。                              |

## 请求参数

表 7-290 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                            |
|--------------|------|--------|---------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                    |

## 响应参数

状态码： 200

表 7-291 响应 Body 参数

| 参数       | 参数类型                                    | 描述     |
|----------|-----------------------------------------|--------|
| tags     | Array of <a href="#">KvItem</a> objects | 标签列表   |
| sys_tags | Array of <a href="#">KvItem</a> objects | 系统标签列表 |

表 7-292 KvItem

| 参数    | 参数类型   | 描述 |
|-------|--------|----|
| key   | String | 键  |
| value | String | 值  |

**状态码： 400**

**表 7-293 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 401**

**表 7-294 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 403**

**表 7-295 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 500**

**表 7-296 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询资源标签列表。

```
GET /v2/{project_id}/{resource_type}/{resource_id}/tags
```

## 响应示例

**状态码： 200**

ok

```
{
 "tags": [{
 "key": "xxx",
 "value": "yyy"
 }],
 "sys_tags": [{
 "key": "_sys_enterprise_project_id",
 "value": "5aa119a8-d25b-45a7-8d1b-88e127885635"
 }]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListFunctionTagsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListFunctionTagsRequest request = new ListFunctionTagsRequest();
 try {
 ListFunctionTagsResponse response = client.listFunctionTags(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 }
 }
}
```



```
 System.out.println(e.getErrorMsg());
 }
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListFunctionTagsRequest()
 response = client.list_function_tags(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).
Build()

request := &model.ListFunctionTagsRequest{}
response, err := client.ListFunctionTags(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.6.10 禁用/启动函数快照

### 功能介绍

禁用/启动函数快照，仅支持java运行时函数，且为非latest版本才能开启函数快照功能。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/fgs/functions/{function\_urn}/snapshots/{action}

表 7-297 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                                             |
|--------------|------|--------|------------------------------------------------------------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。                                                       |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。                                                              |
| action       | 是    | String | 禁用/启用<br>枚举值：<br><ul style="list-style-type: none"> <li>• enable</li> <li>• disable</li> </ul> |

## 请求参数

表 7-298 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 400

表 7-299 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-300 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-301 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-302 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-303 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

- 启动函数快照。  
POST https://{Endpoint}/v2/{project\_id}/fgs/functions/{function\_urn}/snapshots/enable
- 禁用函数快照。  
POST https://{Endpoint}/v2/{project\_id}/fgs/functions/{function\_urn}/snapshots/disable

## 响应示例

状态码： 404

Not Found 找不到资源。

```
{
 "error_code" : "FSS.1051",
 "error_msg" : "Not found the function"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateFuncSnapshotSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 UpdateFuncSnapshotRequest request = new UpdateFuncSnapshotRequest();
 try {
 UpdateFuncSnapshotResponse response = client.updateFuncSnapshot(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = UpdateFuncSnapshotRequest()
 response = client.update_func_snapshot(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.UpdateFuncSnapshotRequest{}
 response, err := client.UpdateFuncSnapshot(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

```
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.6.11 获取指定函数绑定的 servicebridge 函数列表

### 功能介绍

获取指定函数绑定的servicebridge函数列表信息。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/servicebridge/relation

表 7-304 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

表 7-305 Query 参数

| 参数   | 是否必选 | 参数类型   | 描述              |
|------|------|--------|-----------------|
| type | 否    | String | servicebridge类型 |

## 请求参数

表 7-306 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-307 响应 Body 参数

| 参数     | 参数类型                                                | 描述                      |
|--------|-----------------------------------------------------|-------------------------|
| [数组元素] | Array of <a href="#">ListFunctionResult</a> objects | 函数绑定的servicebridge函数列表。 |

表 7-308 ListFunctionResult

| 参数           | 参数类型   | 描述                                    |
|--------------|--------|---------------------------------------|
| resource_id  | String | 资源id。                                 |
| func_urn     | String | 函数的URN（Uniform Resource Name），唯一标识函数。 |
| func_name    | String | 函数名称。                                 |
| domain_id    | String | 域名id。                                 |
| namespace    | String | 租户的project id。                        |
| project_name | String | 租户的project name。                      |



| 参数      | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| package | String | 函数所属的分组Package，用于用户针对函数的自定义分组。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| runtime | String | <p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>Java8</b></li> <li>● <b>Java11</b></li> <li>● <b>Node.js6.10</b></li> <li>● <b>Node.js8.10</b></li> <li>● <b>Node.js10.16</b></li> <li>● <b>Node.js12.13</b></li> <li>● <b>Node.js14.18</b></li> <li>● <b>Node.js16.17</b></li> <li>● <b>Node.js18.15</b></li> <li>● <b>Python2.7</b></li> <li>● <b>Python3.6</b></li> <li>● <b>Python3.10</b></li> <li>● <b>Go1.x</b></li> <li>● <b>C#(.NET Core 2.1)</b></li> <li>● <b>C#(.NET Core 3.1)</b></li> <li>● <b>Custom</b></li> <li>● <b>PHP7.3</b></li> <li>● <b>Python3.9</b></li> <li>● <b>http</b></li> <li>● <b>Custom Image</b></li> <li>● <b>Cangjie1.0</b></li> </ul> |

| 参数            | 参数类型    | 描述                                                                                                                                                                                                                                                                         |
|---------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| timeout       | Integer | 函数执行超时时间，超时函数将被强行停止，范围3~259200秒。                                                                                                                                                                                                                                           |
| handler       | String  | 函数执行入口 规则：xx.xx，必须包含“.” 举例：对于node.js函数：myfunction.handler，则表示函数的文件名为myfunction.js，执行的入口函数名为handler。长度不超过128个字符。<br>最小长度：3<br>最大长度：128                                                                                                                                      |
| memory_size   | Integer | 函数消耗的内存。单位M。取值范围为：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128，最大值为4096。                                                                                                                                                                           |
| gpu_memory    | Integer | 函数消耗的显存，只支持自定义运行时与自定义镜像函数配置GPU。单位MB。取值范围为：1024、2048、3072、4096、5120、6144、7168、8192、9216、10240、11264、12288、13312、14336、15360、16384。最小值为1024，最大值为16384。                                                                                                                       |
| cpu           | Integer | 函数占用的cpu资源。单位为millicore（1 core=1000 millicores）。取值与MemorySize成比例，默认是128M内存占0.1个核（100 millicores）。                                                                                                                                                                          |
| code_type     | String  | 函数代码类型，取值有5种。inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包，主要针对Java函数。Custom-Image-Swr: 函数代码来源于SWR自定义镜像。<br>枚举值：<br><ul style="list-style-type: none"> <li>• inline</li> <li>• zip</li> <li>• obs</li> <li>• jar</li> <li>• Custom-Image-Swr</li> </ul> |
| code_url      | String  | 当code_type为obs时，该值为函数代码包在OBS上的地址，code_type为其他值时，该字段为空。                                                                                                                                                                                                                     |
| code_filename | String  | 函数的文件名，当code_type为jar/zip时必须提供该字段，inline和obs不需要提供。                                                                                                                                                                                                                         |
| code_size     | Long    | 函数大小，单位：字节。                                                                                                                                                                                                                                                                |

| 参数                    | 参数类型                         | 描述                                                                                                                            |
|-----------------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| user_data             | String                       | 用户自定义的name/value信息。在函数中使用的参数。举例：如函数要访问某个主机，可以设置自定义参数：Host={host_ip}，最多定义20个，总长度不超过4KB。                                        |
| encrypted_user_data   | String                       | 用户自定义的name/value信息，用于需要加密的配置。                                                                                                 |
| digest                | String                       | 函数代码SHA512 hash值，用于判断函数是否变化。                                                                                                  |
| version               | String                       | 函数版本号。                                                                                                                        |
| image_name            | String                       | 函数版本的内部标识。                                                                                                                    |
| xrole                 | String                       | 函数使用的权限委托名称，需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。                                                                         |
| app_xrole             | String                       | 函数app使用的权限委托名称，需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。                                                                      |
| description           | String                       | 函数描述。                                                                                                                         |
| last_modified         | String                       | 函数最后一次更新时间。                                                                                                                   |
| func_vpc_id           | String                       | 用户的vpcid。                                                                                                                     |
| strategy_config       | <b>StrategyConfig</b> object | 函数策略配置。                                                                                                                       |
| extend_config         | String                       | 函数扩展配置。                                                                                                                       |
| initializer_handler   | String                       | 函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。举例：对于node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。 |
| initializer_timeout   | Integer                      | 初始化超时时间，超时函数将被强行停止，范围1~300秒。当配置初始化函数时，此参数必填。                                                                                  |
| pre_stop_handler      | String                       | 函数预停止函数的入口，规则：xx.xx，必须包含“.”。举例：对于node.js函数：myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。    |
| pre_stop_timeout      | Integer                      | 初始化超时时间，超时函数将被强行停止，范围1~90秒。                                                                                                   |
| enterprise_project_id | String                       | 企业项目ID，在企业用户创建函数时必填。                                                                                                          |
| long_time             | Boolean                      | 是否允许进行长时间超时设置。                                                                                                                |

| 参数                       | 参数类型             | 描述                                                                                              |
|--------------------------|------------------|-------------------------------------------------------------------------------------------------|
| log_group_id             | String           | 自定义日志查询组id                                                                                      |
| log_stream_id            | String           | 自定义日志查询流id                                                                                      |
| type                     | String           | v2表示为正式版本,v1为废弃版本。<br>枚举值: <ul style="list-style-type: none"> <li>• v1</li> <li>• v2</li> </ul> |
| fail_count               | Integer          | 函数最近1天内执行失败的次数。                                                                                 |
| is_bridge_function       | Boolean          | 是否为bridge函数                                                                                     |
| bind_bridge_functionUrns | Array of strings | 绑定bridge函数的urn列表                                                                                |

表 7-309 StrategyConfig

| 参数             | 参数类型    | 描述                                                      |
|----------------|---------|---------------------------------------------------------|
| concurrency    | Integer | 单函数最大实例数, v1取值0和-1, v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用 |
| concurrent_num | Integer | 单实例并发数, v2版本才支持, 取值1到1000                               |

状态码: 401

表 7-310 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码: 403

表 7-311 响应 Body 参数

| 参数         | 参数类型   | 描述  |
|------------|--------|-----|
| error_code | String | 错误码 |

| 参数        | 参数类型   | 描述   |
|-----------|--------|------|
| error_msg | String | 错误信息 |

状态码： 404

表 7-312 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-313 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

获取指定函数绑定的servicebridge函数列表。

```
GET https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/servicebridge/relation?type=rds
```

## 响应示例

状态码： 200

ok

```
[{
 "func_urn" : "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",
 "func_name" : "bridge-test",
 "domain_id" : "cff01_hk",
 "namespace" : "7aad83af3e8d42e99ac194e8419e2c9b",
 "project_name" : "xxxxxxx",
 "package" : "default",
 "runtime" : "Node.js6.10",
 "timeout" : 3,
 "handler" : "test.handler",
 "memory_size" : 128,
 "cpu" : 300,
 "code_type" : "inline",
 "code_filename" : "index.js",
 "code_size" : 272,
 "digest" :
 "decbce6939297b0b5ec6d1a23bf9c725870f5e69fc338a89a6a4029264688dc26338f56d08b6535de47f15ad53
 8e22ca66613b9a46f807d50b687bb53fded1c6",
}
```

```
"version" : "latest",
"xrole" : "cff",
"image_name" : "latest-191025153727@zehht",
"last_modified" : "2019-10-25 15:37:27",
"strategy_config" : {
 "concurrency" : 0
},
"initializer_handler" : "index.initializer",
"initializer_timeout" : 3,
"pre_stop_handler" : "index.pre_stop_handler",
"pre_stop_timeout" : 3
}]
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListBridgeFunctionsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListBridgeFunctionsRequest request = new ListBridgeFunctionsRequest();
 request.withType("<type>");
 try {
 ListBridgeFunctionsResponse response = client.listBridgeFunctions(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListBridgeFunctionsRequest()
 request.type = "<type>"
 response = client.list_bridge_functions(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListBridgeFunctionsRequest{}
```

```
typeRequest:= "<type>"
request.Type = &typeRequest
response, err := client.ListBridgeFunctions(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.6.12 查询函数快照制作状态

### 功能介绍

查询函数快照制作状态。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/snapshots/{action}

表 7-314 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |



| 参数           | 是否必选 | 参数类型   | 描述                                                                                                 |
|--------------|------|--------|----------------------------------------------------------------------------------------------------|
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。                                                                  |
| action       | 是    | String | 查询快照制作开关状态枚举值： <ul style="list-style-type: none"> <li>• state</li> <li>• enableSnapshot</li> </ul> |

## 请求参数

表 7-315 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-316 响应 Body 参数

| 参数    | 参数类型   | 描述      |
|-------|--------|---------|
| state | String | 快照制作状态  |
| code  | String | 快照制作响应码 |

状态码： 400

表 7-317 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码：401****表 7-318** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码：403****表 7-319** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码：404****表 7-320** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码：500****表 7-321** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**请求示例**

查询函数快照制作状态。

```
GET https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/snapshots/state
```

## 响应示例

**状态码： 200**

OK

```
{
 "state": "enable"
}
```

**状态码： 404**

Not Found 找不到资源。

```
{
 "error_code": "FSS.1051",
 "error_msg": "Not found the function"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowFuncSnapshotStateSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowFuncSnapshotStateRequest request = new ShowFuncSnapshotStateRequest();
 try {
 ShowFuncSnapshotStateResponse response = client.showFuncSnapshotState(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 }
 }
}
```

```
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowFuncSnapshotStateRequest()
 response = client.show_func_snapshot_state(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
```

```
functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.ShowFuncSnapshotStateRequest{}
response, err := client.ShowFuncSnapshotState(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

### 7.6.13 查询资源标签

#### 功能介绍

查询资源标签。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/{resource\_type}/tags

表 7-322 路径参数

| 参数            | 是否必选 | 参数类型   | 描述                                       |
|---------------|------|--------|------------------------------------------|
| project_id    | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| resource_type | 是    | String | 资源类型，此处请填写 functions                     |

## 请求参数

表 7-323 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                            |
|--------------|------|--------|---------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                    |

## 响应参数

状态码： 200

表 7-324 响应 Body 参数

| 参数       | 参数类型                                     | 描述     |
|----------|------------------------------------------|--------|
| tags     | Array of <a href="#">TagItem</a> objects | 标签列表   |
| sys_tags | Array of <a href="#">TagItem</a> objects | 系统标签列表 |

表 7-325 TagItem

| 参数     | 参数类型             | 描述 |
|--------|------------------|----|
| key    | String           | 键  |
| values | Array of strings | 值  |

**状态码： 400**

**表 7-326** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 401**

**表 7-327** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 403**

**表 7-328** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 500**

**表 7-329** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询资源标签。

```
GET https://{Endpoint}/v2/{project_id}/{resource_type}/tags
```

## 响应示例

状态码： 200

ok

```
{
 "tags" : [{
 "key" : "xxx",
 "values" : ["yyy", "zzz"]
 }],
 "sys_tags" : [{
 "key" : "_sys_enterprise_project_id",
 "values" : ["5aa119a8-d25b-45a7-8d1b-88e127885635"]
 }]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowProjectTagsListSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowProjectTagsListRequest request = new ShowProjectTagsListRequest();
 try {
 ShowProjectTagsListResponse response = client.showProjectTagsList(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 }
 }
}
```



```
 System.out.println(e.getErrorMsg());
 }
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowProjectTagsListRequest()
 response = client.show_project_tags_list(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
WithCredential(auth).
Build()

request := &model.ShowProjectTagsListRequest{}
response, err := client.ShowProjectTagsList(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

### 7.6.14 查询资源实例

#### 功能介绍

查询资源实例。

#### 调用方法

请参见[如何调用API](#)。

#### URI

POST /v2/{project\_id}/{resource\_type}/resource-instances/{action}

表 7-330 路径参数

| 参数            | 是否必选 | 参数类型   | 描述                                                                                           |
|---------------|------|--------|----------------------------------------------------------------------------------------------|
| project_id    | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。                                                     |
| resource_type | 是    | String | 资源类型，此处请填写 functions                                                                         |
| action        | 是    | String | 过滤/计数<br>枚举值：<br><ul style="list-style-type: none"> <li>• filter</li> <li>• count</li> </ul> |

## 请求参数

表 7-331 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                            |
|--------------|------|--------|---------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                    |

表 7-332 请求 Body 参数

| 参数              | 是否必选 | 参数类型                                    | 描述                                                                                                |
|-----------------|------|-----------------------------------------|---------------------------------------------------------------------------------------------------|
| without_any_tag | 否    | Boolean                                 | 是否通过标签过滤                                                                                          |
| limit           | 否    | String                                  | 每页显示条数                                                                                            |
| offset          | 否    | String                                  | 查询偏移量                                                                                             |
| action          | 否    | String                                  | 查询指定action<br>枚举值：<br><ul style="list-style-type: none"> <li>• count</li> <li>• filter</li> </ul> |
| matches         | 否    | Array of <a href="#">KvItem</a> objects | 查询指定键值对                                                                                           |

| 参数       | 是否必选 | 参数类型                            | 描述         |
|----------|------|---------------------------------|------------|
| sys_tags | 否    | Array of <b>TagItem</b> objects | 查询指定系统标签列表 |

表 7-333 KvlItem

| 参数    | 是否必选 | 参数类型   | 描述 |
|-------|------|--------|----|
| key   | 否    | String | 键  |
| value | 否    | String | 值  |

表 7-334 TagItem

| 参数     | 是否必选 | 参数类型             | 描述 |
|--------|------|------------------|----|
| key    | 否    | String           | 键  |
| values | 否    | Array of strings | 值  |

## 响应参数

状态码： 200

表 7-335 响应 Body 参数

| 参数          | 参数类型                                                 | 描述     |
|-------------|------------------------------------------------------|--------|
| resources   | Array of <b>ListEnterpriseResourceResult</b> objects | 企业项目列表 |
| total_count | Long                                                 | 资源数    |

表 7-336 ListEnterpriseResourceResult

| 参数              | 参数类型                                       | 描述   |
|-----------------|--------------------------------------------|------|
| resource_id     | String                                     | 资源ID |
| resource_detail | <b>ListEnterpriseResourceDetail</b> object |      |

| 参数            | 参数类型                           | 描述     |
|---------------|--------------------------------|--------|
| tags          | Array of <b>KvItem</b> objects | 标签列表   |
| sys_tags      | Array of <b>KvItem</b> objects | 系统标签列表 |
| resource_name | String                         | 资源名称   |

表 7-337 ListEnterpriseResourceDetail

| 参数       | 参数类型   | 描述    |
|----------|--------|-------|
| detailId | String | 函数urn |

表 7-338 KvItem

| 参数    | 参数类型   | 描述 |
|-------|--------|----|
| key   | String | 键  |
| value | String | 值  |

状态码： 400

表 7-339 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-340 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 403**

**表 7-341** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 404**

**表 7-342** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 500**

**表 7-343** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询资源实例。

POST https://{Endpoint}/v2/{project\_id}/{resource\_type}/resource-instances/{action}

```
{
 "without_any_tag" : true,
 "limit" : 5,
 "matches" : [{
 "key" : "resource_name",
 "value" : "test_function"
 }]
}
```

## 响应示例

**状态码： 200**

ok

- 示例 1

```
{
 "resources": [{
 "resource_id": "34e4516e-e324-412b-914e-c4e568c7d813",
 "resource_detail": {
 "detailId": "urn:fss:{region-id}:xxx:function:default:test_xxx:latest"
 },
 "tags": [],
 "sys_tags": [{
 "key": "_sys_enterprise_project_id",
 "value": "df5edab8-c458-4a4c-b87b-a4d3b0a757ce"
 }],
 "resource_name": "test_v2_1"
 }]
}
```

- 示例 2

1

## SDK 代码示例

SDK代码示例如下。

### Java

查询资源实例。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class ShowResInstanceInfoSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowResInstanceInfoRequest request = new ShowResInstanceInfoRequest();
 ListEnterpriseResourceRequestBody body = new ListEnterpriseResourceRequestBody();
 List<KvItem> listbodyMatches = new ArrayList<>();
 listbodyMatches.add(
 new KvItem()
 .withKey("resource_name")
 .withValue("test_function")
);
 }
}
```

```
body.withMatches(listbodyMatches);
body.withLimit("5");
body.withWithoutAnyTag(true);
request.withBody(body);
try {
 ShowResInstanceInfoResponse response = client.showResInstanceInfo(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

查询资源实例。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowResInstanceInfoRequest()
 listMatchesbody = [
 KvItem(
 key="resource_name",
 value="test_function"
)
]
 request.body = ListEnterpriseResourceRequestBody(
 matches=listMatchesbody,
 limit="5",
 without_any_tag=True
)
 response = client.show_res_instance_info(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```



## Go

查询资源实例。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowResInstanceInfoRequest{
 keyMatches:= "resource_name"
 valueMatches:= "test_function"
 var listMatchesbody = []model.KvItem{
 {
 Key: &keyMatches,
 Value: &valueMatches,
 },
 }
 limitListEnterpriseResourceRequestBody:= "5"
 withoutAnyTagListEnterpriseResourceRequestBody:= true
 request.Body = &model.ListEnterpriseResourceRequestBody{
 Matches: &listMatchesbody,
 Limit: &limitListEnterpriseResourceRequestBody,
 WithoutAnyTag: &withoutAnyTagListEnterpriseResourceRequestBody,
 }
 response, err := client.ShowResInstanceInfo(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

### 7.6.15 删除资源标签

#### 功能介绍

删除资源标签。

#### 调用方法

请参见[如何调用API](#)。

#### URI

DELETE /v2/{project\_id}/{resource\_type}/{resource\_id}/tags/delete

表 7-344 路径参数

| 参数            | 是否必选 | 参数类型   | 描述                                       |
|---------------|------|--------|------------------------------------------|
| project_id    | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| resource_type | 是    | String | 资源类型，此处请填写 functions                     |
| resource_id   | 是    | String | 资源ID                                     |

## 请求参数

表 7-345 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

表 7-346 请求 Body 参数

| 参数       | 是否必选 | 参数类型                           | 描述                                                                                               |
|----------|------|--------------------------------|--------------------------------------------------------------------------------------------------|
| action   | 否    | String                         | action名称<br>枚举值：<br><ul style="list-style-type: none"> <li>• create</li> <li>• delete</li> </ul> |
| tags     | 否    | Array of <b>KvItem</b> objects | 标签列表                                                                                             |
| sys_tags | 否    | Array of <b>KvItem</b> objects | 系统标签列表                                                                                           |

表 7-347 KvItem

| 参数    | 是否必选 | 参数类型   | 描述 |
|-------|------|--------|----|
| key   | 否    | String | 键  |
| value | 否    | String | 值  |

## 响应参数

状态码： 400

表 7-348 响应 Body 参数

| 参数         | 参数类型   | 描述  |
|------------|--------|-----|
| error_code | String | 错误码 |

| 参数        | 参数类型   | 描述   |
|-----------|--------|------|
| error_msg | String | 错误信息 |

## 请求示例

删除资源标签。

```
DELETE https://{Endpoint}/v2/{project_id}/{resource_type}/{resource_id}/tags/delete
{
 "tags": [{
 "key": "testKey1",
 "value": "testValue1"
 }],
 "action": "delete"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

删除资源标签。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class DeleteTagsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
```

```
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
DeleteTagsRequest request = new DeleteTagsRequest();
UpdateFunctionTagsRequestBody body = new UpdateFunctionTagsRequestBody();
List<KvItem> listbodyTags = new ArrayList<>();
listbodyTags.add(
 new KvItem()
 .withKey("testKey1")
 .withValue("testValue1")
);
body.withTags(listbodyTags);
body.withAction("delete");
request.withBody(body);
try {
 DeleteTagsResponse response = client.deleteTags(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

删除资源标签。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = DeleteTagsRequest()
 listTagsbody = [
 KvItem(
 key="testKey1",
 value="testValue1"
)
]
 request.body = UpdateFunctionTagsRequestBody(
 tags=listTagsbody,
 action="delete"
)
```

```
response = client.delete_tags(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

删除资源标签。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.DeleteTagsRequest{}
 keyTags:= "testKey1"
 valueTags:= "testValue1"
 var listTagsbody = []model.KvItem{
 {
 Key: &keyTags,
 Value: &valueTags,
 },
 }
 actionUpdateFunctionTagsRequestBody:= "delete"
 request.Body = &model.UpdateFunctionTagsRequestBody{
 Tags: &listTagsbody,
 Action: &actionUpdateFunctionTagsRequestBody,
 }
 response, err := client.DeleteTags(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                |
|-----|-------------------|
| 204 | No Content        |
| 400 | Bad Request 请求错误。 |

## 错误码

请参见[错误码](#)。

## 7.6.16 创建资源标签

### 功能介绍

创建资源标签。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/{resource\_type}/{resource\_id}/tags/create

表 7-349 路径参数

| 参数            | 是否必选 | 参数类型   | 描述                                       |
|---------------|------|--------|------------------------------------------|
| project_id    | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| resource_type | 是    | String | 资源类型，此处请填写 functions                     |
| resource_id   | 是    | String | 资源ID                                     |

### 请求参数

表 7-350 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                            |
|--------------|------|--------|---------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。 |

| 参数           | 是否必选 | 参数类型   | 描述         |
|--------------|------|--------|------------|
| Content-Type | 是    | String | 消息体的类型（格式） |

表 7-351 请求 Body 参数

| 参数       | 是否必选 | 参数类型                           | 描述                                                                                               |
|----------|------|--------------------------------|--------------------------------------------------------------------------------------------------|
| action   | 否    | String                         | action名称<br>枚举值：<br><ul style="list-style-type: none"> <li>• create</li> <li>• delete</li> </ul> |
| tags     | 否    | Array of <b>KvItem</b> objects | 标签列表                                                                                             |
| sys_tags | 否    | Array of <b>KvItem</b> objects | 系统标签列表                                                                                           |

表 7-352 KvItem

| 参数    | 是否必选 | 参数类型   | 描述 |
|-------|------|--------|----|
| key   | 否    | String | 键  |
| value | 否    | String | 值  |

## 响应参数

状态码： 400

表 7-353 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

创建键值对分别为"testKey1:testValue1"、"testKey2:testValue2"的资源标签。

```
POST https://{Endpoint}/v2/{project_id}/{resource_type}/{resource_id}/tags/create
{
```



```
"tags" : [{
 "key" : "testKey1",
 "value" : "testValue1"
}, {
 "key" : "testKey2",
 "value" : "testValue2"
}],
"action" : "create"
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

创建键值对分别为"testKey1:testValue1"、"testKey2:testValue2"的资源标签。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateTagsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();

 CreateTagsRequest request = new CreateTagsRequest();
 UpdateFunctionTagsRequestBody body = new UpdateFunctionTagsRequestBody();
 List<KvItem> listbodyTags = new ArrayList<>();
 listbodyTags.add(
 new KvItem()
 .withKey("testKey1")
 .withValue("testValue1")
);
 listbodyTags.add(
 new KvItem()
 .withKey("testKey2")
);
 }
}
```

```
 .withValue("testValue2")
);
 body.withTags(listbodyTags);
 body.withAction("create");
 request.withBody(body);
 try {
 CreateTagsResponse response = client.createTags(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

## Python

创建键值对分别为"testKey1:testValue1"、"testKey2:testValue2"的资源标签。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateTagsRequest()
 listTagsbody = [
 Kvltem(
 key="testKey1",
 value="testValue1"
),
 Kvltem(
 key="testKey2",
 value="testValue2"
)
]
 request.body = UpdateFunctionTagsRequestBody(
 tags=listTagsbody,
 action="create"
)
 response = client.create_tags(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
```

```
print(e.request_id)
print(e.error_code)
print(e.error_msg)
```

## Go

创建键值对分别为"testKey1:testValue1"、"testKey2:testValue2"的资源标签。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.CreateTagsRequest{}
 keyTags:= "testKey1"
 valueTags:= "testValue1"
 keyTags1:= "testKey2"
 valueTags1:= "testValue2"
 var listTagsbody = []model.KvItem{
 {
 Key: &keyTags,
 Value: &valueTags,
 },
 {
 Key: &keyTags1,
 Value: &valueTags1,
 },
 }
 actionUpdateFunctionTagsRequestBody:= "create"
 request.Body = &model.UpdateFunctionTagsRequestBody{
 Tags: &listTagsbody,
 Action: &actionUpdateFunctionTagsRequestBody,
 }
 response, err := client.CreateTags(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                |
|-----|-------------------|
| 204 | No Content        |
| 400 | Bad Request 请求错误。 |

## 错误码

请参见[错误码](#)。

## 7.6.17 创建下沉入口

### 功能介绍

创建下沉入口。（该功能目前仅支持华北-北京四、华东-上海一）

### 调用方法

请参见[如何调用API](#)。

## URI

POST /v2/{project\_id}/fgs/vpc-endpoint

表 7-354 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

## 请求参数

表 7-355 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |

| 参数           | 是否必选 | 参数类型   | 描述         |
|--------------|------|--------|------------|
| Content-Type | 是    | String | 消息体的类型（格式） |

表 7-356 请求 Body 参数

| 参数        | 是否必选 | 参数类型   | 描述             |
|-----------|------|--------|----------------|
| flavor    | 否    | String | 选定EP的规格，默认为大规格 |
| xrole     | 否    | String | 制作EP时使用的租户委托名称 |
| vpc_id    | 是    | String | 对接EP使用的租户VPCID |
| subnet_id | 是    | String | 对接EP使用的租户子网ID  |

## 响应参数

状态码： 200

表 7-357 响应 Body 参数

| 参数        | 参数类型             | 描述   |
|-----------|------------------|------|
| endpoints | Array of strings | Ip列表 |
| address   | String           | 域名地址 |

状态码： 400

表 7-358 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-359 响应 Body 参数

| 参数         | 参数类型   | 描述  |
|------------|--------|-----|
| error_code | String | 错误码 |

| 参数        | 参数类型   | 描述   |
|-----------|--------|------|
| error_msg | String | 错误信息 |

状态码： 403

表 7-360 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-361 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-362 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

利用xxx委托，为vpc:a938xxxx，子网:a4a1xxxx创建大规格下沉入口。

```
POST https://{Endpoint}/v2/{project_id}/fgs/vpc-endpoint
{
 "vpc_id": "a938xxxx-7xxx-4xxx-bxxx-cee915xxxxxx",
 "subnet_id": "a4a1xxxx-9xxx-4xxx-8xxx-bb46f7xxxxxx",
 "flavor": "default",
 "xrole": "xxx"
}
```

## 响应示例

状态码： 200

ok

```
{
 "endpoints": [],
 "address": "https://xx"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

利用xxx委托，为vpc:a938xxxx，子网:a4a1xxxx创建大规格下沉入口。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateVpcEndpointSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateVpcEndpointRequest request = new CreateVpcEndpointRequest();
 CreateVpcEndpointRequestBody body = new CreateVpcEndpointRequestBody();
 body.withSubnetId("a4a1xxxx-9xxx-4xxx-8xxx-bb46f7xxxxxx");
 body.withVpcId("a938xxxx-7xxx-4xxx-bxxx-cee915xxxxxx");
 body.withXrole("xxx");
 body.withFlavor("default");
 request.withBody(body);
 try {
 CreateVpcEndpointResponse response = client.createVpcEndpoint(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 }
 }
}
```

```
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
```

## Python

利用xxx委托, 为vpc:a938xxxx, 子网:a4a1xxxx创建大规格下沉入口。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateVpcEndpointRequest()
 request.body = CreateVpcEndpointRequestBody(
 subnet_id="a4a1xxxx-9xxx-4xxx-8xxx-bb46f7xxxxxx",
 vpc_id="a938xxxx-7xxx-4xxx-bxxx-cee915xxxxxx",
 xrole="xxx",
 flavor="default"
)
 response = client.create_vpc_endpoint(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

利用xxx委托, 为vpc:a938xxxx, 子网:a4a1xxxx创建大规格下沉入口。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
```



```
// In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.CreateVpcEndpointRequest{
 xroleCreateVpcEndpointRequestBody:= "xxx"
 flavorCreateVpcEndpointRequestBody:= "default"
 request.Body = &model.CreateVpcEndpointRequestBody{
 SubnetId: "a4a1xxxx-9xxx-4xxx-8xxx-bb46f7xxxxxx",
 VpcId: "a938xxxx-7xxx-4xxx-bxxx-cee915xxxxxx",
 Xrole: &xroleCreateVpcEndpointRequestBody,
 Flavor: &flavorCreateVpcEndpointRequestBody,
 }
}
response, err := client.CreateVpcEndpoint(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.6.18 删除下沉入口

### 功能介绍

删除下沉入口。（该功能目前仅支持华北-北京四、华东-上海一）

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/fgs/vpc-endpoint/{vpc\_id}/{subnet\_id}

表 7-363 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| vpc_id     | 是    | String | vpc ID                                   |
| subnet_id  | 是    | String | 子网编号                                     |

### 请求参数

表 7-364 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

### 响应参数

状态码： 400

表 7-365 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

删除下沉入口。

```
DELETE https://{Endpoint}/v2/{project_id}/fgs/vpc-endpoint/{vpc_id}/{subnet_id}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class DeleteVpcEndpointSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 DeleteVpcEndpointRequest request = new DeleteVpcEndpointRequest();
 try {
 DeleteVpcEndpointResponse response = client.deleteVpcEndpoint(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = DeleteVpcEndpointRequest()
 response = client.delete_vpc_endpoint(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.DeleteVpcEndpointRequest{}
 response, err := client.DeleteVpcEndpoint(request)
```

```
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                |
|-----|-------------------|
| 204 | No Content        |
| 400 | Bad Request 请求错误。 |

## 错误码

请参见[错误码](#)。

## 7.6.19 更新函数置顶状态

### 功能介绍

更新函数置顶状态

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/fgs/functions/{func\_urn}/collect/{state}

表 7-366 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| func_urn   | 是    | String | 函数URN                                    |
| state      | 是    | String | 置顶状态                                     |

## 请求参数

表 7-367 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 500

表 7-368 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

更新函数置顶状态。

```
PUT https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/collect>true
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateFunctionCollectStateSolution {
```

```
public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 UpdateFunctionCollectStateRequest request = new UpdateFunctionCollectStateRequest();
 try {
 UpdateFunctionCollectStateResponse response = client.updateFunctionCollectState(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = UpdateFunctionCollectStateRequest()
 response = client.update_function_collect_state(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
```

```
print(e.error_code)
print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.UpdateFunctionCollectStateRequest{}
 response, err := client.UpdateFunctionCollectState(request)
 if err == nil {
 fmt.Printf("%v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。



## 7.6.20 获取 servicebridge 可用的版本

### 功能介绍

获取servicebridge可用的版本

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/servicebridge/version

表 7-369 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

表 7-370 Query 参数

| 参数   | 是否必选 | 参数类型   | 描述                                   |
|------|------|--------|--------------------------------------|
| type | 是    | String | servicebridge 类型，可选 rds, mqs 或 cache |

### 请求参数

表 7-371 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

### 响应参数

状态码： 200

表 7-372 响应 Body 参数

| 参数     | 参数类型                                          | 描述 |
|--------|-----------------------------------------------|----|
| [数组元素] | Array of <b>ServiceBridge Version</b> objects |    |

表 7-373 ServiceBridgeVersion

| 参数       | 参数类型   | 描述        |
|----------|--------|-----------|
| name     | String | 代码包名      |
| version  | String | 代码版本      |
| code_url | String | 代码所在obs路径 |

状态码： 400

表 7-374 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-375 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-376 响应 Body 参数

| 参数         | 参数类型   | 描述  |
|------------|--------|-----|
| error_code | String | 错误码 |

| 参数        | 参数类型   | 描述   |
|-----------|--------|------|
| error_msg | String | 错误信息 |

状态码： 500

表 7-377 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

获取 rds 类型 bridge函数可用的版本。

```
GET https://{Endpoint}/v2/{project_id}/fgs/servicebridge/version?type=rds
```

## 响应示例

状态码： 200

ok

```
[{
 "name" : "xxx",
 "version" : "xxx",
 "code_url" : "xxx"
}]
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListBridgeVersionsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
```

this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
ListBridgeVersionsRequest request = new ListBridgeVersionsRequest();
request.withType("<type>");
try {
 ListBridgeVersionsResponse response = client.listBridgeVersions(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListBridgeVersionsRequest()
 request.type = "<type>"
 response = client.list_bridge_versions(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListBridgeVersionsRequest{}
 request.Type = "<type>"
 response, err := client.ListBridgeVersions(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.7 函数版本别名

### 7.7.1 获取指定函数所有版本别名列表

#### 功能介绍

获取函数版本别名列表。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/aliases

表 7-378 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见 FunctionGraph函数模型的描述。       |

#### 请求参数

表 7-379 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                            |
|--------------|------|--------|---------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                    |

#### 响应参数

状态码： 200

表 7-380 响应 Body 参数

| 参数     | 参数类型                                                    | 描述       |
|--------|---------------------------------------------------------|----------|
| [数组元素] | Array of <a href="#">ListVersionAliasResult</a> objects | 函数版本别名列表 |

表 7-381 ListVersionAliasResult

| 参数                         | 参数类型                | 描述         |
|----------------------------|---------------------|------------|
| name                       | String              | 要获取的别名名称。  |
| version                    | String              | 别名对应的版本名称。 |
| description                | String              | 别名描述信息。    |
| last_modified              | String              | 别名最后修改时间。  |
| alias_urn                  | String              | 版本别名唯一标识。  |
| additional_version_weights | Map<String,Integer> | 灰度版本信息     |

状态码： 401

表 7-382 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-383 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-384 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-385 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 503

表 7-386 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

当前租户函数别名列表。

GET https://{Endpoint}/v2/{project\_id}/fgs/functions/{function\_urn}/aliases

## 响应示例

状态码： 200

OK

```
[{
 "name" : "a1",
 "version" : "latest",
 "description" : "",
 "last_modified" : "2019-10-31 11:37:58",
 "alias_urn" : "urn:fss:{region}:46b6f338fc3445b8846c71dfb1fbxxx:function:default:xxxx:a1",
 "additional_version_weights" : {
 "v1" : 10
 }
}
```



## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListVersionAliasesSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListVersionAliasesRequest request = new ListVersionAliasesRequest();
 try {
 ListVersionAliasesResponse response = client.listVersionAliases(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
```

```
In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = ListVersionAliasesRequest()
 response = client.list_version_aliases(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListVersionAliasesRequest{}
 response, err := client.ListVersionAliases(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |
| 503 | Service Unavailable 服务不可用     |

## 错误码

请参见[错误码](#)。

## 7.7.2 创建函数版本别名

### 功能介绍

创建函数灰度版本别名。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/fgs/functions/{function\_urn}/aliases

表 7-387 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见 FunctionGraph 函数模型的描述。      |

## 请求参数

表 7-388 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

表 7-389 请求 Body 参数

| 参数                          | 是否必选 | 参数类型                        | 描述         |
|-----------------------------|------|-----------------------------|------------|
| name                        | 是    | String                      | 别名名称。      |
| version                     | 是    | String                      | 别名对应的版本名称。 |
| description                 | 否    | String                      | 别名描述信息。    |
| additional_version_weights  | 否    | Map<String,Integer>         | 百分比灰度配置信息  |
| additional_version_strategy | 否    | Map<String,VersionStrategy> | 指定规则灰度策略信息 |

表 7-390 VersionStrategy

| 参数           | 是否必选 | 参数类型                                  | 描述                                                                                                       |
|--------------|------|---------------------------------------|----------------------------------------------------------------------------------------------------------|
| rules        | 否    | Array of VersionStrategyRules objects | 规则列表                                                                                                     |
| combine_type | 否    | String                                | 所有规则聚合方式。and：所有规则都满足，or：满足其中一个枚举值： <ul style="list-style-type: none"> <li>• and</li> <li>• or</li> </ul> |

表 7-391 VersionStrategyRules

| 参数        | 是否必选 | 参数类型   | 描述                                                            |
|-----------|------|--------|---------------------------------------------------------------|
| rule_type | 否    | String | 参数类型。<br>枚举值：<br>• <b>Header</b>                              |
| param     | 否    | String | 规则参数名, 只支持大小写字母, 数字, 下划线, 中划线。                                |
| op        | 否    | String | 规则匹配操作符, 目前仅需支持 = 或者 in。<br>枚举值：<br>• <b>in</b><br>• <b>=</b> |
| value     | 否    | String | 规则值, 如果op为in, 则为逗号分隔的多值字符串                                    |

## 响应参数

状态码： 200

表 7-392 响应 Body 参数

| 参数                          | 参数类型                        | 描述         |
|-----------------------------|-----------------------------|------------|
| name                        | String                      | 要获取的别名名称。  |
| version                     | String                      | 别名对应的版本名称。 |
| description                 | String                      | 别名描述信息。    |
| last_modified               | String                      | 别名最后修改时间。  |
| alias_urn                   | String                      | 版本别名唯一标识。  |
| additional_version_weights  | Map<String,Integer>         | 百分比灰度配置信息  |
| additional_version_strategy | Map<String,VersionStrategy> | 指定规则灰度策略信息 |

表 7-393 VersionStrategy

| 参数           | 参数类型                                         | 描述                                                                                                                 |
|--------------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| rules        | Array of <b>VersionStrategyRules</b> objects | 规则列表                                                                                                               |
| combine_type | String                                       | 所有规则聚合方式。and: 所有规则都满足, or: 满足其中一个<br>枚举值:<br><ul style="list-style-type: none"> <li>• and</li> <li>• or</li> </ul> |

表 7-394 VersionStrategyRules

| 参数        | 参数类型   | 描述                                                                                                      |
|-----------|--------|---------------------------------------------------------------------------------------------------------|
| rule_type | String | 参数类型。<br>枚举值:<br><ul style="list-style-type: none"> <li>• Header</li> </ul>                             |
| param     | String | 规则参数名, 只支持大小写字母, 数字, 下划线, 中划线。                                                                          |
| op        | String | 规则匹配操作符, 目前仅需支持 = 或者 in。<br>枚举值:<br><ul style="list-style-type: none"> <li>• in</li> <li>• =</li> </ul> |
| value     | String | 规则值, 如果op为in, 则为逗号分隔的多值字符串                                                                              |

状态码: 400

表 7-395 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码: 401

表 7-396 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-397 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-398 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-399 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

- 为函数的v1版本创建别名a1。  
POST https://{Endpoint}/v2/{project\_id}/fgs/functions/{function\_urn}/aliases  

```
{
 "name": "a1",
 "version": "v1"
}
```
- 为函数的v1版本创建别名a1，且开启灰度版本，将v2版本设置为50%随机灰度。  
POST https://{Endpoint}/v2/{project\_id}/fgs/functions/{function\_urn}/aliases

```
{
 "name": "a1",
 "version": "v1",
 "additional_version_weights": {
 "v2": 50
 }
}
```

- 为函数的v1版本创建别名a1，且开启灰度版本，将v2版本设置为指定规则灰度。  
POST https://{Endpoint}/v2/{project\_id}/fgs/functions/{function\_urn}/aliases

```
{
 "name": "a1",
 "version": "v1",
 "additional_version_strategy": {
 "v2": {
 "combine_type": "and",
 "rules": [{
 "rule_type": "Header",
 "param": "version",
 "op": "=",
 "value": "v1"
 }]
 }
 }
}
```

## 响应示例

**状态码： 200**

ok

```
{
 "name": "a1",
 "version": "latest",
 "description": "",
 "last_modified": "2019-10-31 11:37:58",
 "alias_urn": "urn:fss:{region}:46b6f338fc3445b8846c71dfb1fbxxxx:function:default:xxxx:!a1"
}
```

**状态码： 404**

Not Found 找不到资源。

```
{
 "error_code": "FSS.1051",
 "error_msg": "Not found the function"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 为函数的v1版本创建别名a1。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
```



```
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateVersionAliasSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateVersionAliasRequest request = new CreateVersionAliasRequest();
 CreateVersionAliasRequestBody body = new CreateVersionAliasRequestBody();
 body.withVersion("v1");
 body.withName("a1");
 request.withBody(body);
 try {
 CreateVersionAliasResponse response = client.createVersionAlias(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

- 为函数的v1版本创建别名a1，且开启灰度版本，将v2版本设置为50%随机灰度。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.Map;
import java.util.HashMap;

public class CreateVersionAliasSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
```

```
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
CreateVersionAliasRequest request = new CreateVersionAliasRequest();
CreateVersionAliasRequestBody body = new CreateVersionAliasRequestBody();
Map<String, Integer> listbodyAdditionalVersionWeights = new HashMap<>();
listbodyAdditionalVersionWeights.put("v2", 50);
body.withAdditionalVersionWeights(listbodyAdditionalVersionWeights);
body.withVersion("v1");
body.withName("a1");
request.withBody(body);
try {
 CreateVersionAliasResponse response = client.createVersionAlias(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

- 为函数的v1版本创建别名a1，且开启灰度版本，将v2版本设置为指定规则灰度。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.List;
import java.util.ArrayList;
import java.util.Map;
import java.util.HashMap;

public class CreateVersionAliasSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
```

```
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
CreateVersionAliasRequest request = new CreateVersionAliasRequest();
CreateVersionAliasRequestBody body = new CreateVersionAliasRequestBody();
List<VersionStrategyRules> listAdditionalVersionStrategyRules = new ArrayList<>();
listAdditionalVersionStrategyRules.add(
 new VersionStrategyRules()
 .withRuleType(VersionStrategyRules.RuleTypeEnum.fromValue("Header"))
 .withParam("version")
 .withOp(VersionStrategyRules.OpEnum.fromValue("="))
 .withValue("v1")
);
VersionStrategy additionalVersionStrategyAdditionalVersionStrategy = new VersionStrategy();
additionalVersionStrategyAdditionalVersionStrategy.withRules(listAdditionalVersionStrategyRules)
 .withCombineType(VersionStrategy.CombineTypeEnum.fromValue("and"));
Map<String, VersionStrategy> listbodyAdditionalVersionStrategy = new HashMap<>();
listbodyAdditionalVersionStrategy.put("v2", additionalVersionStrategyAdditionalVersionStrategy);
body.withAdditionalVersionStrategy(listbodyAdditionalVersionStrategy);
body.withVersion("v1");
body.withName("a1");
request.withBody(body);
try {
 CreateVersionAliasResponse response = client.createVersionAlias(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

- 为函数的v1版本创建别名a1。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 # environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 # environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateVersionAliasRequest()
 request.body = CreateVersionAliasRequestBody(
 version="v1",
```

```
 name="a1"
)
 response = client.create_version_alias(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

- 为函数的v1版本创建别名a1，且开启灰度版本，将v2版本设置为50%随机灰度。  
# coding: utf-8

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 # environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 # environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateVersionAliasRequest()
 listAdditionalVersionWeightsbody = {
 "v2": 50
 }
 request.body = CreateVersionAliasRequestBody(
 additional_version_weights=listAdditionalVersionWeightsbody,
 version="v1",
 name="a1"
)
 response = client.create_version_alias(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

- 为函数的v1版本创建别名a1，且开启灰度版本，将v2版本设置为指定规则灰度。  
# coding: utf-8

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 # environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 # environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")
```

```
credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = CreateVersionAliasRequest()
 listRulesAdditionalVersionStrategy = [
 VersionStrategyRules(
 rule_type="Header",
 param="version",
 op="=",
 value="v1"
)
]
 additionalVersionStrategyAdditionalVersionStrategy = VersionStrategy(
 rules=listRulesAdditionalVersionStrategy,
 combine_type="and"
)
 listAdditionalVersionStrategybody = {
 "v2": additionalVersionStrategyAdditionalVersionStrategy
 }
 request.body = CreateVersionAliasRequestBody(
 additional_version_strategy=listAdditionalVersionStrategybody,
 version="v1",
 name="a1"
)
 response = client.create_version_alias(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

- 为函数的v1版本创建别名a1。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
```

```
 WithCredential(auth).
 Build()

 request := &model.CreateVersionAliasRequest{}
 request.Body = &model.CreateVersionAliasRequestBody{
 Version: "v1",
 Name: "a1",
 }
 response, err := client.CreateVersionAlias(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

- 为函数的v1版本创建别名a1，且开启灰度版本，将v2版本设置为50%随机灰度。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build()

 request := &model.CreateVersionAliasRequest{}
 var listAdditionalVersionWeightsbody = map[string]int32{
 "v2": int32(50),
 }
 request.Body = &model.CreateVersionAliasRequestBody{
 AdditionalVersionWeights: listAdditionalVersionWeightsbody,
 Version: "v1",
 Name: "a1",
 }
 response, err := client.CreateVersionAlias(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

- 为函数的v1版本创建别名a1，且开启灰度版本，将v2版本设置为指定规则灰度。

```
package main

import (
 "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.CreateVersionAliasRequest{}
 ruleTypeRules:= model.GetVersionStrategyRulesRuleTypeEnum().HEADER
 paramRules:= "version"
 opRules:= model.GetVersionStrategyRulesOpEnum().=
 valueRules:= "v1"
 var listRulesAdditionalVersionStrategy = []model.VersionStrategyRules{
 {
 RuleType: &ruleTypeRules,
 Param: ¶mRules,
 Op: &opRules,
 Value: &valueRules,
 },
 }
 combineTypeAdditionalVersionStrategy:= model.GetVersionStrategyCombineTypeEnum().AND
 additionalVersionStrategyAdditionalVersionStrategy := model.VersionStrategy{
 Rules: &listRulesAdditionalVersionStrategy,
 CombineType: &combineTypeAdditionalVersionStrategy,
 }
 var listAdditionalVersionStrategybody = map[string](model.VersionStrategy){
 "v2": additionalVersionStrategyAdditionalVersionStrategy,
 }
 request.Body = &model.CreateVersionAliasRequestBody{
 AdditionalVersionStrategy: listAdditionalVersionStrategybody,
 Version: "v1",
 Name: "a1",
 }
 response, err := client.CreateVersionAlias(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

### 7.7.3 删除函数版本别名

#### 功能介绍

删除函数版本别名。

#### 调用方法

请参见[如何调用API](#)。

#### URI

DELETE /v2/{project\_id}/fgs/functions/{function\_urn}/aliases/{alias\_name}

表 7-400 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |
| alias_name   | 是    | String | 要删除的别名名称。                                |



## 请求参数

表 7-401 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 401

表 7-402 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-403 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-404 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-405 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

删除指定版本别名。

```
DELETE https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class DeleteVersionAliasSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 DeleteVersionAliasRequest request = new DeleteVersionAliasRequest();
 try {
 DeleteVersionAliasResponse response = client.deleteVersionAlias(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
```

```
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = DeleteVersionAliasRequest()
 response = client.delete_version_alias(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
```

```

WithSk(sk).
Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.DeleteVersionAliasRequest{}
response, err := client.DeleteVersionAlias(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 204 | No Content                    |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

### 7.7.4 获取函数版本的指定别名信息

#### 功能介绍

获取函数指定的版本别名信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/aliases/{alias\_name}

表 7-406 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |
| alias_name   | 是    | String | 要查询的别名名称。                                |

## 请求参数

表 7-407 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-408 响应 Body 参数

| 参数                          | 参数类型                        | 描述         |
|-----------------------------|-----------------------------|------------|
| name                        | String                      | 要获取的别名名称。  |
| version                     | String                      | 别名对应的版本名称。 |
| description                 | String                      | 别名描述信息。    |
| last_modified               | String                      | 别名最后修改时间。  |
| alias_urn                   | String                      | 版本别名唯一标识。  |
| additional_version_weights  | Map<String,Integer>         | 百分比灰度配置信息  |
| additional_version_strategy | Map<String,VersionStrategy> | 指定规则灰度策略信息 |

表 7-409 VersionStrategy

| 参数           | 参数类型                                         | 描述                                                                                                                 |
|--------------|----------------------------------------------|--------------------------------------------------------------------------------------------------------------------|
| rules        | Array of <b>VersionStrategyRules</b> objects | 规则列表                                                                                                               |
| combine_type | String                                       | 所有规则聚合方式。and: 所有规则都满足, or: 满足其中一个<br>枚举值:<br><ul style="list-style-type: none"> <li>• and</li> <li>• or</li> </ul> |

表 7-410 VersionStrategyRules

| 参数        | 参数类型   | 描述                                                                                                      |
|-----------|--------|---------------------------------------------------------------------------------------------------------|
| rule_type | String | 参数类型。<br>枚举值:<br><ul style="list-style-type: none"> <li>• Header</li> </ul>                             |
| param     | String | 规则参数名, 只支持大小写字母, 数字, 下划线, 中划线。                                                                          |
| op        | String | 规则匹配操作符, 目前仅需支持 = 或者 in。<br>枚举值:<br><ul style="list-style-type: none"> <li>• in</li> <li>• =</li> </ul> |
| value     | String | 规则值, 如果op为in, 则为逗号分隔的多值字符串                                                                              |

状态码: 401

表 7-411 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码: 403

表 7-412 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-413 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-414 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询指定函数的版本别名信息。

```
GET https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
```

## 响应示例

状态码： 200

OK

```
{
 "name": "dev",
 "version": "latest",
 "description": "my dev version",
 "last_modified": "2019-10-31 11:37:58",
 "alias_urn": "urn:fs:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:ldev",
 "additional_version_weights": {
 "v1": 10
 }
}
```

状态码： 401

Unauthorized 鉴权失败。

```
{
 "error_code" : "FSS.1053",
 "error_msg" : "Not found the function alias"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowVersionAliasSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowVersionAliasRequest request = new ShowVersionAliasRequest();
 try {
 ShowVersionAliasResponse response = client.showVersionAlias(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
```



```
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowVersionAliasRequest()
 response = client.show_version_alias(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowVersionAliasRequest{}
 response, err := client.ShowVersionAlias(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.7.5 修改函数版本别名信息

### 功能介绍

修改函数版本别名信息。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/fgs/functions/{function\_urn}/aliases/{alias\_name}

表 7-415 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |
| alias_name   | 是    | String | 要更新的别名名称。                                |

## 请求参数

表 7-416 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

表 7-417 请求 Body 参数

| 参数                          | 是否必选 | 参数类型                        | 描述         |
|-----------------------------|------|-----------------------------|------------|
| version                     | 是    | String                      | 别名对应的版本名称。 |
| description                 | 否    | String                      | 别名描述信息。    |
| additional_version_weights  | 否    | Map<String,Integer>         | 灰度版本信息     |
| additional_version_strategy | 否    | Map<String,VersionStrategy> | 指定规则灰度策略信息 |

表 7-418 VersionStrategy

| 参数           | 是否必选 | 参数类型                                  | 描述                                                                                                       |
|--------------|------|---------------------------------------|----------------------------------------------------------------------------------------------------------|
| rules        | 否    | Array of VersionStrategyRules objects | 规则列表                                                                                                     |
| combine_type | 否    | String                                | 所有规则聚合方式。and：所有规则都满足，or：满足其中一个枚举值： <ul style="list-style-type: none"> <li>• and</li> <li>• or</li> </ul> |

表 7-419 VersionStrategyRules

| 参数        | 是否必选 | 参数类型   | 描述                                             |
|-----------|------|--------|------------------------------------------------|
| rule_type | 否    | String | 参数类型。<br>枚举值：<br>• Header                      |
| param     | 否    | String | 规则参数名，只支持大小写字母，数字，下划线，中划线。                     |
| op        | 否    | String | 规则匹配操作符，目前仅需支持 = 或者 in。<br>枚举值：<br>• in<br>• = |
| value     | 否    | String | 规则值，如果op为in，则为逗号分隔的多值字符串                       |

## 响应参数

状态码： 200

表 7-420 响应 Body 参数

| 参数                         | 参数类型                | 描述         |
|----------------------------|---------------------|------------|
| name                       | String              | 要获取的别名名称。  |
| version                    | String              | 别名对应的版本名称。 |
| description                | String              | 别名描述信息。    |
| last_modified              | String              | 别名最后修改时间。  |
| alias_urn                  | String              | 版本别名唯一标识。  |
| additional_version_weights | Map<String,Integer> | 灰度版本信息     |

状态码： 400

表 7-421 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 401**

**表 7-422 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 403**

**表 7-423 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 404**

**表 7-424 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 500**

**表 7-425 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

- 更新函数别名{alias\_name}绑定的函数版本为latest，且配置v1为灰度版本，流量比例为9:1。

```
PUT https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}
```

```
{
 "version": "latest",
 "description": ""
}
```

```
"additional_version_weights": {
 "v1": 10
}
```

- 更新函数别名{alias\_name}绑定的函数版本为v1，且开启灰度版本，将v2版本设置为指定规则灰度。

PUT https://{Endpoint}/v2/{project\_id}/fgs/functions/{function\_urn}/aliases/{alias\_name}

```
{
 "version": "v1",
 "description": "",
 "additional_version_strategy": {
 "v2": {
 "combine_type": "and",
 "rules": [{
 "rule_type": "Header",
 "param": "version",
 "op": "=",
 "value": "v1"
 }]
 }
 }
}
```

## 响应示例

**状态码： 200**

ok

```
{
 "name": "dev",
 "version": "latest",
 "description": "",
 "last_modified": "2019-10-31 11:37:58",
 "alias_urn": "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:!dev",
 "additional_version_weights": {
 "v1": 10
 }
}
```

**状态码： 404**

Not Found 找不到资源。

```
{
 "error_code": "FSS.1051",
 "error_msg": "Not found the function"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

- 更新函数别名{alias\_name}绑定的函数版本为latest，且配置v1为灰度版本，流量比例为9:1。

```
package com.huaweicloud.sdk.test;
```

```
import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
```

```
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.Map;
import java.util.HashMap;

public class UpdateVersionAliasSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 UpdateVersionAliasRequest request = new UpdateVersionAliasRequest();
 UpdateVersionAliasRequestBody body = new UpdateVersionAliasRequestBody();
 Map<String, Integer> listbodyAdditionalVersionWeights = new HashMap<>();
 listbodyAdditionalVersionWeights.put("v1", 10);
 body.withAdditionalVersionWeights(listbodyAdditionalVersionWeights);
 body.withDescription("");
 body.withVersion("latest");
 request.withBody(body);
 try {
 UpdateVersionAliasResponse response = client.updateVersionAlias(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

- 更新函数别名{alias\_name}绑定的函数版本为v1，且开启灰度版本，将v2版本设置为指定规则灰度。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateVersionAliasSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
UpdateVersionAliasRequest request = new UpdateVersionAliasRequest();
UpdateVersionAliasRequestBody body = new UpdateVersionAliasRequestBody();
body.withDescription("");
body.withVersion("v1");
request.withBody(body);
try {
 UpdateVersionAliasResponse response = client.updateVersionAlias(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

- 更新函数别名{alias\_name}绑定的函数版本为latest，且配置v1为灰度版本，流量比例为9:1。

# coding: utf-8

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = UpdateVersionAliasRequest()
 listAdditionalVersionWeightsbody = {
```



```
"v1": 10
}
request.body = UpdateVersionAliasRequestBody(
 additional_version_weights=listAdditionalVersionWeightsbody,
 description="",
 version="latest"
)
response = client.update_version_alias(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

- 更新函数别名{alias\_name}绑定的函数版本为v1，且开启灰度版本，将v2版本设置为指定规则灰度。

```
coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = UpdateVersionAliasRequest()
 request.body = UpdateVersionAliasRequestBody(
 description="",
 version="v1"
)
 response = client.update_version_alias(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

- 更新函数别名{alias\_name}绑定的函数版本为latest，且配置v1为灰度版本，流量比例为9:1。

```
package main
```

```
import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)
```

```
func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.UpdateVersionAliasRequest{}
 var listAdditionalVersionWeightsbody = map[string]int32{
 "v1": int32(10),
 }
 descriptionUpdateVersionAliasRequestBody := ""
 request.Body = &model.UpdateVersionAliasRequestBody{
 AdditionalVersionWeights: listAdditionalVersionWeightsbody,
 Description: &descriptionUpdateVersionAliasRequestBody,
 Version: "latest",
 }
 response, err := client.UpdateVersionAlias(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

- 更新函数别名{alias\_name}绑定的函数版本为v1，且开启灰度版本，将v2版本设置为指定规则灰度。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
```

```

Build()
{
 request := &model.UpdateVersionAliasRequest{
 descriptionUpdateVersionAliasRequestBody:= ""
 request.Body = &model.UpdateVersionAliasRequestBody{
 Description: &descriptionUpdateVersionAliasRequestBody,
 Version: "v1",
 }
 }
 response, err := client.UpdateVersionAlias(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.7.6 获取指定函数的版本列表

### 功能介绍

获取指定函数的版本列表。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/versions

表 7-426 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

表 7-427 Query 参数

| 参数       | 是否必选 | 参数类型   | 描述               |
|----------|------|--------|------------------|
| marker   | 否    | String | 上一次查询到的最后的记录位置。  |
| maxitems | 否    | String | 每次查询获取的最大函数记录数量。 |

## 请求参数

表 7-428 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-429 响应 Body 参数

| 参数          | 参数类型                                                       | 描述      |
|-------------|------------------------------------------------------------|---------|
| versions    | Array of <a href="#">ListFunctionVersionResult</a> objects | 版本列表    |
| next_marker | Long                                                       | 下一次记录位置 |

| 参数    | 参数类型 | 描述   |
|-------|------|------|
| count | Long | 版本总数 |

表 7-430 ListFunctionVersionResult

| 参数           | 参数类型   | 描述                                         |
|--------------|--------|--------------------------------------------|
| func_urn     | String | 函数的URN ( Uniform Resource Name ) , 唯一标识函数。 |
| func_name    | String | 函数名称。                                      |
| domain_id    | String | 域名id。                                      |
| namespace    | String | 租户的project id。                             |
| project_name | String | 租户的project name。                           |
| package      | String | 函数所属的分组Package, 用于用户针对函数的自定义分组。            |

| 参数      | 参数类型    | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| runtime | String  | <p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>Java8</b></li> <li>● <b>Java11</b></li> <li>● <b>Node.js6.10</b></li> <li>● <b>Node.js8.10</b></li> <li>● <b>Node.js10.16</b></li> <li>● <b>Node.js12.13</b></li> <li>● <b>Node.js14.18</b></li> <li>● <b>Node.js16.17</b></li> <li>● <b>Node.js18.15</b></li> <li>● <b>Python2.7</b></li> <li>● <b>Python3.6</b></li> <li>● <b>Python3.10</b></li> <li>● <b>Go1.x</b></li> <li>● <b>C#(.NET Core 2.1)</b></li> <li>● <b>C#(.NET Core 3.1)</b></li> <li>● <b>Custom</b></li> <li>● <b>PHP7.3</b></li> <li>● <b>Python3.9</b></li> <li>● <b>http</b></li> <li>● <b>Custom Image</b></li> <li>● <b>Cangjie1.0</b></li> </ul> |
| timeout | Integer | <p>函数执行超时时间，超时函数将被强行停止，范围3~259200秒。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| 参数                  | 参数类型    | 描述                                                                                                                                                                                                                                                                         |
|---------------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| handler             | String  | 函数执行入口 规则：xx.xx，必须包含“.” 举例：对于node.js函数：myfunction.handler，则表示函数的文件名为myfunction.js，执行的入口函数名为handler。长度不超过128个字符。<br>最小长度：3<br>最大长度：128                                                                                                                                      |
| memory_size         | Integer | 函数消耗的内存。单位M。取值范围为：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128，最大值为4096。                                                                                                                                                                           |
| cpu                 | Integer | 函数占用的cpu资源。单位为millicore（1 core=1000 millicores）。取值与MemorySize成比例，默认是128M内存占0.1个核（100 millicores）。                                                                                                                                                                          |
| code_type           | String  | 函数代码类型，取值有5种。inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包，主要针对Java函数。Custom-Image-Swr: 函数代码来源于SWR自定义镜像。<br>枚举值：<br><ul style="list-style-type: none"> <li>• inline</li> <li>• zip</li> <li>• obs</li> <li>• jar</li> <li>• Custom-Image-Swr</li> </ul> |
| code_url            | String  | 当code_type为obs时，该值为函数代码包在OBS上的地址，code_type为其他值时，该字段为空。                                                                                                                                                                                                                     |
| code_filename       | String  | 函数的文件名，当code_type为jar/zip时必须提供该字段，inline和obs不需要提供。                                                                                                                                                                                                                         |
| code_size           | Long    | 函数大小，单位：字节。                                                                                                                                                                                                                                                                |
| user_data           | String  | 用户自定义的name/value信息。在函数中使用的参数。举例：如函数要访问某个主机，可以设置自定义参数：Host={host_ip}，最多定义20个，总长度不超过4KB。                                                                                                                                                                                     |
| encrypted_user_data | String  | 用户自定义的name/value信息，用于需要加密的配置。                                                                                                                                                                                                                                              |
| digest              | String  | 函数代码SHA512 hash值，用于判断函数是否变化。                                                                                                                                                                                                                                               |
| version             | String  | 函数版本号，由系统自动生成，规则：vYYYYMMDD-HHMMSS（v+年月日-时分秒）。                                                                                                                                                                                                                              |

| 参数                    | 参数类型                              | 描述                                                                                                                                    |
|-----------------------|-----------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| image_name            | String                            | 函数版本的内部标识。                                                                                                                            |
| xrole                 | String                            | 函数配置委托。需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。配置后用户可以通过函数执行入口方法中的context参数获取具有委托中权限的token、ak、sk，用于访问其他云服务。如果用户函数不访问任何云服务，则不用提供委托名称。 |
| app_xrole             | String                            | 函数执行委托。可为函数执行单独配置执行委托，这将减小不必要的性能损耗；不单独配置执行委托时，函数执行和函数配置将使用同一委托。                                                                       |
| last_modified         | String                            | 函数最后一次更新时间。                                                                                                                           |
| func_vpc_id           | String                            | 用户的vpcid                                                                                                                              |
| concurrency           | Integer                           | 0：函数被禁用；-1：函数被启用。                                                                                                                     |
| concurrent_num        | Integer                           | 并发实例数                                                                                                                                 |
| strategy_config       | <b>StrategyConfig</b> object      | 函数策略配置。                                                                                                                               |
| initializer_handler   | String                            | 函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。举例：对于node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。         |
| initializer_timeout   | Integer                           | 初始化超时时间，超时函数将被强行停止，范围1~300秒。当配置初始化函数时，此参数必填。                                                                                          |
| pre_stop_handler      | String                            | 函数预停止函数的入口，规则：xx.xx，必须包含“.”。举例：对于node.js函数：myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。            |
| pre_stop_timeout      | Integer                           | 初始化超时时间，超时函数将被强行停止，范围1~90秒。                                                                                                           |
| long_time             | Boolean                           | 是否是支持长时间运行                                                                                                                            |
| function_async_config | <b>FunctionAsyncConfig</b> object | 函数异步配置返回结构体。                                                                                                                          |
| type                  | String                            | 函数版本                                                                                                                                  |
| enable_dynamic_memory | Boolean                           | 是否启用动态内存功能                                                                                                                            |



| 参数                          | 参数类型                      | 描述                   |
|-----------------------------|---------------------------|----------------------|
| enterprise_project_id       | String                    | 企业项目ID，在企业用户创建函数时必填。 |
| is_stateful_function        | Boolean                   | 是否支持有状态，v2版本支持       |
| enable_auth_header          | Boolean                   | 是否允许在请求头中添加鉴权信息      |
| custom_image                | <b>CustomImage</b> object | 用户容器镜像。              |
| reserved_instance_idle_mode | Boolean                   | 是否开启预留实例闲置模式         |

表 7-431 StrategyConfig

| 参数             | 参数类型    | 描述                                                    |
|----------------|---------|-------------------------------------------------------|
| concurrency    | Integer | 单函数最大实例数，v1取值0和-1，v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用 |
| concurrent_num | Integer | 单实例并发数，v2版本才支持，取值1到1000                               |

表 7-432 FunctionAsyncConfig

| 参数                             | 参数类型                                     | 描述                               |
|--------------------------------|------------------------------------------|----------------------------------|
| max_async_event_age_in_seconds | Integer                                  | 消息最大存活时长，取值范围[60, 86400]。单位：秒。   |
| max_async_retry_attempts       | Integer                                  | 异步调用失败后的最大重试次数，默认值为3。取值范围[0, 8]。 |
| destination_config             | <b>FuncAsyncDestinationConfig</b> object | 异步调用目标。                          |
| created_time                   | String                                   | 异步调用配置的创建时间。                     |
| last_modified                  | String                                   | 异步调用配置的最后更改时间。                   |

表 7-433 FuncAsyncDestinationConfig

| 参数         | 参数类型                         | 描述                                     |
|------------|------------------------------|----------------------------------------|
| on_success | FuncDestinationConfig object | 当函数执行成功时，函数计算将调用该配置对应的目标。              |
| on_failure | FuncDestinationConfig object | 当函数执行失败（系统错误或函数内部错误）时，函数计算将调用该配置对应的目标。 |

表 7-434 FuncDestinationConfig

| 参数          | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                |
|-------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| destination | String | 目标类型。<br><ul style="list-style-type: none"> <li>• OBS：通知到OBS服务。</li> <li>• SMN：通知到SMN服务。</li> <li>• DIS：通知到DIS服务。</li> <li>• FunctionGraph：通知到函数服务。</li> </ul> 枚举值：<br><ul style="list-style-type: none"> <li>• <b>OBS</b></li> <li>• <b>SMN</b></li> <li>• <b>DIS</b></li> <li>• <b>FunctionGraph</b></li> </ul> |
| param       | String | 通知目标服务对应参数,json字符串。<br><ul style="list-style-type: none"> <li>• OBS：包含bucket桶，对象目录前缀prefix，对象默认expires过期时间[0~365]天，0默认不过期。</li> <li>• SMN：包含smn 主题topic_urn。</li> <li>• DIS：包含DIS 通道名stream_name。</li> <li>• FunctionGraph：包含func_urn，函数urn</li> </ul>                                                            |

表 7-435 CustomImage

| 参数      | 参数类型    | 描述           |
|---------|---------|--------------|
| enabled | Boolean | 是否启用         |
| image   | String  | 镜像地址         |
| command | String  | 启动容器镜像的命令    |
| args    | String  | 启动容器镜像的命令行参数 |

| 参数          | 参数类型   | 描述         |
|-------------|--------|------------|
| working_dir | String | 镜像容器工作目录   |
| uid         | String | 镜像容器的用户id  |
| gid         | String | 镜像容器的用户组id |

状态码： 400

表 7-436 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-437 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-438 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-439 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-440 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询函数列表。

GET https://{Endpoint}/v2/{project\_id}/fgs/functions/{function\_urn}/versions

## 响应示例

状态码： 200

OK

```
{
 "versions": [{
 "func_urn": "urn:fs:xxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",
 "func_name": "test",
 "domain_id": "14ee2e3501124efcbca7998baa24xxxx",
 "namespace": "46b6f338fc3445b8846c71dfb1fbxxxx",
 "project_name": "xxxx",
 "package": "default",
 "runtime": "Node.js6.10",
 "timeout": 3,
 "handler": "test.handler",
 "memory_size": 128,
 "cpu": 300,
 "code_type": "inline",
 "code_filename": "index.js",
 "code_size": 272,
 "digest":
"faa825575c45437cddd4e369bea69893bcbe195d478178462ad90984fe72993f3f59d15f41c5373f807f3e05fb9af322c55dabeb16565c386e402413458e6068",
 "version": "latest",
 "image_name": "latest-191025153727@zehht",
 "last_modified": "2019-10-25 15:37:27",
 "strategy_config": {
 "concurrency": 0
 }
 }],
 "next_marker": 5
}
```

状态码： 404

Not Found 找不到资源。

```
{
 "error_code": "FSS.1051",
 "error_msg": "Not found the function"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListFunctionVersionsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListFunctionVersionsRequest request = new ListFunctionVersionsRequest();
 request.withMarker("<marker>");
 request.withMaxItems("<maxitems>");
 try {
 ListFunctionVersionsResponse response = client.listFunctionVersions(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = ListFunctionVersionsRequest()
 request.marker = "<marker>"
 request.maxitems = "<maxitems>"
 response = client.list_function_versions(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListFunctionVersionsRequest{}
 markerRequest := "<marker>"
 request.Marker = &markerRequest
 maxitemsRequest := "<maxitems>"
 request.Maxitems = &maxitemsRequest
 response, err := client.ListFunctionVersions(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

```
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.7.7 发布函数版本

### 功能介绍

发布函数版本。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/fgs/functions/{function\_urn}/versions

表 7-441 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

## 请求参数

表 7-442 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

表 7-443 请求 Body 参数

| 参数          | 是否必选 | 参数类型   | 描述     |
|-------------|------|--------|--------|
| digest      | 否    | String | md5键值  |
| version     | 否    | String | 发布版本名称 |
| description | 否    | String | 发布版本描述 |

## 响应参数

状态码： 200

表 7-444 响应 Body 参数

| 参数           | 参数类型   | 描述                                    |
|--------------|--------|---------------------------------------|
| func_urn     | String | 函数的URN（Uniform Resource Name），唯一标识函数。 |
| func_name    | String | 函数名称。                                 |
| domain_id    | String | 域名id。                                 |
| namespace    | String | 租户的project id。                        |
| project_name | String | 租户的project name。                      |
| package      | String | 函数所属的分组Package，用于用户针对函数的自定义分组。        |



| 参数      | 参数类型    | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| runtime | String  | <p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>Java8</b></li> <li>● <b>Java11</b></li> <li>● <b>Node.js6.10</b></li> <li>● <b>Node.js8.10</b></li> <li>● <b>Node.js10.16</b></li> <li>● <b>Node.js12.13</b></li> <li>● <b>Node.js14.18</b></li> <li>● <b>Node.js16.17</b></li> <li>● <b>Node.js18.15</b></li> <li>● <b>Python2.7</b></li> <li>● <b>Python3.6</b></li> <li>● <b>Python3.10</b></li> <li>● <b>Go1.x</b></li> <li>● <b>C#(.NET Core 2.1)</b></li> <li>● <b>C#(.NET Core 3.1)</b></li> <li>● <b>Custom</b></li> <li>● <b>PHP7.3</b></li> <li>● <b>Python3.9</b></li> <li>● <b>http</b></li> <li>● <b>Custom Image</b></li> <li>● <b>Cangjie1.0</b></li> </ul> |
| timeout | Integer | <p>函数执行超时时间，超时函数将被强行停止，范围3~259200秒。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| 参数                  | 参数类型    | 描述                                                                                                                                                                                                                                                                           |
|---------------------|---------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| handler             | String  | 函数执行入口 规则: xx.xx, 必须包含 “.” 举例: 对于node.js函数: myfunction.handler, 则表示函数的文件名为myfunction.js, 执行的入口函数名为handler。长度不超过128个字符。<br>最小长度: 3<br>最大长度: 128                                                                                                                               |
| memory_size         | Integer | 函数消耗的内存。单位M。取值范围为: 128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128, 最大值为4096。                                                                                                                                                                           |
| cpu                 | Integer | 函数占用的cpu资源。单位为millicore ( 1 core=1000 millicores )。取值与MemorySize成比例, 默认是128M内存占0.1个核 ( 100 millicores )。                                                                                                                                                                     |
| code_type           | String  | 函数代码类型, 取值有5种。inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包, 主要针对Java函数。Custom-Image-Swr: 函数代码来源于SWR自定义镜像。<br>枚举值:<br><ul style="list-style-type: none"> <li>• inline</li> <li>• zip</li> <li>• obs</li> <li>• jar</li> <li>• Custom-Image-Swr</li> </ul> |
| code_url            | String  | 当code_type为obs时, 该值为函数代码包在OBS上的地址, code_type为其他值时, 该字段为空。                                                                                                                                                                                                                    |
| code_filename       | String  | 函数的文件名, 当code_type为jar/zip时必须提供该字段, inline和obs不需要提供。                                                                                                                                                                                                                         |
| code_size           | Long    | 函数大小, 单位: 字节。                                                                                                                                                                                                                                                                |
| user_data           | String  | 用户自定义的name/value信息。在函数中使用的参数。举例: 如函数要访问某个主机, 可以设置自定义参数: Host={host_ip}, 最多定义20个, 总长度不超过4KB。                                                                                                                                                                                  |
| encrypted_user_data | String  | 用户自定义的name/value信息, 用于需要加密的配置。                                                                                                                                                                                                                                               |
| digest              | String  | 函数代码SHA512 hash值, 用于判断函数是否变化。                                                                                                                                                                                                                                                |
| version             | String  | 函数版本号, 由系统自动生成, 规则: vYYYYMMDD-HHMMSS ( v+年月日-时分秒 )。                                                                                                                                                                                                                          |

| 参数                    | 参数类型                               | 描述                                                                                                                                    |
|-----------------------|------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------|
| image_name            | String                             | 函数版本的内部标识。                                                                                                                            |
| xrole                 | String                             | 函数配置委托。需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。配置后用户可以通过函数执行入口方法中的context参数获取具有委托中权限的token、ak、sk，用于访问其他云服务。如果用户函数不访问任何云服务，则不用提供委托名称。 |
| app_xrole             | String                             | 函数执行委托。可为函数执行单独配置执行委托，这将减小不必要的性能损耗；不单独配置执行委托时，函数执行和函数配置将使用同一委托。                                                                       |
| description           | String                             | 函数描述。                                                                                                                                 |
| version_description   | String                             | 函数版本描述。                                                                                                                               |
| last_modified         | String                             | 函数最后一次更新时间。                                                                                                                           |
| func_vpc              | <b>FuncVpc</b> object              | 函数vpc配置。需同时为函数配置具有vpc权限的委托。                                                                                                           |
| mount_config          | <b>MountConfig</b> object          | 函数挂载配置。                                                                                                                               |
| strategy_config       | <b>StrategyConfig</b> object       | 函数策略配置。                                                                                                                               |
| dependencies          | Array of <b>Dependency</b> objects | 函数依赖代码包列表。                                                                                                                            |
| initializer_handler   | String                             | 函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。举例：对于node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。         |
| initializer_timeout   | Integer                            | 初始化超时时间，超时函数将被强行停止，范围1~300秒。当配置初始化函数时，此参数必填。                                                                                          |
| pre_stop_handler      | String                             | 函数预停止函数的入口，规则：xx.xx，必须包含“.”。举例：对于node.js函数：myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。            |
| pre_stop_timeout      | Integer                            | 初始化超时时间，超时函数将被强行停止，范围1~90秒。                                                                                                           |
| enterprise_project_id | String                             | 企业项目ID，在企业用户创建函数时必填。                                                                                                                  |

| 参数                    | 参数类型                              | 描述                                                                                           |
|-----------------------|-----------------------------------|----------------------------------------------------------------------------------------------|
| long_time             | Boolean                           | 是否允许进行长时间超时设置。                                                                               |
| log_group_id          | String                            | 自定义日志查询组id                                                                                   |
| log_stream_id         | String                            | 自定义日志查询流id                                                                                   |
| type                  | String                            | v2表示为正式版本,v1为废弃版本。<br>枚举值: <ul style="list-style-type: none"><li>• v1</li><li>• v2</li></ul> |
| enable_dynamic_memory | Boolean                           | 是否允许动态内存配置                                                                                   |
| function_async_config | <b>FunctionAsyncConfig</b> object | 函数异步配置返回结构体。                                                                                 |

表 7-445 FuncVpc

| 参数              | 参数类型             | 描述             |
|-----------------|------------------|----------------|
| domain_id       | String           | 域名id。          |
| namespace       | String           | 租户的project id。 |
| vpc_name        | String           | 虚拟私有云名称。       |
| vpc_id          | String           | 虚拟私有云唯一标识。     |
| subnet_name     | String           | 子网名称。          |
| subnet_id       | String           | 子网编号。          |
| cidr            | String           | 子网掩码。          |
| gateway         | String           | 网关。            |
| security_groups | Array of strings | 安全组            |

表 7-446 MountConfig

| 参数         | 参数类型                    | 描述      |
|------------|-------------------------|---------|
| mount_user | <b>MountUser</b> object | 挂载用户信息。 |

| 参数          | 参数类型                              | 描述      |
|-------------|-----------------------------------|---------|
| func_mounts | Array of <b>FuncMount</b> objects | 函数挂载列表。 |

表 7-447 MountUser

| 参数            | 参数类型   | 描述                   |
|---------------|--------|----------------------|
| user_id       | String | 用户ID(-1~65534的非0整数)  |
| user_group_id | String | 用户组ID(-1~65534的非0整数) |

表 7-448 FuncMount

| 参数               | 参数类型   | 描述                                                   |
|------------------|--------|------------------------------------------------------|
| mount_type       | String | 挂载类型(sfs/sfsTurbo/ecs), func_mounts非空时必选。            |
| mount_resource   | String | 挂载资源ID (对应云服务ID), func_mounts非空时必选。                  |
| mount_share_path | String | 远端挂载路径 (例如192.168.0.12:/data), 如果mount_type为ecs, 必选。 |
| local_mount_path | String | 函数访问路径, func_mounts非空时必选。                            |

表 7-449 StrategyConfig

| 参数             | 参数类型    | 描述                                                      |
|----------------|---------|---------------------------------------------------------|
| concurrency    | Integer | 单函数最大实例数, v1取值0和-1, v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用 |
| concurrent_num | Integer | 单实例并发数, v2版本才支持, 取值1到1000                               |

表 7-450 Dependency

| 参数    | 参数类型   | 描述              |
|-------|--------|-----------------|
| id    | String | 依赖包版本ID。        |
| owner | String | 依赖包属主的domainId。 |

| 参数      | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| link    | String | 依赖包在OBS上的链接。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
| runtime | String | <p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>Java8</b></li> <li>● <b>Java11</b></li> <li>● <b>Node.js6.10</b></li> <li>● <b>Node.js8.10</b></li> <li>● <b>Node.js10.16</b></li> <li>● <b>Node.js12.13</b></li> <li>● <b>Node.js14.18</b></li> <li>● <b>Node.js16.17</b></li> <li>● <b>Node.js18.15</b></li> <li>● <b>Python2.7</b></li> <li>● <b>Python3.6</b></li> <li>● <b>Python3.10</b></li> <li>● <b>Go1.x</b></li> <li>● <b>C#(.NET Core 2.1)</b></li> <li>● <b>C#(.NET Core 3.1)</b></li> <li>● <b>Custom</b></li> <li>● <b>PHP7.3</b></li> <li>● <b>Python3.9</b></li> <li>● <b>http</b></li> <li>● <b>Custom Image</b></li> <li>● <b>Cangjie1.0</b></li> </ul> |
| etag    | String | 依赖包的md5值                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

| 参数            | 参数类型    | 描述                  |
|---------------|---------|---------------------|
| size          | Long    | 依赖包大小。              |
| name          | String  | 依赖包名称。              |
| description   | String  | 依赖包描述。              |
| file_name     | String  | 依赖包文件名，如果创建方式为zip时。 |
| version       | Long    | 依赖包版本编号。            |
| dep_id        | String  | 依赖包ID               |
| last_modified | Integer | 函数最后一次更新时间。         |

表 7-451 FunctionAsyncConfig

| 参数                             | 参数类型                                              | 描述                               |
|--------------------------------|---------------------------------------------------|----------------------------------|
| max_async_event_age_in_seconds | Integer                                           | 消息最大存活时长，取值范围[60, 86400]。单位：秒。   |
| max_async_retry_attempts       | Integer                                           | 异步调用失败后的最大重试次数，默认值为3。取值范围[0, 8]。 |
| destination_config             | <a href="#">FuncAsyncDestinationConfig</a> object | 异步调用目标。                          |
| created_time                   | String                                            | 异步调用配置的创建时间。                     |
| last_modified                  | String                                            | 异步调用配置的最后更改时间。                   |

表 7-452 FuncAsyncDestinationConfig

| 参数         | 参数类型                                         | 描述                                     |
|------------|----------------------------------------------|----------------------------------------|
| on_success | <a href="#">FuncDestinationConfig</a> object | 当函数执行成功时，函数计算将调用该配置对应的目标。              |
| on_failure | <a href="#">FuncDestinationConfig</a> object | 当函数执行失败（系统错误或函数内部错误）时，函数计算将调用该配置对应的目标。 |

表 7-453 FuncDestinationConfig

| 参数          | 参数类型   | 描述                                                                                                                                                                                                                                                                                                          |
|-------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| destination | String | 目标类型。 <ul style="list-style-type: none"> <li>• OBS：通知到OBS服务。</li> <li>• SMN：通知到SMN服务。</li> <li>• DIS：通知到DIS服务。</li> <li>• FunctionGraph：通知到函数服务。</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• <b>OBS</b></li> <li>• <b>SMN</b></li> <li>• <b>DIS</b></li> <li>• <b>FunctionGraph</b></li> </ul> |
| param       | String | 通知目标服务对应参数,json字符串。 <ul style="list-style-type: none"> <li>• OBS：包含bucket桶，对象目录前缀prefix，对象默认expires过期时间[0~365]天，0默认不过期。</li> <li>• SMN：包含smn 主题topic_urn。</li> <li>• DIS：包含DIS 通道名stream_name。</li> <li>• FunctionGraph：包含func_urn，函数urn</li> </ul>                                                         |

状态码： 400

表 7-454 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-455 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403



表 7-456 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-457 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-458 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

发布函数版本。

```
POST https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/versions
```

```
{
 "version": "v1",
 "description": "v1"
}
```

## 响应示例

状态码： 200

OK

```
{
 "func_urn": "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",
 "func_name": "test",
 "domain_id": "14ee2e3501124efcbca7998baa24xxxx",
 "namespace": "46b6f338fc3445b8846c71dfb1fbxxxx",
 "project_name": "xxxxx",
 "package": "default",
 "runtime": "Node.js6.10",
 "timeout": 3,
}
```

```
"handler" : "test.handler",
"memory_size" : 128,
"cpu" : 300,
"code_type" : "inline",
"code_filename" : "inline.js",
"code_size" : 272,
"digest" :
"faa825575c45437cddd4e369bea69893bcbe195d478178462ad90984fe72993f3f59d15f41c5373f807f3e05fb9
af322c55dabeb16565c386e402413458e6068",
"version" : "latest",
"image_name" : "latest-191025153727@zehht",
"last_modified" : "2019-10-25 15:37:27",
"strategy_config" : {
 "concurrency" : 0
}
}
```

**状态码： 404**

Not Found 找不到资源。

```
{
 "error_code" : "FSS.1051",
 "error_msg" : "Not found the function"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

发布函数版本。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateFunctionVersionSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateFunctionVersionRequest request = new CreateFunctionVersionRequest();
 CreateFunctionVersionRequestBody body = new CreateFunctionVersionRequestBody();
```

```
body.withDescription("v1");
body.withVersion("v1");
request.withBody(body);
try {
 CreateFunctionVersionResponse response = client.createFunctionVersion(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

发布函数版本。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateFunctionVersionRequest()
 request.body = CreateFunctionVersionRequestBody(
 description="v1",
 version="v1"
)
 response = client.create_function_version(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

发布函数版本。

```
package main

import (
```

```

"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.CreateFunctionVersionRequest{
 descriptionCreateFunctionVersionRequestBody:= "v1"
 versionCreateFunctionVersionRequestBody:= "v1"
 request.Body = &model.CreateFunctionVersionRequestBody{
 Description: &descriptionCreateFunctionVersionRequestBody,
 Version: &versionCreateFunctionVersionRequestBody,
 }
 }
 response, err := client.CreateFunctionVersion(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.8 函数指标

### 7.8.1 获取按指定指标排序的函数列表

#### 功能介绍

按指定指标排序的函数列表。

默认统计按错误次数指标统计最近一天失败次数最多的前10个函数

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/fgs/function/report

表 7-459 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

表 7-460 Query 参数

| 参数         | 是否必选 | 参数类型   | 描述                  |
|------------|------|--------|---------------------|
| type       | 否    | String | 指标类型，默认值为failcount。 |
| start_time | 否    | String | 起始时间。               |
| end_time   | 否    | String | 结束时间。               |
| limit      | 否    | String | 指标类型，默认值为failcount。 |

## 请求参数

表 7-461 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-462 响应 Body 参数

| 参数          | 参数类型                                            | 描述         |
|-------------|-------------------------------------------------|------------|
| values      | Array of <a href="#">FunctionMetric</a> objects | 函数Urn及其指标数 |
| next_marker | Long                                            | 下次读取位置     |
| count       | Long                                            | 返回函数总数     |

表 7-463 FunctionMetric

| 参数    | 参数类型    | 描述    |
|-------|---------|-------|
| key   | String  | 函数urn |
| value | Integer | 指标值   |

状态码： 400

表 7-464 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

获取按指定指标排序的函数列表。

```
GET /v2/{project_id}/fgs/function/report
```

## 响应示例

状态码： 200

查询成功

```
{
 "next_marker" : 2,
 "count" : 2,
 "values" : [{
 "key" : "funcUrn1",
 "value" : 1
 }, {
 "key" : "funcUrn2",
 "value" : 2
 }]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListFunctionAsMetricSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListFunctionAsMetricRequest request = new ListFunctionAsMetricRequest();
 request.withType("<type>");
 request.withStartTime("<start_time>");
 request.withEndTime("<end_time>");
 request.withLimit("<limit>");
 }
}
```

```
try {
 ListFunctionAsMetricResponse response = client.listFunctionAsMetric(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListFunctionAsMetricRequest()
 request.type = "<type>"
 request.start_time = "<start_time>"
 request.end_time = "<end_time>"
 request.limit = "<limit>"
 response = client.list_function_as_metric(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
```



```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
// risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
// variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this
// example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.ListFunctionAsMetricRequest{}
typeRequest:= "<type>"
request.Type = &typeRequest
startTimeRequest:= "<start_time>"
request.StartTime = &startTimeRequest
endTimeRequest:= "<end_time>"
request.EndTime = &endTimeRequest
limitRequest:= "<limit>"
request.Limit = &limitRequest
response, err := client.ListFunctionAsMetric(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | 查询成功        |
| 400 | Bad Request |

## 错误码

请参见[错误码](#)。

## 7.8.2 租户函数统计信息

### 功能介绍

租户函数统计信息。

返回三类的统计信息，函数格式和大小使用情况包括配额和使用量，流量报告。通过查询参数filter可以进行过滤，查询参数period可以指定返回的时间段。

## 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/fgs/functions/statistics

表 7-465 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

表 7-466 Query 参数

| 参数     | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                         |
|--------|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| filter | 是    | String | 参数过滤器。monitor_data: 查询统计信息。<br>monthly_report: 查询月度统计信息。<br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>monitor_data</b></li> <li>• <b>monthly_report</b></li> </ul>                               |
| period | 否    | String | 时间段单位为分钟，与filter参数metric配合使用。                                                                                                                                                                              |
| option | 否    | String | 月度统计的维度，filter参数取值为monthly_report时才生效。当取值不在以上范围时，默认取"0"。<br><ul style="list-style-type: none"> <li>• "0": 表示统计本月。</li> <li>• "1": 表示统计上月。</li> <li>• "2": 表示统计最近三个月。</li> <li>• "3": 表示统计最近六个月。</li> </ul> |
| limit  | 否    | String | 本次查询最大返回的数据条数，最大值500，默认值100<br>缺省值： <b>100</b><br>最小长度： <b>1</b><br>最大长度： <b>64</b>                                                                                                                        |

| 参数     | 是否必选 | 参数类型   | 描述                                                                  |
|--------|------|--------|---------------------------------------------------------------------|
| marker | 否    | String | 本次查询起始位置，默认值0<br>缺省值： <b>0</b><br>最小长度： <b>1</b><br>最大长度： <b>64</b> |

## 请求参数

表 7-467 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-468 响应 Body 参数

| 参数         | 参数类型                                             | 描述        |
|------------|--------------------------------------------------|-----------|
| count      | Array of <b>MonthUsed</b> objects                | 月度调用次数    |
| gbs        | Array of <b>MonthUsed</b> objects                | 月度资源用量    |
| gpu_gbs    | Array of <b>MonthUsed</b> objects                | 月度gpu资源用量 |
| statistics | <b>ListFunctionStatisticsResponseBody</b> object | 函数指标      |

表 7-469 MonthUsed

| 参数    | 参数类型   | 描述  |
|-------|--------|-----|
| date  | String | 日期  |
| value | Number | 使用量 |

表 7-470 ListFunctionStatisticsResponseBody

| 参数                    | 参数类型                                             | 描述         |
|-----------------------|--------------------------------------------------|------------|
| count                 | Array of <a href="#">SlaReportsValue</a> objects | 调用次数       |
| duration              | Array of <a href="#">SlaReportsValue</a> objects | 平均时延, 单位毫秒 |
| fail_count            | Array of <a href="#">SlaReportsValue</a> objects | 错误次数       |
| max_duration          | Array of <a href="#">SlaReportsValue</a> objects | 最大时延, 单位毫秒 |
| min_duration          | Array of <a href="#">SlaReportsValue</a> objects | 最小时延, 单位毫秒 |
| reject_count          | Array of <a href="#">SlaReportsValue</a> objects | 被拒绝次数      |
| function_error_count  | Array of <a href="#">SlaReportsValue</a> objects | 函数错误次数     |
| system_error_count    | Array of <a href="#">SlaReportsValue</a> objects | 系统错误次数     |
| reserved_instance_num | Array of <a href="#">SlaReportsValue</a> objects | 预留实例指标     |
| concurrency_num       | Array of <a href="#">SlaReportsValue</a> objects | 弹性实例指标     |

表 7-471 SlaReportsValue

| 参数        | 参数类型    | 描述                                   |
|-----------|---------|--------------------------------------|
| timestamp | Integer | 时间戳                                  |
| value     | Double  | 值，value值为-1表示当前时间段该指标无数据，例如函数未执行等情况。 |

状态码： 400

表 7-472 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

租户函数统计信息。

```
GET /v2/{project_id}/fgs/functions/statistics
```

## 响应示例

状态码： 200

查询成功

```
{
 "statistics": [
 {
 "count": [
 {
 "timestamp": 1596679200000,
 "value": -1
 },
 {
 "timestamp": 1596682800000,
 "value": 2
 },
 {
 "timestamp": 1596686400000,
 "value": -1
 }
],
 "duration": [
 {
 "timestamp": 1596679200000,
 "value": -1
 },
 {
 "timestamp": 1596682800000,
 "value": 950
 },
 {
 "timestamp": 1596686400000,
 "value": -1
 }
],
 "fail_count": [
 {
 "timestamp": 1596679200000,
 "value": -1
 },
 {
 "timestamp": 1596682800000,
 "value": 0
 }
]
 }
]
}
```

```
 }, {
 "timestamp" : 1596686400000,
 "value" : -1
 }],
 "max_duration" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }],
 {
 "timestamp" : 1596682800000,
 "value" : 740
 }, {
 "timestamp" : 1596686400000,
 "value" : -1
 }],
 "min_duration" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }],
 {
 "timestamp" : 1596682800000,
 "value" : 210
 }, {
 "timestamp" : 1596686400000,
 "value" : -1
 }],
 "reject_count" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }],
 {
 "timestamp" : 1596682800000,
 "value" : 0
 }, {
 "timestamp" : 1596686400000,
 "value" : -1
 }]
 }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListStatisticsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
```

```
 .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
ListStatisticsRequest request = new ListStatisticsRequest();
request.withFilter(ListStatisticsRequest.FilterEnum.fromValue("<filter>"));
request.withPeriod("<period>");
request.withOption("<option>");
request.withLimit("<limit>");
request.withMarker("<marker>");
try {
 ListStatisticsResponse response = client.listStatistics(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListStatisticsRequest()
 request.filter = "<filter>"
 request.period = "<period>"
 request.option = "<option>"
 request.limit = "<limit>"
 request.marker = "<marker>"
 response = client.list_statistics(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListStatisticsRequest{}
 request.Filter = model.GetListStatisticsRequestFilterEnum().<FILTER>
 periodRequest:= "<period>"
 request.Period = &periodRequest
 optionRequest:= "<option>"
 request.Option = &optionRequest
 limitRequest:= "<limit>"
 request.Limit = &limitRequest
 markerRequest:= "<marker>"
 request.Marker = &markerRequest
 response, err := client.ListStatistics(request)
 if err == nil {
 fmt.Printf("%v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | 查询成功        |
| 400 | Bad Request |



## 错误码

请参见[错误码](#)。

## 7.8.3 获取指定时间段的函数运行指标

### 功能介绍

获取指定时间段的函数运行指标。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/{func\_urn}/statistics/{period}

表 7-473 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| func_urn   | 是    | String | 函数的URN，详细解释见 FunctionGraph函数模型的描述。       |
| period     | 是    | String | 获取最近多少分钟内函数执行的指标。                        |

### 请求参数

表 7-474 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                            |
|--------------|------|--------|---------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                    |

### 响应参数

状态码： 200

表 7-475 响应 Body 参数

| 参数                    | 参数类型                                             | 描述         |
|-----------------------|--------------------------------------------------|------------|
| count                 | Array of <a href="#">SlaReportsValue</a> objects | 调用次数       |
| duration              | Array of <a href="#">SlaReportsValue</a> objects | 平均时延, 单位毫秒 |
| fail_count            | Array of <a href="#">SlaReportsValue</a> objects | 错误次数       |
| max_duration          | Array of <a href="#">SlaReportsValue</a> objects | 最大时延, 单位毫秒 |
| min_duration          | Array of <a href="#">SlaReportsValue</a> objects | 最小时延, 单位毫秒 |
| reject_count          | Array of <a href="#">SlaReportsValue</a> objects | 被拒绝次数      |
| function_error_count  | Array of <a href="#">SlaReportsValue</a> objects | 函数错误次数     |
| system_error_count    | Array of <a href="#">SlaReportsValue</a> objects | 系统错误次数     |
| reserved_instance_num | Array of <a href="#">SlaReportsValue</a> objects | 预留实例指标     |
| concurrency_num       | Array of <a href="#">SlaReportsValue</a> objects | 弹性实例指标     |

表 7-476 SlaReportsValue

| 参数        | 参数类型    | 描述                                     |
|-----------|---------|----------------------------------------|
| timestamp | Integer | 时间戳                                    |
| value     | Double  | 值, value值为-1表示当前时间段该指标无数据, 例如函数未执行等情况。 |

## 请求示例

获取指定时间段的函数运行指标。

```
GET /v2/{project_id}/fgs/functions/{func_urn}/statistics/{period}
```

## 响应示例

状态码: 200

ok

```
{
 "count": [{
 "timestamp": 1596679200000,
 "value": -1
 }, {
 "timestamp": 1596682800000,
 "value": 2
 }, {
 "timestamp": 1596686400000,
 "value": -1
 }],
 "duration": [{
 "timestamp": 1596679200000,
 "value": -1
 }, {
 "timestamp": 1596682800000,
 "value": 950
 }, {
 "timestamp": 1596686400000,
 "value": -1
 }],
 "fail_count": [{
 "timestamp": 1596679200000,
 "value": -1
 }, {
 "timestamp": 1596682800000,
 "value": 0
 }, {
 "timestamp": 1596686400000,
 "value": -1
 }],
 "max_duration": [{
 "timestamp": 1596679200000,
 "value": -1
 }, {
 "timestamp": 1596682800000,
 "value": 740
 }, {
 "timestamp": 1596686400000,
 "value": -1
 }],
 "min_duration": [{
 "timestamp": 1596679200000,
 "value": -1
 }, {
 "timestamp": 1596682800000,
 "value": 210
 }, {
 "timestamp": 1596686400000,
 "value": -1
 }],
 "reject_count": [{
 "timestamp": 1596679200000,
 "value": -1
 }, {
 "timestamp": 1596682800000,
```

```
"value": 0
}, {
 "timestamp": 1596686400000,
 "value": -1
}]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListFunctionStatisticsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListFunctionStatisticsRequest request = new ListFunctionStatisticsRequest();
 try {
 ListFunctionStatisticsResponse response = client.listFunctionStatistics(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
```

```
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListFunctionStatisticsRequest()
 response = client.list_function_statistics(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListFunctionStatisticsRequest{}
 response, err := client.ListFunctionStatistics(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

```
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | ok |

## 错误码

请参见[错误码](#)。

## 7.8.4 查询函数实例使用情况指标

### 功能介绍

查询函数实例使用情况指标。

- 指标单位为分钟：当查询时间范围小于1小时,指标周期为1分钟 当查询时间范围小于1天,指标周期为30分钟 当查询时间范围大于1天,指标周期为180分钟
- 指标分为如下几类：reservedinstancenum（预留实例使用）、concurrency（实例使用/并发）

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/{func\_urn}/instancereports

表 7-477 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| func_urn   | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

表 7-478 Query 参数

| 参数     | 是否必选 | 参数类型   | 描述                                                                                  |
|--------|------|--------|-------------------------------------------------------------------------------------|
| marker | 否    | String | 本次查询起始位置，默认值0<br>缺省值： <b>0</b><br>最小长度： <b>0</b><br>最大长度： <b>64</b>                 |
| limit  | 否    | String | 本次查询最大返回的数据条数，最大值500，默认值100<br>缺省值： <b>100</b><br>最小长度： <b>0</b><br>最大长度： <b>64</b> |

## 请求参数

表 7-479 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-480 响应 Body 参数

| 参数                  | 参数类型                                             | 描述     |
|---------------------|--------------------------------------------------|--------|
| instanceNum         | Array of <a href="#">SlaReportsValue</a> objects | 弹性实例指标 |
| reservedInstanceNum | Array of <a href="#">SlaReportsValue</a> objects | 预留实例指标 |

表 7-481 SlaReportsValue

| 参数        | 参数类型    | 描述                                   |
|-----------|---------|--------------------------------------|
| timestamp | Integer | 时间戳                                  |
| value     | Double  | 值，value值为-1表示当前时间段该指标无数据，例如函数未执行等情况。 |

状态码： 500

表 7-482 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询函数实例使用情况指标。

```
GET /v2/{project_id}/fgs/functions/{func_urn}/instancereports
```

## 响应示例

状态码： 200

ok

```
{
 "instanceNum": [{
 "timestamp": 1596679200000,
 "value": -1
 }, {
 "timestamp": 1596682800000,
 "value": 2
 }, {
 "timestamp": 1596686400000,
 "value": -1
 }],
 "reservedInstanceNum": [{
 "timestamp": 1596679200000,
 "value": -1
 }, {
 "timestamp": 1596682800000,
 "value": 950
 }, {
 "timestamp": 1596686400000,
 "value": -1
 }]
}
```

## SDK 代码示例

SDK代码示例如下。



## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowFuncReservedInstanceMetricsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowFuncReservedInstanceMetricsRequest request = new ShowFuncReservedInstanceMetricsRequest();
 request.withMarker("<marker>");
 request.withLimit("<limit>");
 try {
 ShowFuncReservedInstanceMetricsResponse response =
 client.showFuncReservedInstanceMetrics(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = ShowFuncReservedInstanceMetricsRequest()
 request.marker = "<marker>"
 request.limit = "<limit>"
 response = client.show_func_reserved_instance_metrics(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowFuncReservedInstanceMetricsRequest{}
 markerRequest := "<marker>"
 request.Marker = &markerRequest
 limitRequest := "<limit>"
 request.Limit = &limitRequest
 response, err := client.ShowFuncReservedInstanceMetrics(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.8.5 查询函数实例流量指标

### 功能介绍

查询函数流量指标。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/{func\_urn}/slareports/{period}

表 7-483 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| func_urn   | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |
| period     | 是    | String | 时间间隔（单位：min）                             |

## 请求参数

表 7-484 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-485 响应 Body 参数

| 参数                   | 参数类型                                             | 描述        |
|----------------------|--------------------------------------------------|-----------|
| concurrency_num      | Array of <a href="#">SlaReportsValue</a> objects | 并发数       |
| count                | Array of <a href="#">SlaReportsValue</a> objects | 调用次数      |
| duration             | Array of <a href="#">SlaReportsValue</a> objects | 平均时延，单位毫秒 |
| fail_count           | Array of <a href="#">SlaReportsValue</a> objects | 错误次数      |
| function_error_count | Array of <a href="#">SlaReportsValue</a> objects | 函数错误次数    |
| system_error_count   | Array of <a href="#">SlaReportsValue</a> objects | 系统错误次数    |
| instance_num         | Array of <a href="#">SlaReportsValue</a> objects | 弹性实例数     |
| max_duration         | Array of <a href="#">SlaReportsValue</a> objects | 最大时延，单位毫秒 |

| 参数                    | 参数类型                                             | 描述        |
|-----------------------|--------------------------------------------------|-----------|
| min_duration          | Array of <a href="#">SlaReportsValue</a> objects | 最小时延，单位毫秒 |
| reject_count          | Array of <a href="#">SlaReportsValue</a> objects | 被拒绝次数     |
| reserved_instance_num | Array of <a href="#">SlaReportsValue</a> objects | 预留实例数     |

表 7-486 SlaReportsValue

| 参数        | 参数类型    | 描述                                      |
|-----------|---------|-----------------------------------------|
| timestamp | Integer | 时间戳                                     |
| value     | Double  | 值，value 值为 -1 表示当前时间段该指标无数据，例如函数未执行等情况。 |

状态码： 500

表 7-487 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询函数实例流量指标。

```
GET /v2/{project_id}/fgs/functions/{func_urn}/slareports/{period}
```

## 响应示例

状态码： 200

ok

```
{
 "concurrency_num": [{
 "timestamp": 1596679200000,
 "value": -1
 }, {
 "timestamp": 1596682800000,
 "value": 2
 }]
}
```

```
 }],
 "count" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }, {
 "timestamp" : 1596682800000,
 "value" : 950
 }],
 "duration" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }, {
 "timestamp" : 1596682800000,
 "value" : 950
 }],
 "fail_count" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }, {
 "timestamp" : 1596682800000,
 "value" : 950
 }],
 "function_error_count" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }, {
 "timestamp" : 1596682800000,
 "value" : 950
 }],
 "instance_num" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }, {
 "timestamp" : 1596682800000,
 "value" : 950
 }],
 "max_duration" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }, {
 "timestamp" : 1596682800000,
 "value" : 950
 }],
 "min_duration" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }, {
 "timestamp" : 1596682800000,
 "value" : 950
 }],
 "reject_count" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }, {
 "timestamp" : 1596682800000,
 "value" : 950
 }],
 "reserved_instance_num" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }, {
 "timestamp" : 1596682800000,
 "value" : 950
 }],
 "system_error_count" : [{
 "timestamp" : 1596679200000,
 "value" : -1
 }, {
 "timestamp" : 1596682800000,
```

```
"value" : 950
}]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowFunctionMetricsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowFunctionMetricsRequest request = new ShowFunctionMetricsRequest();
 try {
 ShowFunctionMetricsResponse response = client.showFunctionMetrics(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *
```

```
if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowFunctionMetricsRequest()
 response = client.show_function_metrics(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowFunctionMetricsRequest{}
 response, err := client.ShowFunctionMetrics(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```



## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

# 7.9 函数日志

## 7.9.1 查询异步日志详情

### 功能介绍

查询异步日志详情

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/async-status-log-detail

表 7-488 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

## 请求参数

表 7-489 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-490 响应 Body 参数

| 参数          | 参数类型   | 描述    |
|-------------|--------|-------|
| group_name  | String | 日志组名称 |
| group_id    | String | 日志组id |
| stream_id   | String | 日志流id |
| stream_name | String | 日志流名称 |

状态码： 401

表 7-491 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询异步日志详情。

```
GET /v2/{project_id}/fgs/functions/async-status-log-detail
```

## 响应示例

状态码： 200

Ok

```
{
 "group_id" : "xxx",
 "group_name" : "xxx",
 "stream_id" : "xxx",
 "stream_name" : "xxx"
}
```

**状态码： 401**

FORBIDDEN

```
{
 "error_code" : "FSS.0401",
 "error_msg" : "invalid token"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowProjectAsyncStatusLogInfoSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowProjectAsyncStatusLogInfoRequest request = new ShowProjectAsyncStatusLogInfoRequest();
 try {
 ShowProjectAsyncStatusLogInfoResponse response = client.showProjectAsyncStatusLogInfo(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

```
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowProjectAsyncStatusLogInfoRequest()
 response = client.show_project_async_status_log_info(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())
```

```
request := &model.ShowProjectAsyncStatusLogInfoRequest{}
response, err := client.ShowProjectAsyncStatusLogInfo(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述        |
|-----|-----------|
| 200 | Ok        |
| 401 | FORBIDDEN |

## 错误码

请参见[错误码](#)。

## 7.9.2 获取指定函数的 lts 日志组日志流配置

### 功能介绍

获取指定函数的lts日志组日志流配置。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/lts-log-detail

表 7-492 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

## 请求参数

表 7-493 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-494 响应 Body 参数

| 参数          | 参数类型   | 描述    |
|-------------|--------|-------|
| group_name  | String | 日志组名称 |
| group_id    | String | 日志组id |
| stream_id   | String | 日志流id |
| stream_name | String | 日志流名称 |

状态码： 403

表 7-495 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

获取指定函数的Its日志组日志流配置。

```
GET /v2/{project_id}/fgs/functions/{urn}/Its-log-detail
```

## 响应示例

状态码： 200

Ok

```
{
 "group_id" : "xxx",
 "stream_id" : "xxx",
 "stream_name" : "xxx"
}
```

**状态码： 403**

FORBIDDEN

```
{
 "error_code" : "FSS.0403",
 "error_msg" : "invalid token"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowLtsLogDetailsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowLtsLogDetailsRequest request = new ShowLtsLogDetailsRequest();
 try {
 ShowLtsLogDetailsResponse response = client.showLtsLogDetails(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

```
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowLtsLogDetailsRequest()
 response = client.show_lts_log_details(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())
```



```
request := &model.ShowLtsLogDetailsRequest{}
response, err := client.ShowLtsLogDetails(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述        |
|-----|-----------|
| 200 | Ok        |
| 403 | FORBIDDEN |

## 错误码

请参见[错误码](#)。

## 7.9.3 开通 lts 日志上报功能

### 功能介绍

开通lts日志上报功能。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/fgs/functions/enable-lts-logs

表 7-496 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

## 请求参数

表 7-497 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 429

表 7-498 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

开通租户Its日志上报功能。

```
POST /v2/{project_id}/fgs/functions/enable-lts-logs
```

## 响应示例

状态码： 429

调用太频繁

```
{
 "error_code": "FSS.0429",
 "error_msg": "api is busy now"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
```

```
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class EnableLtsLogsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 EnableLtsLogsRequest request = new EnableLtsLogsRequest();
 try {
 EnableLtsLogsResponse response = client.enableLtsLogs(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
```

```
request = EnableLtsLogsRequest()
response = client.enable_lts_logs(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.EnableLtsLogsRequest{}
 response, err := client.EnableLtsLogs(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述    |
|-----|-------|
| 200 | ok    |
| 429 | 调用太频繁 |

## 错误码

请参见[错误码](#)。

## 7.9.4 允许异步状态通知

### 功能介绍

允许异步状态通知。

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/fgs/functions/enable-async-status-logs

表 7-499 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

### 请求参数

表 7-500 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

### 响应参数

无

### 请求示例

允许异步状态通知。

```
POST /v2/{project_id}/fgs/functions/enable-async-status-logs
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class EnableAsyncStatusLogSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 EnableAsyncStatusLogRequest request = new EnableAsyncStatusLogRequest();
 try {
 EnableAsyncStatusLogResponse response = client.enableAsyncStatusLog(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *
```

```
if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = EnableAsyncStatusLogRequest()
 response = client.enable_async_status_log(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.EnableAsyncStatusLogRequest{}
 response, err := client.EnableAsyncStatusLog(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | ok |

## 错误码

请参见[错误码](#)。

# 7.10 函数模板

## 7.10.1 获取函数模板列表

### 功能介绍

获取函数模板列表

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/templates

表 7-501 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

表 7-502 Query 参数

| 参数     | 是否必选 | 参数类型   | 描述                                                                   |
|--------|------|--------|----------------------------------------------------------------------|
| marker | 否    | String | 本次查询起始位置，默认值0。<br>缺省值： <b>0</b><br>最小长度： <b>1</b><br>最大长度： <b>64</b> |



| 参数       | 是否必选 | 参数类型   | 描述                                  |
|----------|------|--------|-------------------------------------|
| maxitems | 否    | String | 每次查询获取的最大模板数量。                      |
| ispublic | 否    | String | 是否为公开模板。<br>若要查询函数公开模板，请将此参数设为true。 |
| runtime  | 否    | String | 指定运行时模板。                            |
| scene    | 否    | String | 指定场景模板。                             |
| service  | 否    | String | 指定云服务模板。                            |

## 请求参数

表 7-503 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）。                                                  |

## 响应参数

状态码： 200

表 7-504 响应 Body 参数

| 参数             | 参数类型                                                              | 描述         |
|----------------|-------------------------------------------------------------------|------------|
| func_templates | Array of <a href="#">ShowFunctionTemplateResponseBody</a> objects | 函数模板列表     |
| next_marker    | Integer                                                           | 函数下次记录读取位置 |

表 7-505 ShowFunctionTemplateResponseBody

| 参数 | 参数类型   | 描述   |
|----|--------|------|
| id | String | 模板id |

| 参数                    | 参数类型                                                 | 描述                                 |
|-----------------------|------------------------------------------------------|------------------------------------|
| type                  | Integer                                              | 模板类型                               |
| title                 | String                                               | 模板标题                               |
| template_name         | String                                               | 模板名称                               |
| description           | String                                               | 模板描述                               |
| runtime               | String                                               | 模板执行运行时                            |
| handler               | String                                               | 模板函数执行入口                           |
| code_type             | String                                               | 代码类型                               |
| code                  | String                                               | 代码文件                               |
| timeout               | Integer                                              | 函数执行超时时间，超时函数将被强行停止，范围3 ~ 259200秒。 |
| memory_size           | Integer                                              | 内存大小                               |
| trigger_metadata_list | Array of <a href="#">TriggerMetadataList</a> objects | 触发信息列表                             |
| temp_detail           | <a href="#">TempDetail</a> object                    |                                    |
| user_data             | String                                               | 用户数据                               |
| encrypted_user_data   | String                                               | 加密用户数据                             |
| dependencies          | Array of strings                                     | 模板所需依赖列表                           |
| scene                 | String                                               | 模板使用场景                             |
| service               | String                                               | 模板关联云服务                            |

表 7-506 TriggerMetadataList

| 参数           | 参数类型   | 描述    |
|--------------|--------|-------|
| trigger_name | String | 触发名称  |
| trigger_type | String | 触发器类型 |
| event_type   | String | 事件类型  |
| event_data   | String | 事件数据  |

表 7-507 TempDetail

| 参数      | 参数类型   | 描述   |
|---------|--------|------|
| input   | String | 模板输入 |
| output  | String | 模板输出 |
| warning | String | 警告信息 |

状态码： 400

表 7-508 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-509 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-510 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-511 响应 Body 参数

| 参数         | 参数类型   | 描述  |
|------------|--------|-----|
| error_code | String | 错误码 |

| 参数        | 参数类型   | 描述   |
|-----------|--------|------|
| error_msg | String | 错误信息 |

## 请求示例

获取500个公开函数模板列表。

```
GET /v2/{project_id}/fgs/templates?marker=0&maxitems=500&ispublic=true
```

## 响应示例

状态码： 200

OK

```
{
 "func_templates": [{
 "id": "d3aa6e4c-xxxx-xxxx-9c09-5c50c4xxxxxx",
 "type": 1,
 "title": "access-service-with-http",
 "template_name": "access-service-with-http-php",
 "description": "access service with http.",
 "runtime": "PHP7.3",
 "handler": "index.handler",
 "code_type": "inline",
 "code": "xxxxx",
 "timeout": 30,
 "memory_size": 256,
 "trigger_metadata_list": [],
 "temp_detail": {
 "input": "None",
 "output": "execution succeed: Return to access service information through http/https,",
 "warning": "1. configure the serveraddress environment variables."
 },
 "user_data": "",
 "encrypted_user_data": "",
 "dependencies": [],
 "scene": "basic_function_usage",
 "service": "FunctionGraph"
 }],
 "next_marker": 1
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;
```

```
public class ListFunctionTemplateSolution {
 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListFunctionTemplateRequest request = new ListFunctionTemplateRequest();
 request.withMarker("<marker>");
 request.withMaxItems("<maxitems>");
 request.withIsPublic("<ispublic>");
 request.withRuntime("<runtime>");
 request.withScene("<scene>");
 request.withService("<service>");
 try {
 ListFunctionTemplateResponse response = client.listFunctionTemplate(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
```

```
request = ListFunctionTemplateRequest()
request.marker = "<marker>"
request.maxitems = "<maxitems>"
request.ispublic = "<ispublic>"
request.runtime = "<runtime>"
request.scene = "<scene>"
request.service = "<service>"
response = client.list_function_template(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListFunctionTemplateRequest{}
 markerRequest := "<marker>"
 request.Marker = &markerRequest
 maxitemsRequest := "<maxitems>"
 request.Maxitems = &maxitemsRequest
 ispublicRequest := "<ispublic>"
 request.Ispublic = &ispublicRequest
 runtimeRequest := "<runtime>"
 request.Runtime = &runtimeRequest
 sceneRequest := "<scene>"
 request.Scene = &sceneRequest
 serviceRequest := "<service>"
 request.Service = &serviceRequest
 response, err := client.ListFunctionTemplate(request)
 if err == nil {
 fmt.Printf("%v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.10.2 获取指定函数模板

### 功能介绍

获取指定函数模板

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/templates/{template\_id}

表 7-512 路径参数

| 参数          | 是否必选 | 参数类型   | 描述                                       |
|-------------|------|--------|------------------------------------------|
| project_id  | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| template_id | 是    | String | 指定模板id。                                  |

## 请求参数

表 7-513 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-514 响应 Body 参数

| 参数                    | 参数类型                                                 | 描述                               |
|-----------------------|------------------------------------------------------|----------------------------------|
| id                    | String                                               | 模板id                             |
| type                  | Integer                                              | 模板类型                             |
| title                 | String                                               | 模板标题                             |
| template_name         | String                                               | 模板名称                             |
| description           | String                                               | 模板描述                             |
| runtime               | String                                               | 模板执行运行时                          |
| handler               | String                                               | 模板函数执行入口                         |
| code_type             | String                                               | 代码类型                             |
| code                  | String                                               | 代码文件                             |
| timeout               | Integer                                              | 函数执行超时时间，超时函数将被强行停止，范围3~259200秒。 |
| memory_size           | Integer                                              | 内存大小                             |
| trigger_metadata_list | Array of <a href="#">TriggerMetadataList</a> objects | 触发信息列表                           |
| temp_detail           | <a href="#">TempDetail</a> object                    |                                  |
| user_data             | String                                               | 用户数据                             |



| 参数                  | 参数类型             | 描述       |
|---------------------|------------------|----------|
| encrypted_user_data | String           | 加密用户数据   |
| dependencies        | Array of strings | 模板所需依赖列表 |
| scene               | String           | 模板使用场景   |
| service             | String           | 模板关联云服务  |

表 7-515 TriggerMetadataList

| 参数           | 参数类型   | 描述    |
|--------------|--------|-------|
| trigger_name | String | 触发名称  |
| trigger_type | String | 触发器类型 |
| event_type   | String | 事件类型  |
| event_data   | String | 事件数据  |

表 7-516 TempDetail

| 参数      | 参数类型   | 描述   |
|---------|--------|------|
| input   | String | 模板输入 |
| output  | String | 模板输出 |
| warning | String | 警告信息 |

状态码： 400

表 7-517 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-518 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-519 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-520 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-521 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

获取指定函数模板。

```
GET /v2/{project_id}/fgs/templates/{template_id}
```

## 响应示例

状态码： 200

OK

```
{
 "id": "d3aa6e4c-xxxx-xxxx-9c09-5c50c4xxxxxx",
 "type": 1,
 "title": "access-service-with-http",
 "template_name": "access-service-with-http-php",
 "description": "access service with http.",
 "runtime": "PHP7.3",
 "handler": "index.handler",
 "code_type": "inline",
 "code": "xxxxx",
 "timeout": 30,
 "memory_size": 256,
 "trigger_metadata_list": [],
 "temp_detail": {
 "input": "None",
 "output": "execution succeed: Return to access service information through http/https,",
 "warning": "1. configure the serveraddress environment variables."
 },
 "user_data": "",
 "encrypted_user_data": "",
 "dependencies": [],
 "scene": "basic_function_usage",
 "service": "FunctionGraph"
}
```

**状态码： 404**

Not Found 找不到资源。

```
{
 "error_code": "FSS.1059",
 "error_msg": "The function template does not exist."
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowFunctionTemplateSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
```

```
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
ShowFunctionTemplateRequest request = new ShowFunctionTemplateRequest();
try {
 ShowFunctionTemplateResponse response = client.showFunctionTemplate(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowFunctionTemplateRequest()
 response = client.show_function_template(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.ShowFunctionTemplateRequest{}
response, err := client.ShowFunctionTemplate(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.11 函数预留实例

## 7.11.1 获取函数预留实例配置列表

### 功能介绍

获取函数预留实例配置列表

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/reservedinstanceconfigs

表 7-522 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

表 7-523 Query 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                                   |
|--------------|------|--------|--------------------------------------------------------------------------------------|
| function_urn | 否    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。                                                    |
| marker       | 否    | String | 本次查询起始位置，默认值0。<br>缺省值： <b>0</b><br>最小长度： <b>1</b><br>最大长度： <b>64</b>                 |
| limit        | 否    | String | 本次查询最大返回的数据条数，最大值500，默认值100。<br>缺省值： <b>100</b><br>最小长度： <b>1</b><br>最大长度： <b>64</b> |

## 请求参数

表 7-524 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-525 响应 Body 参数

| 参数                 | 参数类型                                                     | 描述       |
|--------------------|----------------------------------------------------------|----------|
| reserved_instances | Array of <a href="#">ReservedInstanceConfigs</a> objects | 函数预留实例列表 |
| page_info          | <a href="#">PageInfo</a> object                          |          |
| count              | Long                                                     | 函数个数     |

表 7-526 ReservedInstanceConfigs

| 参数             | 参数类型                                 | 描述                    |
|----------------|--------------------------------------|-----------------------|
| function_urn   | String                               | 函数URN                 |
| qualifier_type | String                               | 限定类型, 支持version和alias |
| qualifier_name | String                               | 限定类型对应的取值             |
| min_count      | Integer                              | 预留实例个数                |
| idle_mode      | Boolean                              | 是否开启闲置模式配置            |
| tactics_config | <a href="#">TacticsConfig</a> object |                       |

表 7-527 TacticsConfig

| 参数             | 参数类型                                 | 描述     |
|----------------|--------------------------------------|--------|
| cron_configs   | Array of <b>CronConfig</b> objects   | 定时配置列表 |
| metric_configs | Array of <b>MetricConfig</b> objects | 流量配置列表 |

表 7-528 CronConfig

| 参数           | 参数类型    | 描述                |
|--------------|---------|-------------------|
| name         | String  | 定时配置名称            |
| cron         | String  | 定时表达式             |
| count        | Integer | 拉起预留实例个数          |
| start_time   | Long    | 开始时间戳（epoch时间戳格式） |
| expired_time | Long    | 失效时间戳（epoch时间戳格式） |

表 7-529 MetricConfig

| 参数        | 参数类型    | 描述                                                                                                        |
|-----------|---------|-----------------------------------------------------------------------------------------------------------|
| name      | String  | 流量配置名称                                                                                                    |
| type      | String  | 流量配置类型，当前只支持预留实例使用率一种类型<br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>Concurrency</b></li> </ul> |
| threshold | Integer | 流量阈值                                                                                                      |
| min       | Integer | 流量最小值                                                                                                     |

表 7-530 PageInfo

| 参数              | 参数类型 | 描述      |
|-----------------|------|---------|
| next_marker     | Long | 下一次读取位置 |
| previous_marker | Long | 上一次读取位置 |
| current_count   | Long | 当前页总数   |



**状态码： 400**

**表 7-531 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 401**

**表 7-532 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 403**

**表 7-533 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 404**

**表 7-534 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 500**

表 7-535 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

获取函数预留实例配置列表

GET https://{Endpoint}/v2/{project\_id}/fgs/functions/reservedinstanceconfigs

## 响应示例

状态码：200

OK

```
{
 "reserved_instances": [
 {
 "function_urn": "urn:fss:{region}:46b6f338fc3445b8846c71dfb1fbxxx:function:default:xxxx:latest",
 "qualifier_type": "version",
 "qualifier_name": "latest",
 "min_count": 10,
 "idle_mode": false,
 "tactics_config": {
 "cron_configs": [
 {
 "name": "cronConfig",
 "cron": "0 1 * * *",
 "count": 15,
 "start_time": 1658073600,
 "expired_time": 1658160000
 }
],
 "metric_configs": [
 {
 "name": "metricConfig",
 "type": "Concurrency",
 "threshold": 90,
 "min": 12
 }
]
 }
 }
],
 "page_info": {
 "next_marker": 1,
 "previous_marker": 0,
 "current_count": 1
 },
 "count": 1
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListReservedInstanceConfigsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListReservedInstanceConfigsRequest request = new ListReservedInstanceConfigsRequest();
 request.withFunctionUrn("<function_urn>");
 request.withMarker("<marker>");
 request.withLimit("<limit>");
 try {
 ListReservedInstanceConfigsResponse response = client.listReservedInstanceConfigs(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = ListReservedInstanceConfigsRequest()
 request.function_urn = "<function_urn>"
 request.marker = "<marker>"
 request.limit = "<limit>"
 response = client.list_reserved_instance_configs(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListReservedInstanceConfigsRequest{}
 functionUrnRequest := "<function_urn>"
 request.FunctionUrn = &functionUrnRequest
 markerRequest := "<marker>"
 request.Marker = &markerRequest
 limitRequest := "<limit>"
 request.Limit = &limitRequest
 response, err := client.ListReservedInstanceConfigs(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.11.2 修改函数预留实例数量

### 功能介绍

修改函数预留实例数量。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/fgs/functions/{function\_urn}/reservedinstances

表 7-536 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

## 请求参数

表 7-537 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

表 7-538 请求 Body 参数

| 参数             | 是否必选 | 参数类型                        | 描述         |
|----------------|------|-----------------------------|------------|
| count          | 是    | Integer                     | 预留实例个数     |
| idle_mode      | 否    | Boolean                     | 是否开启闲置模式配置 |
| tactics_config | 否    | <b>TacticsConfig</b> object |            |

表 7-539 TacticsConfig

| 参数             | 是否必选 | 参数类型                                 | 描述     |
|----------------|------|--------------------------------------|--------|
| cron_configs   | 否    | Array of <b>CronConfig</b> objects   | 定时配置列表 |
| metric_configs | 否    | Array of <b>MetricConfig</b> objects | 流量配置列表 |

表 7-540 CronConfig

| 参数         | 是否必选 | 参数类型    | 描述                |
|------------|------|---------|-------------------|
| name       | 否    | String  | 定时配置名称            |
| cron       | 否    | String  | 定时表达式             |
| count      | 否    | Integer | 拉起预留实例个数          |
| start_time | 否    | Long    | 开始时间戳（epoch时间戳格式） |

| 参数           | 是否必选 | 参数类型 | 描述                |
|--------------|------|------|-------------------|
| expired_time | 否    | Long | 失效时间戳（epoch时间戳格式） |

表 7-541 MetricConfig

| 参数        | 是否必选 | 参数类型    | 描述                                                                                                        |
|-----------|------|---------|-----------------------------------------------------------------------------------------------------------|
| name      | 否    | String  | 流量配置名称                                                                                                    |
| type      | 否    | String  | 流量配置类型，当前只支持预留实例使用率一种类型<br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>Concurrency</b></li> </ul> |
| threshold | 否    | Integer | 流量阈值                                                                                                      |
| min       | 否    | Integer | 流量最小值                                                                                                     |

## 响应参数

状态码： 200

表 7-542 响应 Body 参数

| 参数             | 参数类型                           | 描述         |
|----------------|--------------------------------|------------|
| count          | Integer                        | 预留实例个数     |
| idle_mode      | Boolean                        | 是否开启闲置模式配置 |
| tactics_config | <b>TacticsConfig</b><br>object |            |

表 7-543 TacticsConfig

| 参数             | 参数类型                                    | 描述     |
|----------------|-----------------------------------------|--------|
| cron_configs   | Array of <b>CronConfig</b><br>objects   | 定时配置列表 |
| metric_configs | Array of <b>MetricConfig</b><br>objects | 流量配置列表 |

表 7-544 CronConfig

| 参数           | 参数类型    | 描述                |
|--------------|---------|-------------------|
| name         | String  | 定时配置名称            |
| cron         | String  | 定时表达式             |
| count        | Integer | 拉起预留实例个数          |
| start_time   | Long    | 开始时间戳（epoch时间戳格式） |
| expired_time | Long    | 失效时间戳（epoch时间戳格式） |

表 7-545 MetricConfig

| 参数        | 参数类型    | 描述                                                                                                   |
|-----------|---------|------------------------------------------------------------------------------------------------------|
| name      | String  | 流量配置名称                                                                                               |
| type      | String  | 流量配置类型，当前只支持预留实例使用率一种类型<br>枚举值： <ul style="list-style-type: none"><li>• <b>Concurrency</b></li></ul> |
| threshold | Integer | 流量阈值                                                                                                 |
| min       | Integer | 流量最小值                                                                                                |

状态码：400

表 7-546 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码：401

表 7-547 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |



**状态码： 403****表 7-548 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 404****表 7-549 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 500****表 7-550 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

将函数预留实例数量更新为3。

```
PUT https://{Endpoint}/v2/{project_id}/fgs/functions/{function_urn}/reservedinstances
{
 "count" : 3
}
```

## 响应示例

**状态码： 200**

OK

```
{
 "count" : 2
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

将函数预留实例数量更新为3。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateFunctionReservedInstancesCountSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 UpdateFunctionReservedInstancesCountRequest request = new
 UpdateFunctionReservedInstancesCountRequest();
 UpdateFunctionReservedInstancesCountRequestBody body = new
 UpdateFunctionReservedInstancesCountRequestBody();
 body.withCount(3);
 request.withBody(body);
 try {
 UpdateFunctionReservedInstancesCountResponse response =
 client.updateFunctionReservedInstancesCount(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

将函数预留实例数量更新为3。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = UpdateFunctionReservedInstancesCountRequest()
 request.body = UpdateFunctionReservedInstancesCountRequestBody(
 count=3
)
 response = client.update_function_reserved_instances_count(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

将函数预留实例数量更新为3。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())
```

```
request := &model.UpdateFunctionReservedInstancesCountRequest{}
request.Body = &model.UpdateFunctionReservedInstancesCountRequestBody{
 Count: int32(3),
}
response, err := client.UpdateFunctionReservedInstancesCount(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.11.3 获取函数预留实例数量

### 功能介绍

获取函数预留实例数量。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/reservedinstances

表 7-551 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

表 7-552 Query 参数

| 参数     | 是否必选 | 参数类型   | 描述                                                                                               |
|--------|------|--------|--------------------------------------------------------------------------------------------------|
| marker | 否    | String | 上一次查询到的最后的记录位置。<br>缺省值： <b>0</b>                                                                 |
| limit  | 否    | String | 每次查询获取的最大函数记录数量<br>最大值：400 如果不提供该值或者提供的值大于400或等于0，则使用默认值：400 如果该值小于0，则返回参数错误。<br>缺省值： <b>400</b> |
| urn    | 否    | String | 查询指定函数版本预留实例数的函数urn。                                                                             |

## 请求参数

表 7-553 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-554 响应 Body 参数

| 参数                | 参数类型                                                  | 描述       |
|-------------------|-------------------------------------------------------|----------|
| reservedinstances | Array of <a href="#">FuncReservedInstance</a> objects | 函数预留实例列表 |
| page_info         | <a href="#">PageInfo</a> object                       |          |
| count             | Long                                                  | 函数个数     |

表 7-555 FuncReservedInstance

| 参数       | 参数类型   | 描述     |
|----------|--------|--------|
| func_urn | String | 函数urn  |
| count    | Long   | 预留实例数目 |

表 7-556 PageInfo

| 参数              | 参数类型 | 描述      |
|-----------------|------|---------|
| next_marker     | Long | 下一次读取位置 |
| previous_marker | Long | 上一次读取位置 |
| current_count   | Long | 当前页总数   |

状态码： 400

表 7-557 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-558 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-559 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-560 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-561 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询租户预留实例列表。

GET https://{Endpoint}/v2/{project\_id}/fgs/functions/reservedinstances

## 响应示例

状态码： 200

OK

```
{
 "reservedinstances" : [{
 "func_urn" : "urn:fss:xxxxx:46b6f338fc3445b8846c71dfb1fbxxxx:function:csharp:test2-0:latest",
 "count" : 2
 }],
 "page_info" : {
 "next_marker" : 2,
 "previous_marker" : 0,
 "current_count" : 2
 }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListFunctionReservedInstancesSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListFunctionReservedInstancesRequest request = new ListFunctionReservedInstancesRequest();
 request.withMarker("<marker>");
 request.withLimit("<limit>");
 request.withUrn("<urn>");
 try {
 ListFunctionReservedInstancesResponse response = client.listFunctionReservedInstances(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```



```
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListFunctionReservedInstancesRequest()
 request.marker = "<marker>"
 request.limit = "<limit>"
 request.urn = "<urn>"
 response = client.list_function_reserved_instances(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
```

```

 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build()

 request := &model.ListFunctionReservedInstancesRequest{}
 markerRequest:= "<marker>"
 request.Marker = &markerRequest
 limitRequest:= "<limit>"
 request.Limit = &limitRequest
 urnRequest:= "<urn>"
 request.Urn = &urnRequest
 response, err := client.ListFunctionReservedInstances(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.12 函数导入导出

### 7.12.1 导出函数

#### 功能介绍

导出函数

#### 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/export

表 7-562 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

表 7-563 Query 参数

| 参数     | 是否必选 | 参数类型    | 描述                                                                            |
|--------|------|---------|-------------------------------------------------------------------------------|
| config | 否    | Boolean | 是否导出函数配置，默认为false。若无type参数，则必填code=true或config=true至少一项。<br>缺省值： <b>false</b> |
| code   | 否    | Boolean | 是否导出函数代码，默认为false。若无type参数，则必填code=true或config=true至少一项。<br>缺省值： <b>false</b> |
| type   | 否    | String  | 不兼容与code、config参数混用；type=code代表导出代码，type=config代表导出配置                         |

## 请求参数

表 7-564 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 404

表 7-565 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-566 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

- 导出函数代码(新方式)。  
GET /v2/{project\_id}/fgs/functions/{func\_urn}/export?type=code
- 导出函数代码与配置(老方式)。  
GET /v2/{project\_id}/fgs/functions/{func\_urn}/export?code=true&config=true

## 响应示例

状态码： 404

Not Found 找不到资源。

```
{
 "error_code" : "FSS.1051",
 "error_msg" : "Not found the function"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
```

```
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ExportFunctionSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ExportFunctionRequest request = new ExportFunctionRequest();
 request.withConfig("<config>");
 request.withCode("<code>");
 request.withType("<type>");
 try {
 ExportFunctionResponse response = client.exportFunction(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
```

```
request = ExportFunctionRequest()
request.config = <Config>
request.code = <Code>
request.type = "<type>"
response = client.export_function(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ExportFunctionRequest{}
 configRequest := <config>
 request.Config = &configRequest
 codeRequest := <code>
 request.Code = &codeRequest
 typeRequest := "<type>"
 request.Type = &typeRequest
 response, err := client.ExportFunction(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.12.2 导入函数

### 功能介绍

导入函数

### 调用方法

请参见[如何调用API](#)。

## URI

POST /v2/{project\_id}/fgs/functions/import

表 7-567 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

## 请求参数

表 7-568 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

表 7-569 请求 Body 参数

| 参数        | 是否必选 | 参数类型   | 描述                     |
|-----------|------|--------|------------------------|
| func_name | 是    | String | 函数名                    |
| file_name | 是    | String | 文件名                    |
| file_type | 是    | String | 文件类型                   |
| file_code | 是    | String | 函数代码。代码必须要进行 base64 编码 |
| package   | 否    | String | 应用名称，默认为 default       |

## 响应参数

状态码：200

表 7-570 响应 Body 参数

| 参数           | 参数类型   | 描述                                        |
|--------------|--------|-------------------------------------------|
| func_urn     | String | 函数的 URN ( Uniform Resource Name )，唯一标识函数。 |
| func_name    | String | 函数名称。                                     |
| domain_id    | String | 域名 id。                                    |
| namespace    | String | 租户的 project id。                           |
| project_name | String | 租户的 project name。                         |
| package      | String | 函数所属的分组 Package，用于用户针对函数的自定义分组。           |



| 参数      | 参数类型    | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|---------|---------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| runtime | String  | <p>FunctionGraph函数的执行环境 Python2.7: Python语言2.7版本。 Python3.6: Python语言3.6版本。 Python3.9: Python语言3.9版本。 Python3.10: Python语言3.10版本。 Go1.x: Go语言1.x版本。 Java8: Java语言8版本。 Java11: Java语言11版本。 Node.js6.10: Nodejs语言6.10版本。 Node.js8.10: Nodejs语言8.10版本。 Node.js10.16: Nodejs语言10.16版本。 Node.js12.13: Nodejs语言12.13版本。 Node.js14.18: Nodejs语言14.18版本。 Node.js16.17: Nodejs语言16.17版本。 Node.js18.15: Nodejs语言18.15版本。 C#(.NET Core 2.1): C#语言2.1版本。 C#(.NET Core 3.1): C#语言3.1版本。 Custom: 自定义运行时。 PHP7.3: Php语言7.3版本。 Cangjie1.0: 仓颉语言1.0版本。 http: HTTP函数。 Custom Image: 自定义镜像函数。</p> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>Java8</b></li> <li>● <b>Java11</b></li> <li>● <b>Node.js6.10</b></li> <li>● <b>Node.js8.10</b></li> <li>● <b>Node.js10.16</b></li> <li>● <b>Node.js12.13</b></li> <li>● <b>Node.js14.18</b></li> <li>● <b>Node.js16.17</b></li> <li>● <b>Node.js18.15</b></li> <li>● <b>Python2.7</b></li> <li>● <b>Python3.6</b></li> <li>● <b>Python3.10</b></li> <li>● <b>Go1.x</b></li> <li>● <b>C#(.NET Core 2.1)</b></li> <li>● <b>C#(.NET Core 3.1)</b></li> <li>● <b>Custom</b></li> <li>● <b>PHP7.3</b></li> <li>● <b>Python3.9</b></li> <li>● <b>http</b></li> <li>● <b>Custom Image</b></li> <li>● <b>Cangjie1.0</b></li> </ul> |
| timeout | Integer | <p>函数执行超时时间，超时函数将被强行停止，范围3~259200秒。</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

| 参数            | 参数类型    | 描述                                                                                                                                                                                                                                                                         |
|---------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| handler       | String  | 函数执行入口 规则：xx.xx，必须包含“.” 举例：对于node.js函数：myfunction.handler，则表示函数的文件名为myfunction.js，执行的入口函数名为handler。长度不超过128个字符。<br>最小长度：3<br>最大长度：128                                                                                                                                      |
| memory_size   | Integer | 函数消耗的内存。单位M。取值范围为：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。最小值为128，最大值为4096。                                                                                                                                                                           |
| gpu_memory    | Integer | 函数消耗的显存，只支持自定义运行时与自定义镜像函数配置GPU。单位MB。取值范围为：1024、2048、3072、4096、5120、6144、7168、8192、9216、10240、11264、12288、13312、14336、15360、16384。最小值为1024，最大值为16384。                                                                                                                       |
| cpu           | Integer | 函数占用的cpu资源。单位为millicore（1 core=1000 millicores）。取值与MemorySize成比例，默认是128M内存占0.1个核（100 millicores）。                                                                                                                                                                          |
| code_type     | String  | 函数代码类型，取值有5种。inline: UI在线编辑代码。zip: 函数代码为zip包。obs: 函数代码来源于obs存储。jar: 函数代码为jar包，主要针对Java函数。Custom-Image-Swr: 函数代码来源于SWR自定义镜像。<br>枚举值：<br><ul style="list-style-type: none"> <li>• inline</li> <li>• zip</li> <li>• obs</li> <li>• jar</li> <li>• Custom-Image-Swr</li> </ul> |
| code_url      | String  | 当code_type为obs时，该值为函数代码包在OBS上的地址，code_type为其他值时，该字段为空。                                                                                                                                                                                                                     |
| code_filename | String  | 函数的文件名，当code_type为jar/zip时必须提供该字段，inline和obs不需要提供。                                                                                                                                                                                                                         |
| code_size     | Long    | 函数大小，单位：字节。                                                                                                                                                                                                                                                                |
| user_data     | String  | 用户自定义的name/value信息。在函数中使用的参数。举例：如函数要访问某个主机，可以设置自定义参数：Host={host_ip}，最多定义20个，总长度不超过4KB。                                                                                                                                                                                     |

| 参数                  | 参数类型                         | 描述                                                                                                                            |
|---------------------|------------------------------|-------------------------------------------------------------------------------------------------------------------------------|
| digest              | String                       | 函数代码SHA512 hash值，用于判断函数是否变化。                                                                                                  |
| version             | String                       | 函数版本号，由系统自动生成，规则：vYYYYMMDD-HHMMSS（v+年月日-时分秒）。                                                                                 |
| image_name          | String                       | 函数版本的内部标识。                                                                                                                    |
| xrole               | String                       | 函数使用的权限委托名称，需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。                                                                         |
| app_xrole           | String                       | 函数app使用的权限委托名称，需要IAM支持，并在IAM界面创建委托，当函数需要访问其他服务时，必须提供该字段。                                                                      |
| description         | String                       | 函数描述。                                                                                                                         |
| version_description | String                       | 函数版本描述。                                                                                                                       |
| last_modified       | String                       | 函数最后一次更新时间。                                                                                                                   |
| func_vpc            | <b>FuncVpc</b> object        | 函数vpc配置。需同时为函数配置具有vpc权限的委托。                                                                                                   |
| depend_list         | Array of strings             | 依赖id列表                                                                                                                        |
| depend_version_list | Array of strings             | 依赖版本id列表                                                                                                                      |
| strategy_config     | <b>StrategyConfig</b> object | 函数策略配置。                                                                                                                       |
| extend_config       | String                       | 函数扩展配置。                                                                                                                       |
| initializer_handler | String                       | 函数初始化入口，规则：xx.xx，必须包含“.”。当配置初始化函数时，此参数必填。举例：对于node.js函数：myfunction.initializer，则表示函数的文件名为myfunction.js，初始化的入口函数名为initializer。 |
| initializer_timeout | Integer                      | 初始化超时时间，超时函数将被强行停止，范围1~300秒。当配置初始化函数时，此参数必填。                                                                                  |
| pre_stop_handler    | String                       | 函数预停止函数的入口，规则：xx.xx，必须包含“.”。举例：对于node.js函数：myfunction.pre_stop_handler，则表示函数的文件名为myfunction.js，初始化的入口函数名为pre_stop_handler。    |
| pre_stop_timeout    | Integer                      | 初始化超时时间，超时函数将被强行停止，范围1~90秒。                                                                                                   |

| 参数                    | 参数类型   | 描述                   |
|-----------------------|--------|----------------------|
| enterprise_project_id | String | 企业项目ID，在企业用户创建函数时必填。 |

表 7-571 FuncVpc

| 参数              | 参数类型             | 描述             |
|-----------------|------------------|----------------|
| domain_id       | String           | 域名id。          |
| namespace       | String           | 租户的project id。 |
| vpc_name        | String           | 虚拟私有云名称。       |
| vpc_id          | String           | 虚拟私有云唯一标识。     |
| subnet_name     | String           | 子网名称。          |
| subnet_id       | String           | 子网编号。          |
| cidr            | String           | 子网掩码。          |
| gateway         | String           | 网关。            |
| security_groups | Array of strings | 安全组            |

表 7-572 StrategyConfig

| 参数             | 参数类型    | 描述                                                    |
|----------------|---------|-------------------------------------------------------|
| concurrency    | Integer | 单函数最大实例数，v1取值0和-1，v2取值-1到1000 -1代表该函数实例数无限制 0代表该函数被禁用 |
| concurrent_num | Integer | 单实例并发数，v2版本才支持，取值1到1000                               |

状态码： 409

表 7-573 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 500**

**表 7-574 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

导入函数，文件名为"test.zip"，类型为zip，文件base64编码为"xxx"。

```
POST https://{Endpoint}/v2/{project_id}/fgs/functions/import
{
 "func_name": "test",
 "file_name": "test.zip",
 "file_type": "zip",
 "file_code": "xxxxxxxxxxxxxxxxxxxxxxx"
}
```

## 响应示例

**状态码： 200**

ok

```
{
 "func_urn": "urn:fss:{region}:46b6f338fc3445b8846c71dfb1fbd9e8:function:default:test_v1_2",
 "func_name": "test_v1_2",
 "domain_id": "14ee2e35****a7998b*****aa24cabf",
 "namespace": "46b6f338fc3445b8846c71dfb1fbd9e8",
 "project_name": "{region}",
 "package": "default",
 "runtime": "Node.js6.10",
 "timeout": 3,
 "handler": "index.handler",
 "memory_size": 128,
 "cpu": 300,
 "code_type": "zip",
 "code_filename": "index.zip",
 "code_size": 6709,
 "digest":
"faa825575c45437cddd4e369bea69893bcbe195d478178462ad90984fe72993f3f59d15f41c5373f807f3e05fb9af322c55dabeb16565c386e402413458e6068",
 "version": "latest",
 "image_name": "latest-191025153727@zehht",
 "last_modified": "2019-10-25 15:37:27",
 "strategy_config": {
 "concurrency": -1
 },
 "enterprise_project_id": "46b6f338fc3445b8846c71dfb1fbxxxx"
}
```

**状态码： 409**

Conflict

```
{
 "error_code": "FSS.1061",
```

```
"error_msg" : "The function has existed"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

导入函数，文件名为"test.zip"，类型为zip，文件base64编码为"xxx"。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ImportFunctionSolution {
 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ImportFunctionRequest request = new ImportFunctionRequest();
 ImportFunctionRequestBody body = new ImportFunctionRequestBody();
 body.withFileCode("xxxxxxxxxxxxxxxxxxxxxx");
 body.withFileType("zip");
 body.withFileName("test.zip");
 body.withFuncName("test");
 request.withBody(body);
 try {
 ImportFunctionResponse response = client.importFunction(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

导入函数，文件名为"test.zip"，类型为zip，文件base64编码为"xxx"。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ImportFunctionRequest()
 request.body = ImportFunctionRequestBody(
 file_code="xxxxxxxxxxxxxxxxxxxx",
 file_type="zip",
 file_name="test.zip",
 func_name="test"
)
 response = client.import_function(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

导入函数，文件名为"test.zip"，类型为zip，文件base64编码为"xxx"。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
```

```
WithSk(sk).
Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.ImportFunctionRequest{}
request.Body = &model.ImportFunctionRequestBody{
 FileCode: "xxxxxxxxxxxxxxxxxxxxxxxx",
 FileType: "zip",
 FileName: "test.zip",
 FuncName: "test",
}
response, err := client.ImportFunction(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | ok                            |
| 409 | Conflict                      |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.13 函数触发器

### 7.13.1 删除指定函数的所有触发器

#### 功能介绍

删除指定函数所有触发器设置。

在提供函数版本且非latest的情况下，删除对应函数版本的触发器。在提供函数别名的情况下，删除对应函数别名的触发器。在不提供函数版本（也不提供别名）或版本为latest的情况下，删除该函数所有的触发器（包括所有版本和别名）。



## 调用方法

请参见[如何调用API](#)。

## URI

DELETE /v2/{project\_id}/fgs/triggers/{function\_urn}

表 7-575 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

## 请求参数

表 7-576 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）<br>缺省值： <b>application/json</b>                   |

## 响应参数

状态码： 400

表 7-577 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-578 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-579 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-580 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-581 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

删除指定函数所有触发器。

```
DELETE https://{Endpoint}/v2/{project_id}/fgs/triggers/{function_urn}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class BatchDeleteFunctionTriggersSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR_REGION>"))
 .build();
 BatchDeleteFunctionTriggersRequest request = new BatchDeleteFunctionTriggersRequest();
 try {
 BatchDeleteFunctionTriggersResponse response = client.batchDeleteFunctionTriggers(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
```

```
In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = BatchDeleteFunctionTriggersRequest()
 response = client.batch_delete_function_triggers(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.BatchDeleteFunctionTriggersRequest{}
 response, err := client.BatchDeleteFunctionTriggers(request)
 if err == nil {
 fmt.Printf("%v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 204 | No Content                    |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.13.2 获取指定函数的所有触发器

### 功能介绍

获取指定函数的所有触发器设置。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/triggers/{function\_urn}

表 7-582 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

## 请求参数

表 7-583 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）<br>缺省值： <b>application/json</b>                   |

## 响应参数

状态码： 200

表 7-584 响应 Body 参数

| 参数     | 参数类型                                              | 描述 |
|--------|---------------------------------------------------|----|
| [数组元素] | Array of <b>ListFunctionTriggerResult</b> objects |    |

表 7-585 ListFunctionTriggerResult

| 参数         | 参数类型   | 描述     |
|------------|--------|--------|
| trigger_id | String | 触发器ID。 |

| 参数                | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| trigger_type_code | String | <p>触发器类型。（当前OBS触发器暂不支持中国站。）</p> <ul style="list-style-type: none"> <li>● TIMER: "定时触发器。"</li> <li>● APIG: "APIG触发器。"</li> <li>● CTS: "云审计服务触发器。"</li> <li>● DDS: "文档数据库服务触发器。"</li> <li>● DMS: "分布式服务触发器。"</li> <li>● DIS: "数据接入服务触发器。"</li> <li>● LTS: "云日志服务触发器。"</li> <li>● OBS: "对象存储触发器。"</li> <li>● SMN: "消息通知服务触发器。"</li> <li>● KAFKA: "专享版消息通知服务触发器。"</li> </ul> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>TIMER</b></li> <li>● <b>APIG</b></li> <li>● <b>CTS</b></li> <li>● <b>DDS</b></li> <li>● <b>DMS</b></li> <li>● <b>DIS</b></li> <li>● <b>LTS</b></li> <li>● <b>OBS</b></li> <li>● <b>SMN</b></li> <li>● <b>KAFKA</b></li> <li>● <b>RABBITMQ</b></li> <li>● <b>DEDICATEDGATEWAY</b></li> <li>● <b>OPENSOURCEKAFKA</b></li> <li>● <b>APIC</b></li> <li>● <b>GeminiDB Mongo</b></li> <li>● <b>EVENTGRID</b></li> <li>● <b>IOTDA</b></li> </ul> |
| trigger_status    | String | <p>"触发器状态"</p> <ul style="list-style-type: none"> <li>● ACTIVE: 启用状态。</li> <li>● DISABLED: 禁用状态。</li> </ul> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>ACTIVE</b></li> <li>● <b>DISABLED</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

| 参数                | 参数类型                                       | 描述       |
|-------------------|--------------------------------------------|----------|
| event_data        | <b>TriggerEventDataResponseBody</b> object | 触发器源事件。  |
| last_updated_time | String                                     | 最后更新时间。  |
| created_time      | String                                     | 触发器创建时间。 |

表 7-586 TriggerEventDataResponseBody

| 参数            | 参数类型    | 描述                                                                                                                                                                                                                                                                  |
|---------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name          | String  | 触发器名称                                                                                                                                                                                                                                                               |
| schedule_type | String  | 定时触发类型（TIMER触发器参数）。 <ul style="list-style-type: none"> <li>Rate: 指定固定频率（分钟、小时、天数）定期调用函数，单位为分钟时，输入值不能超过60；单位为小时时，输入值不能超过24；单位为天时，输入值不能超过30。</li> <li>Cron: 指定Cron表达式定期调用函数<br/>枚举值： <ul style="list-style-type: none"> <li>Rate</li> <li>Cron</li> </ul> </li> </ul> |
| schedule      | String  | 定时触发规则（TIMER触发器参数）。 <ul style="list-style-type: none"> <li>触发类型为Rate时对应定时规则</li> <li>触发类型为Cron时对应Cron表达式</li> </ul>                                                                                                                                                 |
| user_event    | String  | 附加信息（TIMER触发器参数）。当Timer触发器触发函数执行时，执行事件（函数的event参数）为：{"version": "v1.0", "time": "2018-06-01T08:30:00+08:00", "trigger_type": "TIMER", "trigger_name": "Timer_001", "user_event": "您输入的附加信息"}                                                                        |
| triggerid     | String  | APIG触发器id。（APIG触发器参数）                                                                                                                                                                                                                                               |
| type          | Integer | API接口类型（APIG触发器参数）。 <ul style="list-style-type: none"> <li>1: 公有API</li> <li>2: 私有API</li> </ul>                                                                                                                                                                    |
| path          | String  | APIG接口PATH路径（APIG触发器参数）。                                                                                                                                                                                                                                            |



| 参数         | 参数类型   | 描述                                                                                                                                                                                                          |
|------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| protocol   | String | API的请求协议（APIG触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>• HTTP</li> <li>• HTTPS</li> </ul>                                                                                                   |
| req_method | String | API的请求方式（APIG触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> <li>• HEAD</li> <li>• PATCH</li> <li>• OPTIONS</li> <li>• ANY</li> </ul> |
| group_id   | String | API所属的分组编号（APIG触发器参数）。                                                                                                                                                                                      |
| group_name | String | API所属的分组名称（APIG触发器参数）。                                                                                                                                                                                      |
| match_mode | String | API的匹配方式（APIG触发器参数）。<br><ul style="list-style-type: none"> <li>• SWA：前缀匹配</li> <li>• NORMAL：正常匹配（绝对匹配）</li> </ul> 枚举值：<br><ul style="list-style-type: none"> <li>• SWA</li> <li>• NORMAL</li> </ul>         |
| env_name   | String | API的发布环境（APIG触发器参数）。                                                                                                                                                                                        |
| env_id     | String | API的发布环境id（APIG触发器参数）。                                                                                                                                                                                      |
| api_id     | String | API编号（APIG触发器参数）。                                                                                                                                                                                           |
| api_name   | String | API名称（APIG触发器参数）。                                                                                                                                                                                           |

| 参数              | 参数类型                              | 描述                                                                                                                                                                                                                                                                                        |
|-----------------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| auth            | String                            | API的认证方式（APIG触发器参数）。 <ul style="list-style-type: none"> <li>• IAM：IAM认证，只允许IAM用户能访问，安全级别中等</li> <li>• APP：采用Appkey&amp;Appsecret认证，安全级别高，推荐使用</li> <li>• NONE：无认证模式，所有用户均可访问，不推荐使用</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• IAM</li> <li>• APP</li> <li>• NONE</li> </ul> |
| invoke_url      | String                            | API调用地址（APIG触发器参数）。                                                                                                                                                                                                                                                                       |
| func_info       | <b>ApigTriggerFuncInfo</b> object | APIG触发器函数工作流后端详情（APIG触发器参数）。APIG触发器此参数必填。                                                                                                                                                                                                                                                 |
| sl_domain       | String                            | APIG系统默认分配的子域名（APIG触发器参数）。<br>最小长度：1<br>最大长度：255                                                                                                                                                                                                                                          |
| backend_type    | String                            | API的后端类型（APIG触发器参数）。<br>枚举值： <ul style="list-style-type: none"> <li>• FUNCTION</li> </ul>                                                                                                                                                                                                 |
| instance_id     | String                            | 实例id。DDS、KAFKA、RABBITMQ触发器此参数必填。 <ul style="list-style-type: none"> <li>• APIG触发器：apig实例id</li> <li>• DDS触发器：文档数据库实例id</li> <li>• KAFKA触发器：KAFKA实例id</li> <li>• RABBITMQ触发器：RABBITMQ实例id</li> </ul>                                                                                       |
| roma_app_id     | String                            | API归属的集成应用编号。（APIG触发器参数）                                                                                                                                                                                                                                                                  |
| operations      | Array of strings                  | 自定义操作（CTS触发器参数）。CTS云审计服务类型和操作订阅所需要的事件通知，当CTS云审计服务获取已订阅的操作记录后，通过CTS触发器将采集到的操作记录作为参数传递来调用FunctionGraph函数。                                                                                                                                                                                   |
| collection_name | String                            | 集合名称（DDS触发器参数）。                                                                                                                                                                                                                                                                           |
| db_name         | String                            | 文档数据库名称（DDS触发器参数）。                                                                                                                                                                                                                                                                        |
| db_password     | String                            | 文档数据库密码（DDS触发器参数）。                                                                                                                                                                                                                                                                        |

| 参数                 | 参数类型             | 描述                                                                                                                                                                                                                                                             |
|--------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| db_user            | String           | 文档数据库用户名（DDS触发器参数）。                                                                                                                                                                                                                                            |
| instance_addresses | Array of strings | 文档数据库实例地址（DDS触发器参数）。                                                                                                                                                                                                                                           |
| mode               | String           | 文档数据库实例类型（DDS触发器参数）。 <ul style="list-style-type: none"> <li>• Sharding：集群实例</li> <li>• ReplicaSet：副本集实例</li> <li>• Single：单节点实例</li> </ul>                                                                                                                     |
| batch_size         | Integer          | 批处理大小，单次函数执行处理的最大数据量。DIS、DDS、KAFKA、RABBITMQ触发器此参数必填。 <ul style="list-style-type: none"> <li>• DDS触发器：批处理大小设置1-10,000的范围内</li> <li>• DIS触发器：批处理大小设置1-10,000的范围内</li> <li>• KAFKA触发器：批处理大小设置1-1,000的范围内</li> <li>• RABBITMQ触发器：批处理大小设置1-1,000的范围内</li> </ul>     |
| queue_id           | String           | 队列id（DMS触发器参数）。                                                                                                                                                                                                                                                |
| consumer_group_id  | String           | 消费组id（DMS触发器参数）。                                                                                                                                                                                                                                               |
| polling_interval   | Integer          | 拉取周期。                                                                                                                                                                                                                                                          |
| stream_name        | String           | 通道名称（DIS触发器参数）。                                                                                                                                                                                                                                                |
| sharditerator_type | String           | 起始位置（DIS触发器参数）。 <ul style="list-style-type: none"> <li>• TRIM_HORIZON：从最早被存储至分区的有效记录开始读取。</li> <li>• LATEST：从分区中的最新记录开始读取，此设置可以保证总是读到分区中最新记录。</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• <b>TRIM_HORIZON</b></li> <li>• <b>LATEST</b></li> </ul> |
| polling_unit       | String           | 拉取周期单位（DIS触发器参数）。 <ul style="list-style-type: none"> <li>• s：秒</li> <li>• ms：毫秒</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• <b>s</b></li> <li>• <b>ms</b></li> </ul>                                                                              |

| 参数              | 参数类型             | 描述                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| max_fetch_bytes | Integer          | 最大提取字节数（DIS触发器参数）。<br>最小值： <b>0</b><br>最大值： <b>4194304</b>                                                                                                                                                                                                                                                                                                      |
| is_serial       | String           | 串行处理数据（DIS触发器参数），如果开启该选项，取一次数据处理完之后才会取下一次数据；否则只要拉取周期到了就会取数据进行处理。<br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>true</b></li> <li>• <b>false</b></li> </ul>                                                                                                                                                                                             |
| log_group_id    | String           | 日志组id（LTS触发器参数）。                                                                                                                                                                                                                                                                                                                                                |
| log_topic_id    | String           | 日志流id（LTS触发器参数）。                                                                                                                                                                                                                                                                                                                                                |
| bucket          | String           | 桶名称（OBS触发器参数），用作事件源的OBS存储桶，不能和本用户已有桶重名；不能和其他用户已有的桶重名；创建成功后不支持修改。<br>最小长度： <b>1</b><br>最大长度： <b>64</b>                                                                                                                                                                                                                                                           |
| prefix          | String           | 前缀（OBS触发器参数），输入一个可选性前缀来限制对以此关键字开头的对象的通知。<br>最小长度： <b>0</b><br>最大长度： <b>1024</b>                                                                                                                                                                                                                                                                                 |
| suffix          | String           | 后缀（OBS触发器参数），输入一个可选性后缀来限制对以此关键字结尾的对象的通知<br>最小长度： <b>0</b><br>最大长度： <b>1024</b>                                                                                                                                                                                                                                                                                  |
| events          | Array of strings | 触发事件（OBS触发器参数）。 <ul style="list-style-type: none"> <li>• ObjectCreated：表示所有创建对象的操作，包含Put、Post、Copy对象以及合并段</li> <li>• Put：使用Put方法上传对象</li> <li>• Post：使用Post方法上传对象</li> <li>• Copy：使用copy方法复制对象</li> <li>• CompleteMultipartUpload：表示合并分段任务</li> <li>• ObjectRemoved：表示删除对象</li> <li>• Delete：指定对象版本号删除对象</li> <li>• DeleteMarkerCreated：不指定对象版本号删除对象</li> </ul> |

| 参数                    | 参数类型             | 描述                                                                                                                            |
|-----------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------|
| topic_urn             | String           | 主题URN（SMN触发器参数）。<br>最小长度： <b>1</b><br>最大长度： <b>255</b>                                                                        |
| topic_ids             | Array of strings | KAFKA主题id列表（KAFKA触发器参数）。                                                                                                      |
| kafka_user            | String           | KAFKA账户名（KAFKA触发器参数）。                                                                                                         |
| kafka_password        | String           | KAFKA账户密码（KAFKA触发器参数）。                                                                                                        |
| kafka_connect_address | String           | KAFKA实例连接IP地址（KAFKA触发器参数）。                                                                                                    |
| kafka_ssl_enable      | Boolean          | KAFKA连接是否开启安全认证（KAFKA触发器参数）。                                                                                                  |
| access_password       | String           | RABBITMQ账户密码（RABBITMQ触发器参数）。                                                                                                  |
| access_user           | String           | RABBITMQ账户名（RABBITMQ触发器参数）。                                                                                                   |
| connect_address       | String           | 实例连接IP地址（RABBITMQ触发器参数）。                                                                                                      |
| exchange_name         | String           | 交换机名称（RABBITMQ触发器参数）。                                                                                                         |
| vhost                 | String           | 虚拟机名称（RABBITMQ触发器参数）。                                                                                                         |
| ssl_enable            | Boolean          | RABBITMQ连接是否开启安全认证（RABBITMQ触发器参数）。                                                                                            |
| Key_encode            | Boolean          | EG obs触发器是否对对象加密（EVENTGRID触发器参数）。                                                                                             |
| agency                | String           | 使用的代理（EVENTGRID触发器参数）。                                                                                                        |
| channel_name          | String           | 通道名称（EVENTGRID触发器参数）。                                                                                                         |
| channel_id            | String           | 通道id（EVENTGRID触发器参数）。                                                                                                         |
| source_name           | String           | 事件源名称（EVENTGRID触发器参数）。                                                                                                        |
| created_time          | String           | 创建时间（EVENTGRID触发器参数）。                                                                                                         |
| status                | String           | 触发器状态（EVENTGRID触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>● <b>ACTIVE</b></li> <li>● <b>DISABLE</b></li> </ul> |
| trigger_name          | String           | 触发器名称（EVENTGRID触发器参数）。                                                                                                        |

| 参数          | 参数类型             | 描述                                                  |
|-------------|------------------|-----------------------------------------------------|
| event_types | Array of strings | 事件类型（EVENTGRID触发器参数）。例如：OBS:DWR:ObjectCreated:COPY。 |

表 7-587 ApigTriggerFuncInfo

| 参数              | 参数类型    | 描述                                                                                                                                                                                               |
|-----------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| function_urn    | String  | 函数的URN，详细解释见FunctionGraph函数模型的描述。                                                                                                                                                                |
| invocation_type | String  | 调用函数执行方式。<br><ul style="list-style-type: none"> <li>• sync：同步执行</li> <li>• async：异步执行</li> </ul> 枚举值：<br><ul style="list-style-type: none"> <li>• <b>sync</b></li> <li>• <b>async</b></li> </ul> |
| timeout         | Integer | API网关请求函数服务的超时时间（毫秒）。APIG触发器此参数必填。<br>最小值：1<br>最大值：60000                                                                                                                                         |
| version         | String  | 函数版本信息。                                                                                                                                                                                          |

状态码：400

表 7-588 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码：401

表 7-589 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 403**

**表 7-590 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 404**

**表 7-591 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 500**

**表 7-592 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询指定函数所有触发器。

```
GET https://{Endpoint}/v2/{project_id}/fgs/triggers/{function_urn}
```

## 响应示例

**状态码： 200**

OK

```
[{
 "trigger_id": "1b3b264db3b849118d007884b9bc8a60",
 "trigger_type_code": "APIG",
 "trigger_status": "ACTIVE",
 "event_data": {
 "api_id": "4a59f6a7263c4d1dab3f69c34f98d949",
 "api_name": "API_test_triggers",
 "auth": "IAM",
 "env_id": "DEFAULT_ENVIRONMENT_RELEASE_ID",
 "env_name": "RELEASE",
 "func_info": {
```

```
"function_urn" : "urn:fss:xxxx:46b6f338fc3445b8846c71dfb1fbxxx:function:default:test_triggers",
"invocation_type" : "sync",
"timeout" : 5000,
"version" : "latest"
},
"group_id" : "9205f83fe721481eb490870d380cf31d",
"group_name" : "APIGroup_2mbe",
"invoke_url" : "https://ed94ae9494ba4f70ac24224747202140.{apig_endpoint}/test_triggers",
"match_mode" : "SWA",
"name" : "API_test_triggers",
"path" : "/test_triggers",
"protocol" : "HTTPS",
"req_method" : "ANY",
"triggerid" : "1b3b264db3b849118d007884b9bc8a60",
"type" : 1
},
"last_updated_time" : "2022-11-09 16:37:24",
"created_time" : "2022-11-09 16:37:24"
}]
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListFunctionTriggersSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListFunctionTriggersRequest request = new ListFunctionTriggersRequest();
 try {
 ListFunctionTriggersResponse response = client.listFunctionTriggers(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 }
 }
}
```



```
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListFunctionTriggersRequest()
 response = client.list_function_triggers(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
```

```
functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.ListFunctionTriggersRequest{}
response, err := client.ListFunctionTriggers(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.13.3 创建触发器

### 功能介绍

创建触发器。

- 可以创建的触发器类型包括TIMER、APIG、CTS、DDS、DMS、DIS、LTS、OBS、SMN、KAFKA。（当前OBS触发器暂不支持中国站。）
- DDS和KAFKA触发器创建时默认为DISABLED状态，其他触发器默认为ACTIVE状态。
- TIMER、DDS、DMS、KAFKA、LTS触发器支持禁用，其他触发器不支持。

### 调用方法

请参见[如何调用API](#)。

## URI

POST /v2/{project\_id}/fgs/triggers/{function\_urn}

表 7-593 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见 FunctionGraph函数模型的描述。       |

## 请求参数

表 7-594 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                            |
|--------------|------|--------|---------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）<br>缺省值： <b>application/json</b>                    |

表 7-595 请求 Body 参数

| 参数                | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| trigger_type_code | 是    | String | <p>触发器类型。（当前OBS触发器暂不支持中国站。）</p> <ul style="list-style-type: none"> <li>• TIMER: 定时触发器。</li> <li>• APIG: APIGW触发器。</li> <li>• CTS: 云审计触发器，需要先开通云审计服务。</li> <li>• DDS: 文档数据库触发器，需要开启函数vpc。</li> <li>• DMS: 分布式消息服务触发器，需要配置dms委托。</li> <li>• DIS: 数据接入服务触发器，需要配置dis委托。</li> <li>• LTS: 云审计日志服务触发器，需要配置lts委托。</li> <li>• OBS: 对象存储服务触发器。</li> <li>• KAFKA: 专享版本kafka触发器。</li> </ul> <p>枚举值:</p> <ul style="list-style-type: none"> <li>• <b>TIMER</b></li> <li>• <b>APIG</b></li> <li>• <b>CTS</b></li> <li>• <b>DDS</b></li> <li>• <b>DMS</b></li> <li>• <b>DIS</b></li> <li>• <b>LTS</b></li> <li>• <b>OBS</b></li> <li>• <b>SMN</b></li> <li>• <b>KAFKA</b></li> <li>• <b>RABBITMQ</b></li> <li>• <b>DEDICATEDGATEWAY</b></li> <li>• <b>OPENSOURCEKAFKA</b></li> <li>• <b>APIC</b></li> <li>• <b>GeminiDB Mongo</b></li> <li>• <b>EVENTGRID</b></li> <li>• <b>IOTDA</b></li> </ul> |

| 参数              | 是否必选 | 参数类型                                      | 描述                                                                                                                             |
|-----------------|------|-------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------|
| trigger_status  | 否    | String                                    | 触发器状态，取值为ACTIVE,DISABLED。<br>枚举值： <ul style="list-style-type: none"> <li>● <b>ACTIVE</b></li> <li>● <b>DISABLED</b></li> </ul> |
| event_type_code | 否    | String                                    | 消息代码。                                                                                                                          |
| event_data      | 是    | <b>TriggerEventDataRequestBody</b> object | 触发器源事件。                                                                                                                        |

表 7-596 TriggerEventDataRequestBody

| 参数            | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                              |
|---------------|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name          | 否    | String | <ul style="list-style-type: none"> <li>● <b>TIMER</b>触发器：触发器名称</li> <li>● <b>APIG</b>触发器：API名称</li> <li>● <b>CTS</b>触发器：通知名称</li> <li>● <b>OBS</b>触发器：事件通知名称，默认值为触发器id</li> </ul>                                                                                                                                               |
| schedule_type | 否    | String | 定时触发类型（ <b>TIMER</b> 触发器参数）。 <b>TIMER</b> 触发器此参数必填 <ul style="list-style-type: none"> <li>● <b>Rate</b>：指定固定频率（分钟、小时、天数）定期调用函数，单位为分钟时，输入值不能超过60；单位为小时时，输入值不能超过24；单位为天时，输入值不能超过30。</li> <li>● <b>Cron</b>：指定Cron表达式定期调用函数</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>● <b>Rate</b></li> <li>● <b>Cron</b></li> </ul> |
| schedule      | 否    | String | 定时触发规则（ <b>TIMER</b> 触发器参数）。 <b>TIMER</b> 触发器此参数必填。 <ul style="list-style-type: none"> <li>● 触发类型为<b>Rate</b>时对应定时规则</li> <li>● 触发类型为<b>Cron</b>时对应Cron表达式</li> </ul>                                                                                                                                                           |

| 参数         | 是否必选 | 参数类型    | 描述                                                                                                                                                                                                                       |
|------------|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| user_event | 否    | String  | 附加信息（TIMER触发器参数）。当Timer触发器触发函数执行时，执行事件（函数的event参数）为：{"version": "v1.0", "time": "2018-06-01T08:30:00+08:00", "trigger_type": "TIMER", "trigger_name": "Timer_001", "user_event": "您输入的附加信息"}                             |
| type       | 否    | Integer | API接口类型（APIG触发器参数）。APIG触发器此参数必填。<br><ul style="list-style-type: none"> <li>• 1: 公有API</li> <li>• 2: 私有API</li> </ul>                                                                                                     |
| path       | 否    | String  | APIG接口PATH路径（APIG触发器参数）。APIG触发器此参数必填。                                                                                                                                                                                    |
| protocol   | 否    | String  | API的请求协议（APIG触发器参数）。APIG触发器此参数必填。<br>枚举值：<br><ul style="list-style-type: none"> <li>• HTTP</li> <li>• HTTPS</li> </ul>                                                                                                   |
| req_method | 否    | String  | API的请求方式（APIG触发器参数）。APIG触发器此参数必填。<br>枚举值：<br><ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> <li>• HEAD</li> <li>• PATCH</li> <li>• OPTIONS</li> <li>• ANY</li> </ul> |
| group_id   | 否    | String  | API所属的分组编号（APIG触发器参数）。APIG触发器此参数必填。                                                                                                                                                                                      |
| group_name | 否    | String  | API所属的分组名称                                                                                                                                                                                                               |

| 参数           | 是否必选 | 参数类型                       | 描述                                                                                                                                                                                                                                                                                            |
|--------------|------|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| match_mode   | 否    | String                     | API的匹配方式（APIG触发器参数）。APIG触发器此参数必填。<br><ul style="list-style-type: none"> <li>SWA：前缀匹配</li> <li>NORMAL：正常匹配（绝对匹配）</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>SWA</li> <li>NORMAL</li> </ul>                                                                                         |
| env_name     | 否    | String                     | API的发布环境（APIG触发器参数）。APIG触发器此参数必填。                                                                                                                                                                                                                                                             |
| env_id       | 否    | String                     | API的发布环境id（APIG触发器参数）。APIG触发器此参数必填。                                                                                                                                                                                                                                                           |
| auth         | 否    | String                     | API的认证方式（APIG触发器参数）。APIG触发器此参数必填。<br><ul style="list-style-type: none"> <li>IAM：IAM认证，只允许IAM用户能访问，安全级别中等</li> <li>APP：采用Appkey&amp;Appsecret认证，安全级别高，推荐使用</li> <li>NONE：无认证模式，所有用户均可访问，不推荐使用</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>IAM</li> <li>APP</li> <li>NONE</li> </ul> |
| func_info    | 否    | ApigTriggerFuncInfo object | APIG触发器函数 workflow 后端详情（APIG触发器参数）。APIG触发器此参数必填。                                                                                                                                                                                                                                              |
| sl_domain    | 否    | String                     | APIG系统默认分配的子域名（APIG触发器参数）。<br>最小长度：1<br>最大长度：255                                                                                                                                                                                                                                              |
| backend_type | 否    | String                     | API的后端类型（APIG触发器参数）。<br>枚举值： <ul style="list-style-type: none"> <li>FUNCTION</li> </ul>                                                                                                                                                                                                       |

| 参数                | 是否必选 | 参数类型             | 描述                                                                                                                                                                                                                                                         |
|-------------------|------|------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| operations        | 否    | Array of strings | 自定义操作（CTS触发器参数）。CTS触发器此参数必填。CTS云审计服务类型和操作订阅所需要的事件通知，当CTS云审计服务获取已订阅的操作记录后，通过CTS触发器将采集到的操作记录作为参数传递来调用FunctionGraph函数。                                                                                                                                        |
| instance_id       | 否    | String           | 实例id。DDS、KAFKA、RABBITMQ触发器此参数必填。 <ul style="list-style-type: none"> <li>• APIG触发器：apig实例id</li> <li>• DDS触发器：文档数据库实例id</li> <li>• KAFKA触发器：KAFKA实例id</li> <li>• RABBITMQ触发器：RABBITMQ实例id</li> </ul>                                                        |
| collection_name   | 否    | String           | 集合名称（DDS触发器参数）。DDS触发器此参数必填。                                                                                                                                                                                                                                |
| db_name           | 否    | String           | 文档数据库名称（DDS触发器参数）。DDS触发器此参数必填。                                                                                                                                                                                                                             |
| db_password       | 否    | String           | 文档数据库密码（DDS触发器参数）。DDS触发器此参数必填。                                                                                                                                                                                                                             |
| batch_size        | 否    | Integer          | 批处理大小，单次函数执行处理的最大数据量。DIS、DDS、KAFKA、RABBITMQ触发器此参数必填。 <ul style="list-style-type: none"> <li>• DDS触发器：批处理大小设置1-10,000的范围内</li> <li>• DIS触发器：批处理大小设置1-10,000的范围内</li> <li>• KAFKA触发器：批处理大小设置1-1,000的范围内</li> <li>• RABBITMQ触发器：批处理大小设置1-1,000的范围内</li> </ul> |
| queue_id          | 否    | String           | 队列id（DMS触发器参数）。DMS触发器此参数必填。                                                                                                                                                                                                                                |
| consumer_group_id | 否    | String           | 消费组id（DMS触发器参数）。DMS触发器此参数必填。                                                                                                                                                                                                                               |
| polling_interval  | 否    | Integer          | 拉取周期。DIS、DMS触发器此参数必填。                                                                                                                                                                                                                                      |



| 参数                 | 是否必选 | 参数类型    | 描述                                                                                                                                                                                                                                                        |
|--------------------|------|---------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| stream_name        | 否    | String  | 通道名称（DIS触发器参数）。DIS触发器此参数必填。                                                                                                                                                                                                                               |
| sharditerator_type | 否    | String  | 起始位置（DIS触发器参数）。DIS触发器此参数必填。<br><ul style="list-style-type: none"> <li>TRIM_HORIZON: 从最早被存储至分区的有效记录开始读取。</li> <li>LATEST: 从分区中的最新记录开始读取，此设置可以保证总是读到分区中最新记录。</li> </ul> 枚举值: <ul style="list-style-type: none"> <li>TRIM_HORIZON</li> <li>LATEST</li> </ul> |
| polling_unit       | 否    | String  | 拉取周期单位（DIS触发器参数）。DIS触发器此参数必填。<br><ul style="list-style-type: none"> <li>s: 秒</li> <li>ms: 毫秒</li> </ul> 枚举值: <ul style="list-style-type: none"> <li>s</li> <li>ms</li> </ul>                                                                              |
| max_fetch_bytes    | 否    | Integer | 最大提取字节数（DIS触发器参数）。<br>最小值: 0<br>最大值: 4194304                                                                                                                                                                                                              |
| is_serial          | 否    | String  | 串行处理数据（DIS触发器参数），如果开启该选项，取一次数据处理完之后才会取下一次数据；否则只要拉取周期到了就会取数据进行处理。DIS触发器此参数必填。<br>枚举值: <ul style="list-style-type: none"> <li>true</li> <li>false</li> </ul>                                                                                                |
| log_group_id       | 否    | String  | 日志组id（LTS触发器参数）。LTS触发器此参数必填。                                                                                                                                                                                                                              |
| log_topic_id       | 否    | String  | 日志流id（LTS触发器参数）。LTS触发器此参数必填。                                                                                                                                                                                                                              |

| 参数        | 是否必选 | 参数类型             | 描述                                                                                                                                                                                                                                                                                                                                                          |
|-----------|------|------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| bucket    | 否    | String           | 桶名称（OBS触发器参数），用作事件源的OBS存储桶，不能和本用户已有桶重名；不能和其他用户已有的桶重名；创建成功后不支持修改。OBS触发器此参数必填。<br>最小长度：1<br>最大长度：64                                                                                                                                                                                                                                                           |
| prefix    | 否    | String           | 前缀（OBS触发器参数），输入一个可选性前缀来限制对以此关键字开头的对象的通知。<br>最小长度：0<br>最大长度：1024                                                                                                                                                                                                                                                                                             |
| suffix    | 否    | String           | 后缀（OBS触发器参数），输入一个可选性后缀来限制对以此关键字结尾的对象的通知<br>最小长度：0<br>最大长度：1024                                                                                                                                                                                                                                                                                              |
| events    | 否    | Array of strings | 触发事件（OBS触发器参数）。OBS触发器此参数必填。 <ul style="list-style-type: none"> <li>ObjectCreated：表示所有创建对象的操作，包含Put、Post、Copy对象以及合并段</li> <li>Put：使用Put方法上传对象</li> <li>Post：使用Post方法上传对象</li> <li>Copy：使用copy方法复制对象</li> <li>CompleteMultipartUpload：表示合并分段任务</li> <li>ObjectRemoved：表示删除对象</li> <li>Delete：指定对象版本号删除对象</li> <li>DeleteMarkerCreated：不指定对象版本号删除对象</li> </ul> |
| topic_urn | 否    | String           | 主题URN（SMN触发器参数）。SMN触发器此参数必填。<br>最小长度：1<br>最大长度：255                                                                                                                                                                                                                                                                                                          |

| 参数                    | 是否必选 | 参数类型             | 描述                                            |
|-----------------------|------|------------------|-----------------------------------------------|
| topic_ids             | 否    | Array of strings | KAFKA主题id列表（KAFKA触发器参数）。KAFKA触发器此参数必填。        |
| kafka_user            | 否    | String           | KAFKA账户名（KAFKA触发器参数）。                         |
| kafka_password        | 否    | String           | KAFKA账户密码（KAFKA触发器参数）。                        |
| kafka_connect_address | 否    | String           | KAFKA实例连接IP地址（KAFKA触发器参数）。                    |
| kafka_ssl_enable      | 否    | Boolean          | KAFKA连接是否开启安全认证（KAFKA触发器参数）。                  |
| access_password       | 否    | String           | RABBITMQ账户密码（RABBITMQ触发器参数）。RABBITMQ触发器此参数必填。 |
| access_user           | 否    | String           | RABBITMQ账户名（RABBITMQ触发器参数）。                   |
| connect_address       | 否    | String           | 实例连接IP地址（RABBITMQ触发器参数）。                      |
| exchange_name         | 否    | String           | 交换机名称（RABBITMQ触发器参数）。RABBITMQ触发器此参数必填。        |
| vhost                 | 否    | String           | 虚拟机名称（RABBITMQ触发器参数）。                         |
| ssl_enable            | 否    | Boolean          | RABBITMQ连接是否开启安全认证（RABBITMQ触发器参数）。            |
| Key_encode            | 否    | Boolean          | EG obs触发器是否对对象加密（EVENTGRID触发器参数）。             |
| agency                | 否    | String           | 使用的代理（EVENTGRID触发器参数）。                        |
| channel_name          | 否    | String           | 通道名称（EVENTGRID触发器参数）。                         |
| channel_id            | 否    | String           | 通道id（EVENTGRID触发器参数）。                         |
| source_name           | 否    | String           | 事件源名称（EVENTGRID触发器参数）。                        |
| created_time          | 否    | String           | 创建时间（EVENTGRID触发器参数）。                         |

| 参数           | 是否必选 | 参数类型             | 描述                                                                                                                            |
|--------------|------|------------------|-------------------------------------------------------------------------------------------------------------------------------|
| status       | 否    | String           | 触发器状态（EVENTGRID触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>ACTIVE</b></li> <li>• <b>DISABLE</b></li> </ul> |
| trigger_name | 否    | String           | 触发器名称（EVENTGRID触发器参数）。                                                                                                        |
| event_types  | 否    | Array of strings | 事件类型（EVENTGRID触发器参数）。例如：<br>OBS:DWR:ObjectCreated:COPY。                                                                       |

表 7-597 ApigTriggerFuncInfo

| 参数              | 是否必选 | 参数类型    | 描述                                                                                                                                                                                               |
|-----------------|------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| function_urn    | 否    | String  | 函数的URN，详细解释见FunctionGraph函数模型的描述。                                                                                                                                                                |
| invocation_type | 否    | String  | 调用函数执行方式。<br><ul style="list-style-type: none"> <li>• sync：同步执行</li> <li>• async：异步执行</li> </ul> 枚举值：<br><ul style="list-style-type: none"> <li>• <b>sync</b></li> <li>• <b>async</b></li> </ul> |
| timeout         | 是    | Integer | API网关请求函数服务的超时时间（毫秒）。APIG触发器此参数必填。<br>最小值： <b>1</b><br>最大值： <b>60000</b>                                                                                                                         |
| version         | 否    | String  | 函数版本信息。                                                                                                                                                                                          |

## 响应参数

状态码： 201

表 7-598 响应 Body 参数

| 参数                | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| trigger_id        | String | 触发器ID。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| trigger_type_code | String | <p>触发器类型。（当前OBS触发器暂不支持中国站。）</p> <ul style="list-style-type: none"> <li>• TIMER: "定时触发器。"</li> <li>• APIG: "APIG触发器。"</li> <li>• CTS: "云审计服务触发器。"</li> <li>• DDS: "文档数据库服务触发器。"</li> <li>• DMS: "分布式服务触发器。"</li> <li>• DIS: "数据接入服务触发器。"</li> <li>• LTS: "云日志服务触发器。"</li> <li>• OBS: "对象存储触发器。"</li> <li>• SMN: "消息通知服务触发器。"</li> <li>• KAFKA: "专享版消息通知服务触发器。"</li> </ul> <p>枚举值：</p> <ul style="list-style-type: none"> <li>• <b>TIMER</b></li> <li>• <b>APIG</b></li> <li>• <b>CTS</b></li> <li>• <b>DDS</b></li> <li>• <b>DMS</b></li> <li>• <b>DIS</b></li> <li>• <b>LTS</b></li> <li>• <b>OBS</b></li> <li>• <b>SMN</b></li> <li>• <b>KAFKA</b></li> <li>• <b>RABBITMQ</b></li> <li>• <b>DEDICATEDGATEWAY</b></li> <li>• <b>OPENSOURCEKAFKA</b></li> <li>• <b>APIC</b></li> <li>• <b>GeminiDB Mongo</b></li> <li>• <b>EVENTGRID</b></li> <li>• <b>IOTDA</b></li> </ul> |

| 参数                | 参数类型                                | 描述                                                                                                                                                                                     |
|-------------------|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| trigger_status    | String                              | "触发器状态"<br><ul style="list-style-type: none"> <li>ACTIVE: 启用状态。</li> <li>DISABLED: 禁用状态。</li> </ul> 枚举值:<br><ul style="list-style-type: none"> <li>ACTIVE</li> <li>DISABLED</li> </ul> |
| event_data        | TriggerEventDataResponseBody object | 触发器源事件。                                                                                                                                                                                |
| last_updated_time | String                              | 最后更新时间。                                                                                                                                                                                |
| created_time      | String                              | 触发器创建时间。                                                                                                                                                                               |

表 7-599 TriggerEventDataResponseBody

| 参数            | 参数类型   | 描述                                                                                                                                                                                                                                                                   |
|---------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name          | String | 触发器名称                                                                                                                                                                                                                                                                |
| schedule_type | String | 定时触发类型（TIMER触发器参数）。<br><ul style="list-style-type: none"> <li>Rate: 指定固定频率（分钟、小时、天数）定期调用函数，单位为分钟时，输入值不能超过60；单位为小时时，输入值不能超过24；单位为天时，输入值不能超过30。</li> <li>Cron: 指定Cron表达式定期调用函数</li> </ul> 枚举值:<br><ul style="list-style-type: none"> <li>Rate</li> <li>Cron</li> </ul> |
| schedule      | String | 定时触发规则（TIMER触发器参数）。<br><ul style="list-style-type: none"> <li>触发类型为Rate时对应定时规则</li> <li>触发类型为Cron时对应Cron表达式</li> </ul>                                                                                                                                               |
| user_event    | String | 附加信息（TIMER触发器参数）。当Timer触发器触发函数执行时，执行事件（函数的event参数）为：{"version": "v1.0", "time": "2018-06-01T08:30:00+08:00", "trigger_type": "TIMER", "trigger_name": "Timer_001", "user_event": "您输入的附加信息"}                                                                         |
| triggerid     | String | APIG触发器id。（APIG触发器参数）                                                                                                                                                                                                                                                |

| 参数         | 参数类型    | 描述                                                                                                                                                                                                       |
|------------|---------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| type       | Integer | API接口类型（APIG触发器参数）。 <ul style="list-style-type: none"> <li>• 1: 公有API</li> <li>• 2: 私有API</li> </ul>                                                                                                     |
| path       | String  | APIG接口PATH路径（APIG触发器参数）。                                                                                                                                                                                 |
| protocol   | String  | API的请求协议（APIG触发器参数）。<br>枚举值： <ul style="list-style-type: none"> <li>• HTTP</li> <li>• HTTPS</li> </ul>                                                                                                   |
| req_method | String  | API的请求方式（APIG触发器参数）。<br>枚举值： <ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> <li>• HEAD</li> <li>• PATCH</li> <li>• OPTIONS</li> <li>• ANY</li> </ul> |
| group_id   | String  | API所属的分组编号（APIG触发器参数）。                                                                                                                                                                                   |
| group_name | String  | API所属的分组名称（APIG触发器参数）。                                                                                                                                                                                   |
| match_mode | String  | API的匹配方式（APIG触发器参数）。 <ul style="list-style-type: none"> <li>• SWA: 前缀匹配</li> <li>• NORMAL: 正常匹配（绝对匹配）</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• SWA</li> <li>• NORMAL</li> </ul>          |
| env_name   | String  | API的发布环境（APIG触发器参数）。                                                                                                                                                                                     |
| env_id     | String  | API的发布环境id（APIG触发器参数）。                                                                                                                                                                                   |
| api_id     | String  | API编号（APIG触发器参数）。                                                                                                                                                                                        |
| api_name   | String  | API名称（APIG触发器参数）。                                                                                                                                                                                        |

| 参数              | 参数类型                              | 描述                                                                                                                                                                                                                                                                                        |
|-----------------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| auth            | String                            | API的认证方式（APIG触发器参数）。 <ul style="list-style-type: none"> <li>• IAM：IAM认证，只允许IAM用户能访问，安全级别中等</li> <li>• APP：采用Appkey&amp;Appsecret认证，安全级别高，推荐使用</li> <li>• NONE：无认证模式，所有用户均可访问，不推荐使用</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• IAM</li> <li>• APP</li> <li>• NONE</li> </ul> |
| invoke_url      | String                            | API调用地址（APIG触发器参数）。                                                                                                                                                                                                                                                                       |
| func_info       | <b>ApigTriggerFuncInfo</b> object | APIG触发器函数工作流后端详情（APIG触发器参数）。APIG触发器此参数必填。                                                                                                                                                                                                                                                 |
| sl_domain       | String                            | APIG系统默认分配的子域名（APIG触发器参数）。<br>最小长度：1<br>最大长度：255                                                                                                                                                                                                                                          |
| backend_type    | String                            | API的后端类型（APIG触发器参数）。<br>枚举值： <ul style="list-style-type: none"> <li>• FUNCTION</li> </ul>                                                                                                                                                                                                 |
| instance_id     | String                            | 实例id。DDS、KAFKA、RABBITMQ触发器此参数必填。 <ul style="list-style-type: none"> <li>• APIG触发器：apig实例id</li> <li>• DDS触发器：文档数据库实例id</li> <li>• KAFKA触发器：KAFKA实例id</li> <li>• RABBITMQ触发器：RABBITMQ实例id</li> </ul>                                                                                       |
| roma_app_id     | String                            | API归属的集成应用编号。（APIG触发器参数）                                                                                                                                                                                                                                                                  |
| operations      | Array of strings                  | 自定义操作（CTS触发器参数）。CTS云审计服务类型和操作订阅所需要的事件通知，当CTS云审计服务获取已订阅的操作记录后，通过CTS触发器将采集到的操作记录作为参数传递来调用FunctionGraph函数。                                                                                                                                                                                   |
| collection_name | String                            | 集合名称（DDS触发器参数）。                                                                                                                                                                                                                                                                           |
| db_name         | String                            | 文档数据库名称（DDS触发器参数）。                                                                                                                                                                                                                                                                        |
| db_password     | String                            | 文档数据库密码（DDS触发器参数）。                                                                                                                                                                                                                                                                        |



| 参数                 | 参数类型             | 描述                                                                                                                                                                                                                                                             |
|--------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| db_user            | String           | 文档数据库用户名（DDS触发器参数）。                                                                                                                                                                                                                                            |
| instance_addresses | Array of strings | 文档数据库实例地址（DDS触发器参数）。                                                                                                                                                                                                                                           |
| mode               | String           | 文档数据库实例类型（DDS触发器参数）。 <ul style="list-style-type: none"> <li>• Sharding：集群实例</li> <li>• ReplicaSet：副本集实例</li> <li>• Single：单节点实例</li> </ul>                                                                                                                     |
| batch_size         | Integer          | 批处理大小，单次函数执行处理的最大数据量。DIS、DDS、KAFKA、RABBITMQ触发器此参数必填。 <ul style="list-style-type: none"> <li>• DDS触发器：批处理大小设置1-10,000的范围内</li> <li>• DIS触发器：批处理大小设置1-10,000的范围内</li> <li>• KAFKA触发器：批处理大小设置1-1,000的范围内</li> <li>• RABBITMQ触发器：批处理大小设置1-1,000的范围内</li> </ul>     |
| queue_id           | String           | 队列id（DMS触发器参数）。                                                                                                                                                                                                                                                |
| consumer_group_id  | String           | 消费组id（DMS触发器参数）。                                                                                                                                                                                                                                               |
| polling_interval   | Integer          | 拉取周期。                                                                                                                                                                                                                                                          |
| stream_name        | String           | 通道名称（DIS触发器参数）。                                                                                                                                                                                                                                                |
| sharditerator_type | String           | 起始位置（DIS触发器参数）。 <ul style="list-style-type: none"> <li>• TRIM_HORIZON：从最早被存储至分区的有效记录开始读取。</li> <li>• LATEST：从分区中的最新记录开始读取，此设置可以保证总是读到分区中最新记录。</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• <b>TRIM_HORIZON</b></li> <li>• <b>LATEST</b></li> </ul> |
| polling_unit       | String           | 拉取周期单位（DIS触发器参数）。 <ul style="list-style-type: none"> <li>• s：秒</li> <li>• ms：毫秒</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• <b>s</b></li> <li>• <b>ms</b></li> </ul>                                                                              |

| 参数              | 参数类型             | 描述                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| max_fetch_bytes | Integer          | 最大提取字节数（DIS触发器参数）。<br>最小值： <b>0</b><br>最大值： <b>4194304</b>                                                                                                                                                                                                                                                                                                      |
| is_serial       | String           | 串行处理数据（DIS触发器参数），如果开启该选项，取一次数据处理完之后才会取下一次数据；否则只要拉取周期到了就会取数据进行处理。<br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>true</b></li> <li>• <b>false</b></li> </ul>                                                                                                                                                                                             |
| log_group_id    | String           | 日志组id（LTS触发器参数）。                                                                                                                                                                                                                                                                                                                                                |
| log_topic_id    | String           | 日志流id（LTS触发器参数）。                                                                                                                                                                                                                                                                                                                                                |
| bucket          | String           | 桶名称（OBS触发器参数），用作事件源的OBS存储桶，不能和本用户已有桶重名；不能和其他用户已有的桶重名；创建成功后不支持修改。<br>最小长度： <b>1</b><br>最大长度： <b>64</b>                                                                                                                                                                                                                                                           |
| prefix          | String           | 前缀（OBS触发器参数），输入一个可选性前缀来限制对以此关键字开头的对象的通知。<br>最小长度： <b>0</b><br>最大长度： <b>1024</b>                                                                                                                                                                                                                                                                                 |
| suffix          | String           | 后缀（OBS触发器参数），输入一个可选性后缀来限制对以此关键字结尾的对象的通知<br>最小长度： <b>0</b><br>最大长度： <b>1024</b>                                                                                                                                                                                                                                                                                  |
| events          | Array of strings | 触发事件（OBS触发器参数）。 <ul style="list-style-type: none"> <li>• ObjectCreated：表示所有创建对象的操作，包含Put、Post、Copy对象以及合并段</li> <li>• Put：使用Put方法上传对象</li> <li>• Post：使用Post方法上传对象</li> <li>• Copy：使用copy方法复制对象</li> <li>• CompleteMultipartUpload：表示合并分段任务</li> <li>• ObjectRemoved：表示删除对象</li> <li>• Delete：指定对象版本号删除对象</li> <li>• DeleteMarkerCreated：不指定对象版本号删除对象</li> </ul> |

| 参数                    | 参数类型             | 描述                                                                                                                            |
|-----------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------|
| topic_urn             | String           | 主题URN（SMN触发器参数）。<br>最小长度： <b>1</b><br>最大长度： <b>255</b>                                                                        |
| topic_ids             | Array of strings | KAFKA主题id列表（KAFKA触发器参数）。                                                                                                      |
| kafka_user            | String           | KAFKA账户名（KAFKA触发器参数）。                                                                                                         |
| kafka_password        | String           | KAFKA账户密码（KAFKA触发器参数）。                                                                                                        |
| kafka_connect_address | String           | KAFKA实例连接IP地址（KAFKA触发器参数）。                                                                                                    |
| kafka_ssl_enable      | Boolean          | KAFKA连接是否开启安全认证（KAFKA触发器参数）。                                                                                                  |
| access_password       | String           | RABBITMQ账户密码（RABBITMQ触发器参数）。                                                                                                  |
| access_user           | String           | RABBITMQ账户名（RABBITMQ触发器参数）。                                                                                                   |
| connect_address       | String           | 实例连接IP地址（RABBITMQ触发器参数）。                                                                                                      |
| exchange_name         | String           | 交换机名称（RABBITMQ触发器参数）。                                                                                                         |
| vhost                 | String           | 虚拟机名称（RABBITMQ触发器参数）。                                                                                                         |
| ssl_enable            | Boolean          | RABBITMQ连接是否开启安全认证（RABBITMQ触发器参数）。                                                                                            |
| Key_encode            | Boolean          | EG obs触发器是否对对象加密（EVENTGRID触发器参数）。                                                                                             |
| agency                | String           | 使用的代理（EVENTGRID触发器参数）。                                                                                                        |
| channel_name          | String           | 通道名称（EVENTGRID触发器参数）。                                                                                                         |
| channel_id            | String           | 通道id（EVENTGRID触发器参数）。                                                                                                         |
| source_name           | String           | 事件源名称（EVENTGRID触发器参数）。                                                                                                        |
| created_time          | String           | 创建时间（EVENTGRID触发器参数）。                                                                                                         |
| status                | String           | 触发器状态（EVENTGRID触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>● <b>ACTIVE</b></li> <li>● <b>DISABLE</b></li> </ul> |
| trigger_name          | String           | 触发器名称（EVENTGRID触发器参数）。                                                                                                        |

| 参数          | 参数类型             | 描述                                                  |
|-------------|------------------|-----------------------------------------------------|
| event_types | Array of strings | 事件类型（EVENTGRID触发器参数）。例如：OBS:DWR:ObjectCreated:COPY。 |

表 7-600 ApigTriggerFuncInfo

| 参数              | 参数类型    | 描述                                                                                                                                                                                               |
|-----------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| function_urn    | String  | 函数的URN，详细解释见FunctionGraph函数模型的描述。                                                                                                                                                                |
| invocation_type | String  | 调用函数执行方式。<br><ul style="list-style-type: none"> <li>• sync：同步执行</li> <li>• async：异步执行</li> </ul> 枚举值：<br><ul style="list-style-type: none"> <li>• <b>sync</b></li> <li>• <b>async</b></li> </ul> |
| timeout         | Integer | API网关请求函数服务的超时时间（毫秒）。APIG触发器此参数必填。<br>最小值：1<br>最大值：60000                                                                                                                                         |
| version         | String  | 函数版本信息。                                                                                                                                                                                          |

状态码：400

表 7-601 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码：401

表 7-602 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 403**

**表 7-603 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 404**

**表 7-604 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 500**

**表 7-605 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**请求示例**

- 创建名为obs-event-1ec7的obs触发器，触发事件为input-picture2桶中有新文件上传。

```
POST https://{Endpoint}/v2/{project_id}/fgs/triggers/{function_urn}

{
 "trigger_type_code": "OBS",
 "trigger_status": "ACTIVE",
 "event_data": {
 "bucket": "input-picture2",
 "events": ["s3:ObjectCreated:*"],
 "name": "obs-event-1ec7"
 }
}
```

- 创建名为API\_test\_function的apig共享版触发器，创建无认证方式的临时域名，超时时间为5000s。

```
POST https://{Endpoint}/v2/{project_id}/fgs/triggers/{function_urn}

{
 "trigger_type_code": "APIG",
 "trigger_status": "ACTIVE",
 "event_data": {
```

```
"func_info": {
 "timeout": 5000
},
"name": "API_test_function",
"env_id": "DEFAULT_ENVIRONMENT_RELEASE_ID",
"env_name": "RELEASE",
"protocol": "HTTPS",
"auth": "NONE",
"group_id": "318b09eb214a41488c667ecd51667e3x",
"sl_domain": "318b09eb214a41488c667ecd51667e3x.apig.{region}.example.com",
"match_mode": "SWA",
"req_method": "ANY",
"backend_type": "FUNCTION",
"type": 1,
"path": "/test_gxz_v1_billing"
}
}
```

- 创建名为API\_test\_function的apig专享版触发器，创建无认证方式的临时域名，超时时间为5000s。

POST https://{Endpoint}/v2/{project\_id}/fgs/triggers/{function\_urn}

```
{
 "trigger_type_code": "DEDICATEDGATEWAY",
 "trigger_status": "ACTIVE",
 "event_data": {
 "func_info": {
 "timeout": 5000
 },
 "name": "API_test_function",
 "env_id": "DEFAULT_ENVIRONMENT_RELEASE_ID",
 "env_name": "RELEASE",
 "protocol": "HTTPS",
 "auth": "IAM",
 "group_id": "5106210cf79e4xxxx",
 "sl_domain": "5106210cf79e4xxxx.apic.{region_id}.example.com",
 "match_mode": "SWA",
 "req_method": "ANY",
 "backend_type": "FUNCTION",
 "type": 1,
 "path": "/test_gxz_v1_billing"
 }
}
```

- 创建86ad3efcxx实例的kafka触发器，从PUSH\_TOKEN\_RECORD消费组中拉取消息触发函数执行。

POST https://{Endpoint}/v2/{project\_id}/fgs/triggers/{function\_urn}

```
{
 "trigger_type_code": "KAFKA",
 "trigger_status": "DISABLED",
 "event_data": {
 "instance_id": "86ad3efc-c019-460e-91e7-317b1a89f95x",
 "topic_ids": ["PUSH_TOKEN_RECORD"],
 "batch_size": 100
 }
}
```

- 创建timer触发器，每3分钟触发一次函数执行。

POST https://{Endpoint}/v2/{project\_id}/fgs/triggers/{function\_urn}

```
{
 "trigger_type_code": "TIMER",
 "trigger_status": "ACTIVE",
 "event_data": {
 "name": "Timer-l8v2",
 "schedule": "3m",
 "schedule_type": "Rate"
 }
}
```

- 创建指定lts触发器(需配置访问LTS服务权限的委托)。  
POST https://{Endpoint}/v2/{project\_id}/fgs/triggers/{function\_urn}  

```
{
 "trigger_type_code": "LTS",
 "trigger_status": "ACTIVE",
 "event_data": {
 "log_group_id": "ccd3fa30-7c86-4590-8a9d-XXX",
 "log_topic_id": "5b4944dd-9603-481d-9b91-XXX"
 }
}
```
- 创建指定smn触发器。  
POST https://{Endpoint}/v2/{project\_id}/fgs/triggers/{function\_urn}  

```
{
 "trigger_type_code": "SMN",
 "trigger_status": "ACTIVE",
 "event_data": {
 "topic_urn": "urn:smn:{region}:{project_id}:Failed_times"
 }
}
```
- 创建名为eg-obs-trigger的EG OBS应用事件源触发器，触发事件为XXX桶中使用Copy方法复制对象（对象名前缀为images/）。  
POST https://{Endpoint}/v2/{project\_id}/fgs/triggers/{function\_urn}  

```
{
 "trigger_type_code": "EVENTGRID",
 "event_data": {
 "channel_name": "default",
 "channel_id": "xxx",
 "source_name": "HC.OBS.DWR",
 "trigger_name": "eg-obs-trigger",
 "event_types": [
 "OBS:DWR:ObjectCreated:COPY"
],
 "agency": "fgs_to_eg",
 "bucket": "xxx",
 "Key_encode": true,
 "prefix": "images/",
 "suffix": null
 },
 "trigger_status": "ACTIVE"
}
```

## 响应示例

状态码： 201

Created

```
{
 "trigger_id": "9a14fae1-78cf-4185-ac7a-429eb6dc41fb",
 "trigger_type_code": "TIMER",
 "trigger_status": "ACTIVE",
 "event_data": {
 "name": "Timer-cpg3",
 "schedule": "3m",
 "schedule_type": "Rate"
 },
 "last_updated_time": "2022-11-09 16:37:24",
 "created_time": "2022-11-09 16:37:24"
}
```

## SDK 代码示例

SDK代码示例如下。

## Java

- 创建名为obs-event-1ec7的obs触发器，触发事件为input-picture2桶中有新文件上传。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateFunctionTriggerSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateFunctionTriggerRequest request = new CreateFunctionTriggerRequest();
 CreateFunctionTriggerRequestBody body = new CreateFunctionTriggerRequestBody();
 List<String> listEventDataEvents = new ArrayList<>();
 listEventDataEvents.add("s3:ObjectCreated:*");
 TriggerEventDataRequestBody eventDatabody = new TriggerEventDataRequestBody();
 eventDatabody.withName("obs-event-1ec7")
 .withBucket("input-picture2")
 .withEvents(listEventDataEvents);
 body.withEventData(eventDatabody);

 body.withTriggerStatus(CreateFunctionTriggerRequestBody.TriggerStatusEnum.fromValue("ACTIVE"));

 body.withTriggerTypeCode(CreateFunctionTriggerRequestBody.TriggerTypeCodeEnum.fromValue("OBS"
));
 request.withBody(body);
 try {
 CreateFunctionTriggerResponse response = client.createFunctionTrigger(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```



- 创建名为API\_test\_function的apig共享版触发器，创建无认证方式的临时域名，超时时间为5000s。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateFunctionTriggerSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateFunctionTriggerRequest request = new CreateFunctionTriggerRequest();
 CreateFunctionTriggerRequestBody body = new CreateFunctionTriggerRequestBody();
 ApigTriggerFuncInfo funcInfoEventData = new ApigTriggerFuncInfo();
 funcInfoEventData.withTimeout(5000);
 TriggerEventDataRequestBody eventDatabody = new TriggerEventDataRequestBody();
 eventDatabody.withName("API_test_function")
 .withType(1)
 .withPath("/test_gxz_v1_billing")
 .withProtocol(TriggerEventDataRequestBody.ProtocolEnum.fromValue("HTTPS"))
 .withReqMethod(TriggerEventDataRequestBody.RequestMethodEnum.fromValue("ANY"))
 .withGroupId("318b09eb214a41488c667ecd51667e3x")
 .withMatchMode(TriggerEventDataRequestBody.MatchModeEnum.fromValue("SWA"))
 .withEnvName("RELEASE")
 .withEnvId("DEFAULT_ENVIRONMENT_RELEASE_ID")
 .withAuth(TriggerEventDataRequestBody.AuthEnum.fromValue("NONE"))
 .withFuncInfo(funcInfoEventData)
 .withSLDomain("318b09eb214a41488c667ecd51667e3x.apig.{region}.example.com")
 .withBackendType(TriggerEventDataRequestBody.BackendTypeEnum.fromValue("FUNCTION"));
 ;
 body.withEventData(eventDatabody);

 body.withTriggerStatus(CreateFunctionTriggerRequestBody.TriggerStatusEnum.fromValue("ACTIVE"));

 body.withTriggerTypeCode(CreateFunctionTriggerRequestBody.TriggerTypeCodeEnum.fromValue("APIG
 "));
 request.withBody(body);
 try {
 CreateFunctionTriggerResponse response = client.createFunctionTrigger(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 }
 }
}
```

```
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

- 创建名为API\_test\_function的apig专享版触发器，创建无认证方式的临时域名，超时时间为5000s。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateFunctionTriggerSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();

 CreateFunctionTriggerRequest request = new CreateFunctionTriggerRequest();
 CreateFunctionTriggerRequestBody body = new CreateFunctionTriggerRequestBody();
 ApigTriggerFuncInfo funcInfoEventData = new ApigTriggerFuncInfo();
 funcInfoEventData.withTimeout(5000);
 TriggerEventDataRequestBody eventDatabody = new TriggerEventDataRequestBody();
 eventDatabody.withName("API_test_function")
 .withType(1)
 .withPath("/test_gxz_v1_billing")
 .withProtocol(TriggerEventDataRequestBody.ProtocolEnum.fromValue("HTTPS"))
 .withReqMethod(TriggerEventDataRequestBody.ReqMethodEnum.fromValue("ANY"))
 .withGroupId("5106210cf79e4xxxx")
 .withMatchMode(TriggerEventDataRequestBody.MatchModeEnum.fromValue("SWA"))
 .withEnvName("RELEASE")
 .withEnvId("DEFAULT_ENVIRONMENT_RELEASE_ID")
 .withAuth(TriggerEventDataRequestBody.AuthEnum.fromValue("IAM"))
 .withFuncInfo(funcInfoEventData)
 .withSlDomain("5106210cf79e4xxxx.apic.{region_id}.example.com")
 .withBackendType(TriggerEventDataRequestBody.BackendTypeEnum.fromValue("FUNCTION"));

 body.withEventData(eventDatabody);

 body.withTriggerStatus(CreateFunctionTriggerRequestBody.TriggerStatusEnum.fromValue("ACTIVE"));

 body.withTriggerTypeCode(CreateFunctionTriggerRequestBody.TriggerTypeCodeEnum.fromValue("DEDI
 CATEDGATEWAY"));
 request.withBody(body);
 try {
 CreateFunctionTriggerResponse response = client.createFunctionTrigger(request);
 }
 }
}
```

```
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

- 创建86ad3efcxx实例的kafka触发器，从PUSH\_TOKEN\_RECORD消费组中拉取消息触发函数执行。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class CreateFunctionTriggerSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateFunctionTriggerRequest request = new CreateFunctionTriggerRequest();
 CreateFunctionTriggerRequestBody body = new CreateFunctionTriggerRequestBody();
 List<String> listEventDataTopicIds = new ArrayList<>();
 listEventDataTopicIds.add("PUSH_TOKEN_RECORD");
 TriggerEventDataRequestBody eventDatabody = new TriggerEventDataRequestBody();
 eventDatabody.withInstanceId("86ad3efc-c019-460e-91e7-317b1a89f95x")
 .withBatchSize(100)
 .withTopicIds(listEventDataTopicIds);
 body.withEventData(eventDatabody);

 body.withTriggerStatus(CreateFunctionTriggerRequestBody.TriggerStatusEnum.fromValue("DISABLED")
);

 body.withTriggerTypeCode(CreateFunctionTriggerRequestBody.TriggerTypeEnum.fromValue("KAFK
 A"));
 request.withBody(body);
 try {
 CreateFunctionTriggerResponse response = client.createFunctionTrigger(request);
 System.out.println(response.toString());
 }
```

```
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

- 创建timer触发器，每3分钟触发一次函数执行。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateFunctionTriggerSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateFunctionTriggerRequest request = new CreateFunctionTriggerRequest();
 CreateFunctionTriggerRequestBody body = new CreateFunctionTriggerRequestBody();
 TriggerEventDataRequestBody eventDatabody = new TriggerEventDataRequestBody();
 eventDatabody.withName("Timer-l8v2")
 .withScheduleType(TriggerEventDataRequestBody.ScheduleTypeEnum.fromValue("Rate"))
 .withSchedule("3m");
 body.withEventData(eventDatabody);

 body.withTriggerStatus(CreateFunctionTriggerRequestBody.TriggerStatusEnum.fromValue("ACTIVE"));

 body.withTriggerTypeCode(CreateFunctionTriggerRequestBody.TriggerTypeCodeEnum.fromValue("TIME
R"));
 request.withBody(body);
 try {
 CreateFunctionTriggerResponse response = client.createFunctionTrigger(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 }
 }
}
```

```
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

- 创建指定lts触发器(需配置访问LTS服务权限的委托)。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateFunctionTriggerSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();

 CreateFunctionTriggerRequest request = new CreateFunctionTriggerRequest();
 CreateFunctionTriggerRequestBody body = new CreateFunctionTriggerRequestBody();
 TriggerEventDataRequestBody eventDatabody = new TriggerEventDataRequestBody();
 eventDatabody.withLogGroupId("ccd3fa30-7c86-4590-8a9d-XXX")
 .withLogTopicId("5b4944dd-9603-481d-9b91-XXX");
 body.withEventData(eventDatabody);

 body.withTriggerStatus(CreateFunctionTriggerRequestBody.TriggerStatusEnum.fromValue("ACTIVE"));

 body.withTriggerTypeCode(CreateFunctionTriggerRequestBody.TriggerTypeCodeEnum.fromValue("LTS")
);
 request.withBody(body);
 try {
 CreateFunctionTriggerResponse response = client.createFunctionTrigger(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

- 创建指定smn触发器。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateFunctionTriggerSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateFunctionTriggerRequest request = new CreateFunctionTriggerRequest();
 CreateFunctionTriggerRequestBody body = new CreateFunctionTriggerRequestBody();
 TriggerEventDataRequestBody eventDatabody = new TriggerEventDataRequestBody();
 eventDatabody.withTopicUrn("urn:smn:{region}:{project_id}:Failed_times");
 body.withEventData(eventDatabody);

 body.withTriggerStatus(CreateFunctionTriggerRequestBody.TriggerStatusEnum.fromValue("ACTIVE"));

 body.withTriggerTypeCode(CreateFunctionTriggerRequestBody.TriggerTypeCodeEnum.fromValue("SMN"));
 request.withBody(body);
 try {
 CreateFunctionTriggerResponse response = client.createFunctionTrigger(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

- 创建名为obs-event-1ec7的obs触发器，触发事件为input-picture2桶中有新文件上传。

```
coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
```

```
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateFunctionTriggerRequest()
 listEventsEventData = [
 "s3:ObjectCreated:*"
]
 eventDatabody = TriggerEventDataRequestBody(
 name="obs-event-1ec7",
 bucket="input-picture2",
 events=listEventsEventData
)
 request.body = CreateFunctionTriggerRequestBody(
 event_data=eventDatabody,
 trigger_status="ACTIVE",
 trigger_type_code="OBS"
)
 response = client.create_function_trigger(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

- 创建名为API\_test\_function的apig共享版触发器，创建无认证方式的临时域名，超时时间为5000s。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateFunctionTriggerRequest()
```

```
funcInfoEventData = ApigTriggerFuncInfo(
 timeout=5000
)
eventDatabody = TriggerEventDataRequestBody(
 name="API_test_function",
 type=1,
 path="/test_gxz_v1_billing",
 protocol="HTTPS",
 req_method="ANY",
 group_id="318b09eb214a41488c667ecd51667e3x",
 match_mode="SWA",
 env_name="RELEASE",
 env_id="DEFAULT_ENVIRONMENT_RELEASE_ID",
 auth="NONE",
 func_info=funcInfoEventData,
 sl_domain="318b09eb214a41488c667ecd51667e3x.apig.{region}.example.com",
 backend_type="FUNCTION"
)
request.body = CreateFunctionTriggerRequestBody(
 event_data=eventDatabody,
 trigger_status="ACTIVE",
 trigger_type_code="APIG"
)
response = client.create_function_trigger(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

- 创建名为API\_test\_function的apig专享版触发器，创建无认证方式的临时域名，超时时间为5000s。

# coding: utf-8

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 # environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 # environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateFunctionTriggerRequest()
 funcInfoEventData = ApigTriggerFuncInfo(
 timeout=5000
)
 eventDatabody = TriggerEventDataRequestBody(
 name="API_test_function",
 type=1,
 path="/test_gxz_v1_billing",
 protocol="HTTPS",
 req_method="ANY",
 group_id="5106210cf79e4xxxx",
 match_mode="SWA",
```



```
env_name="RELEASE",
env_id="DEFAULT_ENVIRONMENT_RELEASE_ID",
auth="IAM",
func_info=funcInfoEventData,
sl_domain="5106210cf79e4xxxx.apic.{region_id}.example.com",
backend_type="FUNCTION"
)
request.body = CreateFunctionTriggerRequestBody(
 event_data=eventDatabody,
 trigger_status="ACTIVE",
 trigger_type_code="DEDICATEDGATEWAY"
)
response = client.create_function_trigger(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

- 创建86ad3efc实例的kafka触发器，从PUSH\_TOKEN\_RECORD消费组中拉取消息触发函数执行。

```
coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 # environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 # environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateFunctionTriggerRequest()
 listTopicIdsEventData = [
 "PUSH_TOKEN_RECORD"
]
 eventDatabody = TriggerEventDataRequestBody(
 instance_id="86ad3efc-c019-460e-91e7-317b1a89f95x",
 batch_size=100,
 topic_ids=listTopicIdsEventData
)
 request.body = CreateFunctionTriggerRequestBody(
 event_data=eventDatabody,
 trigger_status="DISABLED",
 trigger_type_code="KAFKA"
)
 response = client.create_function_trigger(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

- 创建timer触发器，每3分钟触发一次函数执行。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 # environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 # environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateFunctionTriggerRequest()
 eventDatabody = TriggerEventDataRequestBody(
 name="Timer-l8v2",
 schedule_type="Rate",
 schedule="3m"
)
 request.body = CreateFunctionTriggerRequestBody(
 event_data=eventDatabody,
 trigger_status="ACTIVE",
 trigger_type_code="TIMER"
)
 response = client.create_function_trigger(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

- 创建指定LTS触发器(需配置访问LTS服务权限的委托)。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 # environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 # environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
```

```
request = CreateFunctionTriggerRequest()
eventDatabody = TriggerEventDataRequestBody(
 log_group_id="ccd3fa30-7c86-4590-8a9d-XXX",
 log_topic_id="5b4944dd-9603-481d-9b91-XXX"
)
request.body = CreateFunctionTriggerRequestBody(
 event_data=eventDatabody,
 trigger_status="ACTIVE",
 trigger_type_code="LTS"
)
response = client.create_function_trigger(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

- 创建指定smn触发器。

```
coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 # security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 # environment variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before
 # running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 # environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateFunctionTriggerRequest()
 eventDatabody = TriggerEventDataRequestBody(
 topic_urn="urn:smn:{region}:{project_id}:Failed_times"
)
 request.body = CreateFunctionTriggerRequestBody(
 event_data=eventDatabody,
 trigger_status="ACTIVE",
 trigger_type_code="SMN"
)
 response = client.create_function_trigger(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

- 创建名为obs-event-1ec7的obs触发器，触发事件为input-picture2桶中有新文件上传。

```
package main
```

```
import (
 "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.CreateFunctionTriggerRequest{}
 var listEventsEventData = []string{
 "s3:ObjectCreated:*",
 }
 nameEventData := "obs-event-1ec7"
 bucketEventData := "input-picture2"
 eventDataBody := &model.TriggerEventDataRequestBody{
 Name: &nameEventData,
 Bucket: &bucketEventData,
 Events: &listEventsEventData,
 }
 triggerStatusCreateFunctionTriggerRequestBody :=
model.GetCreateFunctionTriggerRequestBodyTriggerStatusEnum().ACTIVE
 request.Body = &model.CreateFunctionTriggerRequestBody{
 EventData: eventDataBody,
 TriggerStatus: &triggerStatusCreateFunctionTriggerRequestBody,
 TriggerTypeCode: model.GetCreateFunctionTriggerRequestBodyTriggerTypeCodeEnum().OBS,
 }
 response, err := client.CreateFunctionTrigger(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

- 创建名为API\_test\_function的apig共享版触发器，创建无认证方式的临时域名，超时时间为5000s。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
```

running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.CreateFunctionTriggerRequest{
 funcInfoEventData := &model.ApiTriggerFuncInfo{
 Timeout: int32(5000),
 }
 nameEventData := "API_test_function"
 typeEventData := int32(1)
 pathEventData := "/test_gxz_v1_billing"
 protocolEventData := model.GetTriggerEventDataRequestBodyProtocolEnum().HTTPS
 reqMethodEventData := model.GetTriggerEventDataRequestBodyReqMethodEnum().ANY
 groupIdEventData := "318b09eb214a41488c667ecd51667e3x"
 matchModeEventData := model.GetTriggerEventDataRequestBodyMatchModeEnum().SWA
 envNameEventData := "RELEASE"
 envIdEventData := "DEFAULT_ENVIRONMENT_RELEASE_ID"
 authEventData := model.GetTriggerEventDataRequestBodyAuthEnum().NONE
 slDomainEventData := "318b09eb214a41488c667ecd51667e3x.apig.{region}.example.com"
 backendTypeEventData := model.GetTriggerEventDataRequestBodyBackendTypeEnum().FUNCTION
 eventDataBody := &model.TriggerEventDataRequestBody{
 Name: &nameEventData,
 Type: &typeEventData,
 Path: &pathEventData,
 Protocol: &protocolEventData,
 ReqMethod: &reqMethodEventData,
 GroupId: &groupIdEventData,
 MatchMode: &matchModeEventData,
 EnvName: &envNameEventData,
 EnvId: &envIdEventData,
 Auth: &authEventData,
 FuncInfo: funcInfoEventData,
 SlDomain: &slDomainEventData,
 BackendType: &backendTypeEventData,
 }
 triggerStatusCreateFunctionTriggerRequestBody :=
model.GetCreateFunctionTriggerRequestBodyTriggerStatusEnum().ACTIVE
 request.Body = &model.CreateFunctionTriggerRequestBody{
 EventData: eventDataBody,
 TriggerStatus: &triggerStatusCreateFunctionTriggerRequestBody,
 TriggerTypeCode: model.GetCreateFunctionTriggerRequestBodyTriggerTypeCodeEnum().APIG,
 }
 response, err := client.CreateFunctionTrigger(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

- 创建名为API\_test\_function的apig专享版触发器，创建无认证方式的临时域名，超时时间为5000s。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.CreateFunctionTriggerRequest{
 funcInfoEventData := &model.ApiTriggerFuncInfo{
 Timeout: int32(5000),
 }
 nameEventData := "API_test_function"
 typeEventData := int32(1)
 pathEventData := "/test_gxz_v1_billing"
 protocolEventData := model.GetTriggerEventDataRequestBodyProtocolEnum().HTTPS
 reqMethodEventData := model.GetTriggerEventDataRequestBodyReqMethodEnum().ANY
 groupIdEventData := "5106210cf79e4xxxx"
 matchModeEventData := model.GetTriggerEventDataRequestBodyMatchModeEnum().SWA
 envNameEventData := "RELEASE"
 envIdEventData := "DEFAULT_ENVIRONMENT_RELEASE_ID"
 authEventData := model.GetTriggerEventDataRequestBodyAuthEnum().IAM
 slDomainEventData := "5106210cf79e4xxxx.apic.{region_id}.example.com"
 backendTypeEventData := model.GetTriggerEventDataRequestBodyBackendTypeEnum().FUNCTION
 eventDataBody := &model.TriggerEventDataRequestBody{
 Name: &nameEventData,
 Type: &typeEventData,
 Path: &pathEventData,
 Protocol: &protocolEventData,
 ReqMethod: &reqMethodEventData,
 GroupId: &groupIdEventData,
 MatchMode: &matchModeEventData,
 EnvName: &envNameEventData,
 EnvId: &envIdEventData,
 Auth: &authEventData,
 FuncInfo: funcInfoEventData,
 SlDomain: &slDomainEventData,
 BackendType: &backendTypeEventData,
 }
 triggerStatusCreateFunctionTriggerRequestBody :=
 model.GetCreateFunctionTriggerRequestBodyTriggerStatusEnum().ACTIVE
 request.Body = &model.CreateFunctionTriggerRequestBody{
 EventData: eventDataBody,
 TriggerStatus: &triggerStatusCreateFunctionTriggerRequestBody,
 TriggerTypeCode:
 model.GetCreateFunctionTriggerRequestBodyTriggerTypeCodeEnum().DEDICATEDGATEWAY,
 }
 response, err := client.CreateFunctionTrigger(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
 }
}
```

- 创建86ad3efc实例的kafka触发器，从PUSH\_TOKEN\_RECORD消费组中拉取消息触发函数执行。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.CreateFunctionTriggerRequest{
 var listTopicIdsEventData = []string{
 "PUSH_TOKEN_RECORD",
 }
 instanceIdEventData:= "86ad3efc-c019-460e-91e7-317b1a89f95x"
 batchSizeEventData:= int32(100)
 eventDatabody := &model.TriggerEventDataRequestBody{
 InstanceId: &instanceIdEventData,
 BatchSize: &batchSizeEventData,
 TopicIds: &listTopicIdsEventData,
 }
 triggerStatusCreateFunctionTriggerRequestBody:=
 model.GetCreateFunctionTriggerRequestBodyTriggerStatusEnum().DISABLED
 request.Body = &model.CreateFunctionTriggerRequestBody{
 EventData: eventDatabody,
 TriggerStatus: &triggerStatusCreateFunctionTriggerRequestBody,
 TriggerTypeCode: model.GetCreateFunctionTriggerRequestBodyTriggerTypeCodeEnum().KAFKA,
 }
 response, err := client.CreateFunctionTrigger(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
 }
}
```

- 创建timer触发器，每3分钟触发一次函数执行。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
)
```

```
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.CreateFunctionTriggerRequest{
 nameEventData:= "Timer-l8v2"
 scheduleTypeEventData:= model.GetTriggerEventDataRequestBodyScheduleTypeEnum().RATE
 scheduleEventData:= "3m"
 eventDatabody := &model.TriggerEventDataRequestBody{
 Name: &nameEventData,
 ScheduleType: &scheduleTypeEventData,
 Schedule: &scheduleEventData,
 }
 triggerStatusCreateFunctionTriggerRequestBody:=
 model.GetCreateFunctionTriggerRequestBodyTriggerStatusEnum().ACTIVE
 request.Body = &model.CreateFunctionTriggerRequestBody{
 EventData: eventDatabody,
 TriggerStatus: &triggerStatusCreateFunctionTriggerRequestBody,
 TriggerTypeCode: model.GetCreateFunctionTriggerRequestBodyTriggerTypeEnum().TIMER,
 }
 response, err := client.CreateFunctionTrigger(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
 }
}
```

- 创建指定LTS触发器(需配置访问LTS服务权限的委托)。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
```



```
WithAk(ak).
WithSk(sk).
Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.CreateFunctionTriggerRequest{
 logGroupIdEventData:= "ccd3fa30-7c86-4590-8a9d-XXX"
 logTopicIdEventData:= "5b4944dd-9603-481d-9b91-XXX"
 eventDatabody := &model.TriggerEventDataRequestBody{
 LogGroupId: &logGroupIdEventData,
 LogTopicId: &logTopicIdEventData,
 }
}
triggerStatusCreateFunctionTriggerRequestBody:=
model.GetCreateFunctionTriggerRequestBodyTriggerStatusEnum().ACTIVE
request.Body = &model.CreateFunctionTriggerRequestBody{
 EventData: eventDatabody,
 TriggerStatus: &triggerStatusCreateFunctionTriggerRequestBody,
 TriggerTypeCode: model.GetCreateFunctionTriggerRequestBodyTriggerTypeCodeEnum().LTS,
}
response, err := client.CreateFunctionTrigger(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

- 创建指定smn触发器。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before
 // running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local
 // environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.CreateFunctionTriggerRequest{
 topicUrnEventData:= "urn:smn:{region}:{project_id}:Failed_times"
 eventDatabody := &model.TriggerEventDataRequestBody{
 TopicUrn: &topicUrnEventData,
 }
 }
 triggerStatusCreateFunctionTriggerRequestBody:=
```

```
model.CreateFunctionTriggerRequestBodyTriggerTypeEnum().ACTIVE
request.Body = &model.CreateFunctionTriggerRequestBody{
 EventData: eventDataBody,
 TriggerStatus: &triggerStatusCreateFunctionTriggerRequestBody,
 TriggerTypeCode: model.CreateFunctionTriggerRequestBodyTriggerTypeEnum().SMN,
}
response, err := client.CreateFunctionTrigger(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 201 | Created                       |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.13.4 删除触发器

### 功能介绍

删除触发器。

### 调用方法

请参见[如何调用API](#)。

### URI

```
DELETE /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/
{trigger_id}
```

表 7-606 路径参数

| 参数                | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                   |
|-------------------|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| project_id        | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。                                                                                                                                                                                                                                                                                                                                                             |
| function_urn      | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。                                                                                                                                                                                                                                                                                                                                                                    |
| trigger_type_code | 是    | String | 触发器类型代码。（当前OBS触发器暂不支持中国站。）<br>枚举值：<br><ul style="list-style-type: none"> <li>• TIMER</li> <li>• APIG</li> <li>• CTS</li> <li>• DDS</li> <li>• DMS</li> <li>• DIS</li> <li>• LTS</li> <li>• OBS</li> <li>• SMN</li> <li>• KAFKA</li> <li>• RABBITMQ</li> <li>• DEDICATEDGATEWAY</li> <li>• OPENSOURCEKAFKA</li> <li>• APIC</li> <li>• GeminiDB Mongo</li> <li>• EVENTGRID</li> <li>• IOTDA</li> </ul> |
| trigger_id        | 是    | String | 触发器编码。                                                                                                                                                                                                                                                                                                                                                                                               |

## 请求参数

表 7-607 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |

| 参数           | 是否必选 | 参数类型   | 描述                                         |
|--------------|------|--------|--------------------------------------------|
| Content-Type | 是    | String | 消息体的类型（格式）<br>缺省值： <b>application/json</b> |

## 响应参数

状态码：400

表 7-608 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码：401

表 7-609 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码：403

表 7-610 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码：404

表 7-611 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-612 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

删除指定触发器。

```
DELETE https://{Endpoint}/v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class DeleteFunctionTriggerSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 DeleteFunctionTriggerRequest request = new DeleteFunctionTriggerRequest();
 try {
 DeleteFunctionTriggerResponse response = client.deleteFunctionTrigger(request);
 System.out.println(response.toString());
 }
 }
}
```

```
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = DeleteFunctionTriggerRequest()
 response = client.delete_function_trigger(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")
```

```
auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.DeleteFunctionTriggerRequest{}
response, err := client.DeleteFunctionTrigger(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 204 | No Content                    |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.13.5 获取指定触发器的信息

### 功能介绍

获取特定触发器的信息。

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/fgs/triggers/{function\_urn}/{trigger\_type\_code}/{trigger\_id}

表 7-613 路径参数

| 参数                | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                  |
|-------------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| project_id        | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。                                                                                                                                                                                                                                                                                                                                                            |
| function_urn      | 是    | String | 函数的URN，详细解释见 FunctionGraph 函数模型的描述。                                                                                                                                                                                                                                                                                                                                                                 |
| trigger_type_code | 是    | String | 触发器类型代码。（当前 OBS 触发器暂不支持中国站。）<br>枚举值： <ul style="list-style-type: none"> <li>• TIMER</li> <li>• APIG</li> <li>• CTS</li> <li>• DDS</li> <li>• DMS</li> <li>• DIS</li> <li>• LTS</li> <li>• OBS</li> <li>• SMN</li> <li>• KAFKA</li> <li>• RABBITMQ</li> <li>• DEDICATEDGATEWAY</li> <li>• OPENSOURCEKAFKA</li> <li>• APIC</li> <li>• GeminiDB Mongo</li> <li>• EVENTGRID</li> <li>• IOTDA</li> </ul> |
| trigger_id        | 是    | String | 触发器编码。                                                                                                                                                                                                                                                                                                                                                                                              |



## 请求参数

表 7-614 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）<br>缺省值： <b>application/json</b>                   |

## 响应参数

状态码： 200

表 7-615 响应 Body 参数

| 参数         | 参数类型   | 描述     |
|------------|--------|--------|
| trigger_id | String | 触发器ID。 |

| 参数                | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| trigger_type_code | String | <p>触发器类型。（当前OBS触发器暂不支持中国站。）</p> <ul style="list-style-type: none"> <li>● TIMER: "定时触发器。"</li> <li>● APIG: "APIG触发器。"</li> <li>● CTS: "云审计服务触发器。"</li> <li>● DDS: "文档数据库服务触发器。"</li> <li>● DMS: "分布式服务触发器。"</li> <li>● DIS: "数据接入服务触发器。"</li> <li>● LTS: "云日志服务触发器。"</li> <li>● OBS: "对象存储触发器。"</li> <li>● SMN: "消息通知服务触发器。"</li> <li>● KAFKA: "专享版消息通知服务触发器。"</li> </ul> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>TIMER</b></li> <li>● <b>APIG</b></li> <li>● <b>CTS</b></li> <li>● <b>DDS</b></li> <li>● <b>DMS</b></li> <li>● <b>DIS</b></li> <li>● <b>LTS</b></li> <li>● <b>OBS</b></li> <li>● <b>SMN</b></li> <li>● <b>KAFKA</b></li> <li>● <b>RABBITMQ</b></li> <li>● <b>DEDICATEDGATEWAY</b></li> <li>● <b>OPENSOURCEKAFKA</b></li> <li>● <b>APIC</b></li> <li>● <b>GeminiDB Mongo</b></li> <li>● <b>EVENTGRID</b></li> <li>● <b>IOTDA</b></li> </ul> |
| trigger_status    | String | <p>"触发器状态"</p> <ul style="list-style-type: none"> <li>● ACTIVE: 启用状态。</li> <li>● DISABLED: 禁用状态。</li> </ul> <p>枚举值:</p> <ul style="list-style-type: none"> <li>● <b>ACTIVE</b></li> <li>● <b>DISABLED</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |

| 参数                | 参数类型                                       | 描述       |
|-------------------|--------------------------------------------|----------|
| event_data        | <b>TriggerEventDataResponseBody</b> object | 触发器源事件。  |
| last_updated_time | String                                     | 最后更新时间。  |
| created_time      | String                                     | 触发器创建时间。 |

表 7-616 TriggerEventDataResponseBody

| 参数            | 参数类型    | 描述                                                                                                                                                                                                                                                                  |
|---------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name          | String  | 触发器名称                                                                                                                                                                                                                                                               |
| schedule_type | String  | 定时触发类型（TIMER触发器参数）。 <ul style="list-style-type: none"> <li>Rate: 指定固定频率（分钟、小时、天数）定期调用函数，单位为分钟时，输入值不能超过60；单位为小时时，输入值不能超过24；单位为天时，输入值不能超过30。</li> <li>Cron: 指定Cron表达式定期调用函数<br/>枚举值： <ul style="list-style-type: none"> <li>Rate</li> <li>Cron</li> </ul> </li> </ul> |
| schedule      | String  | 定时触发规则（TIMER触发器参数）。 <ul style="list-style-type: none"> <li>触发类型为Rate时对应定时规则</li> <li>触发类型为Cron时对应Cron表达式</li> </ul>                                                                                                                                                 |
| user_event    | String  | 附加信息（TIMER触发器参数）。当Timer触发器触发函数执行时，执行事件（函数的event参数）为：{"version": "v1.0", "time": "2018-06-01T08:30:00+08:00", "trigger_type": "TIMER", "trigger_name": "Timer_001", "user_event": "您输入的附加信息"}                                                                        |
| triggerid     | String  | APIG触发器id。（APIG触发器参数）                                                                                                                                                                                                                                               |
| type          | Integer | API接口类型（APIG触发器参数）。 <ul style="list-style-type: none"> <li>1: 公有API</li> <li>2: 私有API</li> </ul>                                                                                                                                                                    |
| path          | String  | APIG接口PATH路径（APIG触发器参数）。                                                                                                                                                                                                                                            |

| 参数         | 参数类型   | 描述                                                                                                                                                                                                          |
|------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| protocol   | String | API的请求协议（APIG触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>• HTTP</li> <li>• HTTPS</li> </ul>                                                                                                   |
| req_method | String | API的请求方式（APIG触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> <li>• HEAD</li> <li>• PATCH</li> <li>• OPTIONS</li> <li>• ANY</li> </ul> |
| group_id   | String | API所属的分组编号（APIG触发器参数）。                                                                                                                                                                                      |
| group_name | String | API所属的分组名称（APIG触发器参数）。                                                                                                                                                                                      |
| match_mode | String | API的匹配方式（APIG触发器参数）。<br><ul style="list-style-type: none"> <li>• SWA：前缀匹配</li> <li>• NORMAL：正常匹配（绝对匹配）</li> </ul> 枚举值：<br><ul style="list-style-type: none"> <li>• SWA</li> <li>• NORMAL</li> </ul>         |
| env_name   | String | API的发布环境（APIG触发器参数）。                                                                                                                                                                                        |
| env_id     | String | API的发布环境id（APIG触发器参数）。                                                                                                                                                                                      |
| api_id     | String | API编号（APIG触发器参数）。                                                                                                                                                                                           |
| api_name   | String | API名称（APIG触发器参数）。                                                                                                                                                                                           |

| 参数              | 参数类型                              | 描述                                                                                                                                                                                                                                                                                        |
|-----------------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| auth            | String                            | API的认证方式（APIG触发器参数）。 <ul style="list-style-type: none"> <li>• IAM：IAM认证，只允许IAM用户能访问，安全级别中等</li> <li>• APP：采用Appkey&amp;Appsecret认证，安全级别高，推荐使用</li> <li>• NONE：无认证模式，所有用户均可访问，不推荐使用</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• IAM</li> <li>• APP</li> <li>• NONE</li> </ul> |
| invoke_url      | String                            | API调用地址（APIG触发器参数）。                                                                                                                                                                                                                                                                       |
| func_info       | <b>ApigTriggerFuncInfo</b> object | APIG触发器函数工作流后端详情（APIG触发器参数）。APIG触发器此参数必填。                                                                                                                                                                                                                                                 |
| sl_domain       | String                            | APIG系统默认分配的子域名（APIG触发器参数）。<br>最小长度：1<br>最大长度：255                                                                                                                                                                                                                                          |
| backend_type    | String                            | API的后端类型（APIG触发器参数）。<br>枚举值： <ul style="list-style-type: none"> <li>• FUNCTION</li> </ul>                                                                                                                                                                                                 |
| instance_id     | String                            | 实例id。DDS、KAFKA、RABBITMQ触发器此参数必填。 <ul style="list-style-type: none"> <li>• APIG触发器：apig实例id</li> <li>• DDS触发器：文档数据库实例id</li> <li>• KAFKA触发器：KAFKA实例id</li> <li>• RABBITMQ触发器：RABBITMQ实例id</li> </ul>                                                                                       |
| roma_app_id     | String                            | API归属的集成应用编号。（APIG触发器参数）                                                                                                                                                                                                                                                                  |
| operations      | Array of strings                  | 自定义操作（CTS触发器参数）。CTS云审计服务类型和操作订阅所需要的事件通知，当CTS云审计服务获取已订阅的操作记录后，通过CTS触发器将采集到的操作记录作为参数传递来调用FunctionGraph函数。                                                                                                                                                                                   |
| collection_name | String                            | 集合名称（DDS触发器参数）。                                                                                                                                                                                                                                                                           |
| db_name         | String                            | 文档数据库名称（DDS触发器参数）。                                                                                                                                                                                                                                                                        |
| db_password     | String                            | 文档数据库密码（DDS触发器参数）。                                                                                                                                                                                                                                                                        |

| 参数                 | 参数类型             | 描述                                                                                                                                                                                                                                                             |
|--------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| db_user            | String           | 文档数据库用户名（DDS触发器参数）。                                                                                                                                                                                                                                            |
| instance_addresses | Array of strings | 文档数据库实例地址（DDS触发器参数）。                                                                                                                                                                                                                                           |
| mode               | String           | 文档数据库实例类型（DDS触发器参数）。 <ul style="list-style-type: none"> <li>• Sharding：集群实例</li> <li>• ReplicaSet：副本集实例</li> <li>• Single：单节点实例</li> </ul>                                                                                                                     |
| batch_size         | Integer          | 批处理大小，单次函数执行处理的最大数据量。DIS、DDS、KAFKA、RABBITMQ触发器此参数必填。 <ul style="list-style-type: none"> <li>• DDS触发器：批处理大小设置1-10,000的范围内</li> <li>• DIS触发器：批处理大小设置1-10,000的范围内</li> <li>• KAFKA触发器：批处理大小设置1-1,000的范围内</li> <li>• RABBITMQ触发器：批处理大小设置1-1,000的范围内</li> </ul>     |
| queue_id           | String           | 队列id（DMS触发器参数）。                                                                                                                                                                                                                                                |
| consumer_group_id  | String           | 消费组id（DMS触发器参数）。                                                                                                                                                                                                                                               |
| polling_interval   | Integer          | 拉取周期。                                                                                                                                                                                                                                                          |
| stream_name        | String           | 通道名称（DIS触发器参数）。                                                                                                                                                                                                                                                |
| sharditerator_type | String           | 起始位置（DIS触发器参数）。 <ul style="list-style-type: none"> <li>• TRIM_HORIZON：从最早被存储至分区的有效记录开始读取。</li> <li>• LATEST：从分区中的最新记录开始读取，此设置可以保证总是读到分区中最新记录。</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• <b>TRIM_HORIZON</b></li> <li>• <b>LATEST</b></li> </ul> |
| polling_unit       | String           | 拉取周期单位（DIS触发器参数）。 <ul style="list-style-type: none"> <li>• s：秒</li> <li>• ms：毫秒</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• <b>s</b></li> <li>• <b>ms</b></li> </ul>                                                                              |

| 参数              | 参数类型             | 描述                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| max_fetch_bytes | Integer          | 最大提取字节数（DIS触发器参数）。<br>最小值： <b>0</b><br>最大值： <b>4194304</b>                                                                                                                                                                                                                                                                                                      |
| is_serial       | String           | 串行处理数据（DIS触发器参数），如果开启该选项，取一次数据处理完之后才会取下一次数据；否则只要拉取周期到了就会取数据进行处理。<br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>true</b></li> <li>• <b>false</b></li> </ul>                                                                                                                                                                                             |
| log_group_id    | String           | 日志组id（LTS触发器参数）。                                                                                                                                                                                                                                                                                                                                                |
| log_topic_id    | String           | 日志流id（LTS触发器参数）。                                                                                                                                                                                                                                                                                                                                                |
| bucket          | String           | 桶名称（OBS触发器参数），用作事件源的OBS存储桶，不能和本用户已有桶重名；不能和其他用户已有的桶重名；创建成功后不支持修改。<br>最小长度： <b>1</b><br>最大长度： <b>64</b>                                                                                                                                                                                                                                                           |
| prefix          | String           | 前缀（OBS触发器参数），输入一个可选性前缀来限制对以此关键字开头的对象的通知。<br>最小长度： <b>0</b><br>最大长度： <b>1024</b>                                                                                                                                                                                                                                                                                 |
| suffix          | String           | 后缀（OBS触发器参数），输入一个可选性后缀来限制对以此关键字结尾的对象的通知<br>最小长度： <b>0</b><br>最大长度： <b>1024</b>                                                                                                                                                                                                                                                                                  |
| events          | Array of strings | 触发事件（OBS触发器参数）。 <ul style="list-style-type: none"> <li>• ObjectCreated：表示所有创建对象的操作，包含Put、Post、Copy对象以及合并段</li> <li>• Put：使用Put方法上传对象</li> <li>• Post：使用Post方法上传对象</li> <li>• Copy：使用copy方法复制对象</li> <li>• CompleteMultipartUpload：表示合并分段任务</li> <li>• ObjectRemoved：表示删除对象</li> <li>• Delete：指定对象版本号删除对象</li> <li>• DeleteMarkerCreated：不指定对象版本号删除对象</li> </ul> |

| 参数                    | 参数类型             | 描述                                                                                                                            |
|-----------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------|
| topic_urn             | String           | 主题URN（SMN触发器参数）。<br>最小长度： <b>1</b><br>最大长度： <b>255</b>                                                                        |
| topic_ids             | Array of strings | KAFKA主题id列表（KAFKA触发器参数）。                                                                                                      |
| kafka_user            | String           | KAFKA账户名（KAFKA触发器参数）。                                                                                                         |
| kafka_password        | String           | KAFKA账户密码（KAFKA触发器参数）。                                                                                                        |
| kafka_connect_address | String           | KAFKA实例连接IP地址（KAFKA触发器参数）。                                                                                                    |
| kafka_ssl_enable      | Boolean          | KAFKA连接是否开启安全认证（KAFKA触发器参数）。                                                                                                  |
| access_password       | String           | RABBITMQ账户密码（RABBITMQ触发器参数）。                                                                                                  |
| access_user           | String           | RABBITMQ账户名（RABBITMQ触发器参数）。                                                                                                   |
| connect_address       | String           | 实例连接IP地址（RABBITMQ触发器参数）。                                                                                                      |
| exchange_name         | String           | 交换机名称（RABBITMQ触发器参数）。                                                                                                         |
| vhost                 | String           | 虚拟机名称（RABBITMQ触发器参数）。                                                                                                         |
| ssl_enable            | Boolean          | RABBITMQ连接是否开启安全认证（RABBITMQ触发器参数）。                                                                                            |
| Key_encode            | Boolean          | EG obs触发器是否对对象加密（EVENTGRID触发器参数）。                                                                                             |
| agency                | String           | 使用的代理（EVENTGRID触发器参数）。                                                                                                        |
| channel_name          | String           | 通道名称（EVENTGRID触发器参数）。                                                                                                         |
| channel_id            | String           | 通道id（EVENTGRID触发器参数）。                                                                                                         |
| source_name           | String           | 事件源名称（EVENTGRID触发器参数）。                                                                                                        |
| created_time          | String           | 创建时间（EVENTGRID触发器参数）。                                                                                                         |
| status                | String           | 触发器状态（EVENTGRID触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>● <b>ACTIVE</b></li> <li>● <b>DISABLE</b></li> </ul> |
| trigger_name          | String           | 触发器名称（EVENTGRID触发器参数）。                                                                                                        |



| 参数          | 参数类型             | 描述                                                  |
|-------------|------------------|-----------------------------------------------------|
| event_types | Array of strings | 事件类型（EVENTGRID触发器参数）。例如：OBS:DWR:ObjectCreated:COPY。 |

表 7-617 ApigTriggerFuncInfo

| 参数              | 参数类型    | 描述                                                                                                                                                                                               |
|-----------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| function_urn    | String  | 函数的URN，详细解释见FunctionGraph函数模型的描述。                                                                                                                                                                |
| invocation_type | String  | 调用函数执行方式。<br><ul style="list-style-type: none"> <li>• sync：同步执行</li> <li>• async：异步执行</li> </ul> 枚举值：<br><ul style="list-style-type: none"> <li>• <b>sync</b></li> <li>• <b>async</b></li> </ul> |
| timeout         | Integer | API网关请求函数服务的超时时间（毫秒）。APIG触发器此参数必填。<br>最小值：1<br>最大值：60000                                                                                                                                         |
| version         | String  | 函数版本信息。                                                                                                                                                                                          |

状态码：400

表 7-618 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码：401

表 7-619 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-620 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-621 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-622 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询指定触发器。

```
GET https://{Endpoint}/v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}
```

## 响应示例

状态码： 200

OK

```
{
 "trigger_id": "9a14fae1-78cf-4185-ac7a-429eb6dc41fb",
 "trigger_type_code": "TIMER",
 "trigger_status": "ACTIVE",
 "event_data": {
 "name": "Timer-cpg3",
 "schedule": "3m",
 "schedule_type": "Rate"
 },
 "last_updated_time": "2022-11-09 16:37:24",
}
```

```
"created_time" : "2019-10-29171553"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowFunctionTriggerSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowFunctionTriggerRequest request = new ShowFunctionTriggerRequest();
 try {
 ShowFunctionTriggerResponse response = client.showFunctionTrigger(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *
```

```
if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowFunctionTriggerRequest()
 response = client.show_function_trigger(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowFunctionTriggerRequest{}
 response, err := client.ShowFunctionTrigger(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | OK                            |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.13.6 更新触发器

### 功能介绍

更新触发器

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/fgs/triggers/{function\_urn}/{trigger\_type\_code}/{trigger\_id}

表 7-623 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

| 参数                | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                |
|-------------------|------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| trigger_type_code | 是    | String | 触发器类型代码。（当前OBS触发器暂不支持中国站。）<br>枚举值： <ul style="list-style-type: none"> <li>• TIMER</li> <li>• APIG</li> <li>• CTS</li> <li>• DDS</li> <li>• DMS</li> <li>• DIS</li> <li>• LTS</li> <li>• OBS</li> <li>• SMN</li> <li>• KAFKA</li> <li>• RABBITMQ</li> <li>• DEDICATEDGATEWAY</li> <li>• OPENSOURCEKAFKA</li> <li>• APIC</li> <li>• GeminiDB Mongo</li> <li>• EVENTGRID</li> <li>• IOTDA</li> </ul> |
| trigger_id        | 是    | String | 触发器编码。                                                                                                                                                                                                                                                                                                                                                                                            |

## 请求参数

表 7-624 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）<br>缺省值： <b>application/json</b>                   |

表 7-625 请求 Body 参数

| 参数             | 是否必选 | 参数类型                                          | 描述                                                                                        |
|----------------|------|-----------------------------------------------|-------------------------------------------------------------------------------------------|
| trigger_status | 否    | String                                        | 触发器状态<br>枚举值： <ul style="list-style-type: none"><li>• ACTIVE</li><li>• DISABLED</li></ul> |
| event_data     | 否    | Array of <b>UpdateriggerEventData</b> objects | 触发器更新事件                                                                                   |

表 7-626 UpdateriggerEventData

| 参数               | 是否必选 | 参数类型    | 描述                                                                              |
|------------------|------|---------|---------------------------------------------------------------------------------|
| is_serial        | 否    | Boolean | 串行处理数据                                                                          |
| max_fetch_bytes  | 否    | Integer | 最大字节数                                                                           |
| polling_interval | 否    | Integer | 拉取周期                                                                            |
| polling_unit     | 否    | String  | 拉取周期单位<br>枚举值： <ul style="list-style-type: none"><li>• ms</li><li>• s</li></ul> |

## 响应参数

状态码： 200

表 7-627 响应 Body 参数

| 参数         | 参数类型   | 描述     |
|------------|--------|--------|
| trigger_id | String | 触发器ID。 |

| 参数                | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-------------------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| trigger_type_code | String | <p>触发器类型。（当前OBS触发器暂不支持中国站。）</p> <ul style="list-style-type: none"> <li>• TIMER: "定时触发器。"</li> <li>• APIG: "APIG触发器。"</li> <li>• CTS: "云审计服务触发器。"</li> <li>• DDS: "文档数据库服务触发器。"</li> <li>• DMS: "分布式服务触发器。"</li> <li>• DIS: "数据接入服务触发器。"</li> <li>• LTS: "云日志服务触发器。"</li> <li>• OBS: "对象存储触发器。"</li> <li>• SMN: "消息通知服务触发器。"</li> <li>• KAFKA: "专享版消息通知服务触发器。"</li> </ul> <p>枚举值:</p> <ul style="list-style-type: none"> <li>• <b>TIMER</b></li> <li>• <b>APIG</b></li> <li>• <b>CTS</b></li> <li>• <b>DDS</b></li> <li>• <b>DMS</b></li> <li>• <b>DIS</b></li> <li>• <b>LTS</b></li> <li>• <b>OBS</b></li> <li>• <b>SMN</b></li> <li>• <b>KAFKA</b></li> <li>• <b>RABBITMQ</b></li> <li>• <b>DEDICATEDGATEWAY</b></li> <li>• <b>OPENSOURCEKAFKA</b></li> <li>• <b>APIC</b></li> <li>• <b>GeminiDB Mongo</b></li> <li>• <b>EVENTGRID</b></li> <li>• <b>IOTDA</b></li> </ul> |
| trigger_status    | String | <p>"触发器状态"</p> <ul style="list-style-type: none"> <li>• ACTIVE: 启用状态。</li> <li>• DISABLED: 禁用状态。</li> </ul> <p>枚举值:</p> <ul style="list-style-type: none"> <li>• <b>ACTIVE</b></li> <li>• <b>DISABLED</b></li> </ul>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |



| 参数                | 参数类型                                       | 描述       |
|-------------------|--------------------------------------------|----------|
| event_data        | <b>TriggerEventDataResponseBody</b> object | 触发器源事件。  |
| last_updated_time | String                                     | 最后更新时间。  |
| created_time      | String                                     | 触发器创建时间。 |

表 7-628 TriggerEventDataResponseBody

| 参数            | 参数类型    | 描述                                                                                                                                                                                                                                                                  |
|---------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name          | String  | 触发器名称                                                                                                                                                                                                                                                               |
| schedule_type | String  | 定时触发类型（TIMER触发器参数）。 <ul style="list-style-type: none"><li>• Rate: 指定固定频率（分钟、小时、天数）定期调用函数，单位为分钟时，输入值不能超过60；单位为小时时，输入值不能超过24；单位为天时，输入值不能超过30。</li><li>• Cron: 指定Cron表达式定期调用函数<br/>枚举值：<ul style="list-style-type: none"><li>• Rate</li><li>• Cron</li></ul></li></ul> |
| schedule      | String  | 定时触发规则（TIMER触发器参数）。 <ul style="list-style-type: none"><li>• 触发类型为Rate时对应定时规则</li><li>• 触发类型为Cron时对应Cron表达式</li></ul>                                                                                                                                                |
| user_event    | String  | 附加信息（TIMER触发器参数）。当Timer触发器触发函数执行时，执行事件（函数的event参数）为：{"version": "v1.0", "time": "2018-06-01T08:30:00+08:00", "trigger_type": "TIMER", "trigger_name": "Timer_001", "user_event": "您输入的附加信息"}                                                                        |
| triggerid     | String  | APIG触发器id。（APIG触发器参数）                                                                                                                                                                                                                                               |
| type          | Integer | API接口类型（APIG触发器参数）。 <ul style="list-style-type: none"><li>• 1: 公有API</li><li>• 2: 私有API</li></ul>                                                                                                                                                                   |
| path          | String  | APIG接口PATH路径（APIG触发器参数）。                                                                                                                                                                                                                                            |

| 参数         | 参数类型   | 描述                                                                                                                                                                                                          |
|------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| protocol   | String | API的请求协议（APIG触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>• HTTP</li> <li>• HTTPS</li> </ul>                                                                                                   |
| req_method | String | API的请求方式（APIG触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>• GET</li> <li>• POST</li> <li>• PUT</li> <li>• DELETE</li> <li>• HEAD</li> <li>• PATCH</li> <li>• OPTIONS</li> <li>• ANY</li> </ul> |
| group_id   | String | API所属的分组编号（APIG触发器参数）。                                                                                                                                                                                      |
| group_name | String | API所属的分组名称（APIG触发器参数）。                                                                                                                                                                                      |
| match_mode | String | API的匹配方式（APIG触发器参数）。<br><ul style="list-style-type: none"> <li>• SWA：前缀匹配</li> <li>• NORMAL：正常匹配（绝对匹配）</li> </ul> 枚举值：<br><ul style="list-style-type: none"> <li>• SWA</li> <li>• NORMAL</li> </ul>         |
| env_name   | String | API的发布环境（APIG触发器参数）。                                                                                                                                                                                        |
| env_id     | String | API的发布环境id（APIG触发器参数）。                                                                                                                                                                                      |
| api_id     | String | API编号（APIG触发器参数）。                                                                                                                                                                                           |
| api_name   | String | API名称（APIG触发器参数）。                                                                                                                                                                                           |

| 参数              | 参数类型                              | 描述                                                                                                                                                                                                                                                                                        |
|-----------------|-----------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| auth            | String                            | API的认证方式（APIG触发器参数）。 <ul style="list-style-type: none"> <li>• IAM：IAM认证，只允许IAM用户能访问，安全级别中等</li> <li>• APP：采用Appkey&amp;Appsecret认证，安全级别高，推荐使用</li> <li>• NONE：无认证模式，所有用户均可访问，不推荐使用</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• IAM</li> <li>• APP</li> <li>• NONE</li> </ul> |
| invoke_url      | String                            | API调用地址（APIG触发器参数）。                                                                                                                                                                                                                                                                       |
| func_info       | <b>ApigTriggerFuncInfo</b> object | APIG触发器函数工作流后端详情（APIG触发器参数）。APIG触发器此参数必填。                                                                                                                                                                                                                                                 |
| sl_domain       | String                            | APIG系统默认分配的子域名（APIG触发器参数）。<br>最小长度：1<br>最大长度：255                                                                                                                                                                                                                                          |
| backend_type    | String                            | API的后端类型（APIG触发器参数）。<br>枚举值： <ul style="list-style-type: none"> <li>• FUNCTION</li> </ul>                                                                                                                                                                                                 |
| instance_id     | String                            | 实例id。DDS、KAFKA、RABBITMQ触发器此参数必填。 <ul style="list-style-type: none"> <li>• APIG触发器：apig实例id</li> <li>• DDS触发器：文档数据库实例id</li> <li>• KAFKA触发器：KAFKA实例id</li> <li>• RABBITMQ触发器：RABBITMQ实例id</li> </ul>                                                                                       |
| roma_app_id     | String                            | API归属的集成应用编号。（APIG触发器参数）                                                                                                                                                                                                                                                                  |
| operations      | Array of strings                  | 自定义操作（CTS触发器参数）。CTS云审计服务类型和操作订阅所需要的事件通知，当CTS云审计服务获取已订阅的操作记录后，通过CTS触发器将采集到的操作记录作为参数传递来调用FunctionGraph函数。                                                                                                                                                                                   |
| collection_name | String                            | 集合名称（DDS触发器参数）。                                                                                                                                                                                                                                                                           |
| db_name         | String                            | 文档数据库名称（DDS触发器参数）。                                                                                                                                                                                                                                                                        |
| db_password     | String                            | 文档数据库密码（DDS触发器参数）。                                                                                                                                                                                                                                                                        |

| 参数                 | 参数类型             | 描述                                                                                                                                                                                                                                                             |
|--------------------|------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| db_user            | String           | 文档数据库用户名（DDS触发器参数）。                                                                                                                                                                                                                                            |
| instance_addresses | Array of strings | 文档数据库实例地址（DDS触发器参数）。                                                                                                                                                                                                                                           |
| mode               | String           | 文档数据库实例类型（DDS触发器参数）。 <ul style="list-style-type: none"> <li>• Sharding：集群实例</li> <li>• ReplicaSet：副本集实例</li> <li>• Single：单节点实例</li> </ul>                                                                                                                     |
| batch_size         | Integer          | 批处理大小，单次函数执行处理的最大数据量。DIS、DDS、KAFKA、RABBITMQ触发器此参数必填。 <ul style="list-style-type: none"> <li>• DDS触发器：批处理大小设置1-10,000的范围内</li> <li>• DIS触发器：批处理大小设置1-10,000的范围内</li> <li>• KAFKA触发器：批处理大小设置1-1,000的范围内</li> <li>• RABBITMQ触发器：批处理大小设置1-1,000的范围内</li> </ul>     |
| queue_id           | String           | 队列id（DMS触发器参数）。                                                                                                                                                                                                                                                |
| consumer_group_id  | String           | 消费组id（DMS触发器参数）。                                                                                                                                                                                                                                               |
| polling_interval   | Integer          | 拉取周期。                                                                                                                                                                                                                                                          |
| stream_name        | String           | 通道名称（DIS触发器参数）。                                                                                                                                                                                                                                                |
| sharditerator_type | String           | 起始位置（DIS触发器参数）。 <ul style="list-style-type: none"> <li>• TRIM_HORIZON：从最早被存储至分区的有效记录开始读取。</li> <li>• LATEST：从分区中的最新记录开始读取，此设置可以保证总是读到分区中最新记录。</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• <b>TRIM_HORIZON</b></li> <li>• <b>LATEST</b></li> </ul> |
| polling_unit       | String           | 拉取周期单位（DIS触发器参数）。 <ul style="list-style-type: none"> <li>• s：秒</li> <li>• ms：毫秒</li> </ul> 枚举值： <ul style="list-style-type: none"> <li>• <b>s</b></li> <li>• <b>ms</b></li> </ul>                                                                              |

| 参数              | 参数类型             | 描述                                                                                                                                                                                                                                                                                                                                                              |
|-----------------|------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| max_fetch_bytes | Integer          | 最大提取字节数（DIS触发器参数）。<br>最小值： <b>0</b><br>最大值： <b>4194304</b>                                                                                                                                                                                                                                                                                                      |
| is_serial       | String           | 串行处理数据（DIS触发器参数），如果开启该选项，取一次数据处理完之后才会取下一次数据；否则只要拉取周期到了就会取数据进行处理。<br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>true</b></li> <li>• <b>false</b></li> </ul>                                                                                                                                                                                             |
| log_group_id    | String           | 日志组id（LTS触发器参数）。                                                                                                                                                                                                                                                                                                                                                |
| log_topic_id    | String           | 日志流id（LTS触发器参数）。                                                                                                                                                                                                                                                                                                                                                |
| bucket          | String           | 桶名称（OBS触发器参数），用作事件源的OBS存储桶，不能和本用户已有桶重名；不能和其他用户已有的桶重名；创建成功后不支持修改。<br>最小长度： <b>1</b><br>最大长度： <b>64</b>                                                                                                                                                                                                                                                           |
| prefix          | String           | 前缀（OBS触发器参数），输入一个可选性前缀来限制对以此关键字开头的对象的通知。<br>最小长度： <b>0</b><br>最大长度： <b>1024</b>                                                                                                                                                                                                                                                                                 |
| suffix          | String           | 后缀（OBS触发器参数），输入一个可选性后缀来限制对以此关键字结尾的对象的通知<br>最小长度： <b>0</b><br>最大长度： <b>1024</b>                                                                                                                                                                                                                                                                                  |
| events          | Array of strings | 触发事件（OBS触发器参数）。 <ul style="list-style-type: none"> <li>• ObjectCreated：表示所有创建对象的操作，包含Put、Post、Copy对象以及合并段</li> <li>• Put：使用Put方法上传对象</li> <li>• Post：使用Post方法上传对象</li> <li>• Copy：使用copy方法复制对象</li> <li>• CompleteMultipartUpload：表示合并分段任务</li> <li>• ObjectRemoved：表示删除对象</li> <li>• Delete：指定对象版本号删除对象</li> <li>• DeleteMarkerCreated：不指定对象版本号删除对象</li> </ul> |

| 参数                    | 参数类型             | 描述                                                                                                                            |
|-----------------------|------------------|-------------------------------------------------------------------------------------------------------------------------------|
| topic_urn             | String           | 主题URN（SMN触发器参数）。<br>最小长度： <b>1</b><br>最大长度： <b>255</b>                                                                        |
| topic_ids             | Array of strings | KAFKA主题id列表（KAFKA触发器参数）。                                                                                                      |
| kafka_user            | String           | KAFKA账户名（KAFKA触发器参数）。                                                                                                         |
| kafka_password        | String           | KAFKA账户密码（KAFKA触发器参数）。                                                                                                        |
| kafka_connect_address | String           | KAFKA实例连接IP地址（KAFKA触发器参数）。                                                                                                    |
| kafka_ssl_enable      | Boolean          | KAFKA连接是否开启安全认证（KAFKA触发器参数）。                                                                                                  |
| access_password       | String           | RABBITMQ账户密码（RABBITMQ触发器参数）。                                                                                                  |
| access_user           | String           | RABBITMQ账户名（RABBITMQ触发器参数）。                                                                                                   |
| connect_address       | String           | 实例连接IP地址（RABBITMQ触发器参数）。                                                                                                      |
| exchange_name         | String           | 交换机名称（RABBITMQ触发器参数）。                                                                                                         |
| vhost                 | String           | 虚拟机名称（RABBITMQ触发器参数）。                                                                                                         |
| ssl_enable            | Boolean          | RABBITMQ连接是否开启安全认证（RABBITMQ触发器参数）。                                                                                            |
| Key_encode            | Boolean          | EG obs触发器是否对对象加密（EVENTGRID触发器参数）。                                                                                             |
| agency                | String           | 使用的代理（EVENTGRID触发器参数）。                                                                                                        |
| channel_name          | String           | 通道名称（EVENTGRID触发器参数）。                                                                                                         |
| channel_id            | String           | 通道id（EVENTGRID触发器参数）。                                                                                                         |
| source_name           | String           | 事件源名称（EVENTGRID触发器参数）。                                                                                                        |
| created_time          | String           | 创建时间（EVENTGRID触发器参数）。                                                                                                         |
| status                | String           | 触发器状态（EVENTGRID触发器参数）。<br>枚举值：<br><ul style="list-style-type: none"> <li>● <b>ACTIVE</b></li> <li>● <b>DISABLE</b></li> </ul> |
| trigger_name          | String           | 触发器名称（EVENTGRID触发器参数）。                                                                                                        |

| 参数          | 参数类型             | 描述                                                  |
|-------------|------------------|-----------------------------------------------------|
| event_types | Array of strings | 事件类型（EVENTGRID触发器参数）。例如：OBS:DWR:ObjectCreated:COPY。 |

表 7-629 ApigTriggerFuncInfo

| 参数              | 参数类型    | 描述                                                                                                                                                                                   |
|-----------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| function_urn    | String  | 函数的URN，详细解释见FunctionGraph函数模型的描述。                                                                                                                                                    |
| invocation_type | String  | 调用函数执行方式。 <ul style="list-style-type: none"><li>• sync：同步执行</li><li>• async：异步执行</li></ul> 枚举值： <ul style="list-style-type: none"><li>• <b>sync</b></li><li>• <b>async</b></li></ul> |
| timeout         | Integer | API网关请求函数服务的超时时间（毫秒）。APIG触发器此参数必填。<br>最小值：1<br>最大值：60000                                                                                                                             |
| version         | String  | 函数版本信息。                                                                                                                                                                              |

状态码：404

表 7-630 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

更新触发器状态为开启。

```
PUT https://{Endpoint}/v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}
{
 "trigger_status": "ACTIVE"
}
```

## 响应示例

**状态码： 200**

OK

```
{
 "trigger_id" : "9a14fae1-78cf-4185-ac7a-429eb6dc41fb",
 "trigger_type_code" : "TIMER",
 "trigger_status" : "ACTIVE",
 "event_data" : {
 "name" : "Timer-cpg3",
 "schedule" : "3m",
 "schedule_type" : "Rate"
 },
 "last_updated_time" : "2022-11-09 16:37:24",
 "created_time" : "2019-10-29 17:15:53"
}
```

**状态码： 404**

Not Found

```
{
 "error_code" : "FSS.0404",
 "error_msg" : "not found"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

更新触发器状态为开启。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateTriggerSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 }
}
```



```
UpdateTriggerRequest request = new UpdateTriggerRequest();
UpdateTriggerRequestBody body = new UpdateTriggerRequestBody();
body.withTriggerStatus(UpdateTriggerRequestBody.TriggerStatusEnum.fromValue("ACTIVE"));
request.withBody(body);
try {
 UpdateTriggerResponse response = client.updateTrigger(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

更新触发器状态为开启。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = UpdateTriggerRequest()
 request.body = UpdateTriggerRequestBody(
 trigger_status="ACTIVE"
)
 response = client.update_trigger(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

更新触发器状态为开启。

```
package main

import (
```

```
"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.UpdateTriggerRequest{}
 triggerStatusUpdateTriggerRequestBody:=
 model.GetUpdateTriggerRequestBodyTriggerStatusEnum().ACTIVE
 request.Body = &model.UpdateTriggerRequestBody{
 TriggerStatus: &triggerStatusUpdateTriggerRequestBody,
 }
 response, err := client.UpdateTrigger(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述        |
|-----|-----------|
| 200 | OK        |
| 404 | Not Found |

## 错误码

请参见[错误码](#)。

## 7.14 函数流

## 7.14.1 同步执行工作流

### 功能介绍

以同步执行方式启动工作流（仅快速模式工作流支持），暂不支持国际站局点调用

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/fgs/workflows/{workflow\_id}/sync-executions

表 7-631 路径参数

| 参数          | 是否必选 | 参数类型   | 描述                                                         |
|-------------|------|--------|------------------------------------------------------------|
| project_id  | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |
| workflow_id | 是    | String | 工作流定义ID<br>最小长度：1<br>最大长度：512                              |

### 请求参数

表 7-632 请求 Body 参数

| 参数      | 是否必选 | 参数类型   | 描述                                |
|---------|------|--------|-----------------------------------|
| headers | 否    | Object | 执行时需要的Header                      |
| input   | 是    | Object | 定义执行时的入参，支持使用JSONPATH进行映射，以及指定默认值 |

### 响应参数

状态码：200

表 7-633 响应 Body 参数

| 参数           | 参数类型                                          | 描述                                                                                                                                      |
|--------------|-----------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------|
| execution_id | String                                        | 流程实例ID<br>最小长度：1<br>最大长度：64                                                                                                             |
| status       | String                                        | 流程执行最终状态<br>最小长度：1<br>最大长度：64<br>枚举值：<br><ul style="list-style-type: none"> <li>• success</li> <li>• fail</li> <li>• timeout</li> </ul> |
| output       | Object                                        | 工作流的执行结果，JSON格式，仅在status为success时有值                                                                                                     |
| errors       | Array of SyncExecutionNodeErrorDetail objects | 流程执行错误信息，仅在status为fail时有值<br>数组长度：0 - 200                                                                                               |
| begin_time   | String                                        | 流程实例创建时间，格式：yyyy-MM-ddTHH:mm:ssZ，UTC时间<br>最小长度：0<br>最大长度：64                                                                             |
| end_time     | String                                        | 流程实例结束时间，格式：yyyy-MM-ddTHH:mm:ssZ，UTC时间<br>最小长度：0<br>最大长度：64                                                                             |

表 7-634 SyncExecutionNodeErrorDetail

| 参数            | 参数类型   | 描述                            |
|---------------|--------|-------------------------------|
| node_id       | String | 流程节点ID<br>最小长度：1<br>最大长度：64   |
| error_message | String | 错误详细信息<br>最小长度：1<br>最大长度：2048 |

| 参数         | 参数类型   | 描述                                                          |
|------------|--------|-------------------------------------------------------------|
| begin_time | String | 流程实例创建时间，格式：yyyy-MM-ddTHH:mm:ssZ，UTC时间<br>最小长度：0<br>最大长度：64 |
| end_time   | String | 流程实例结束时间，格式：yyyy-MM-ddTHH:mm:ssZ，UTC时间<br>最小长度：0<br>最大长度：64 |

状态码：400

表 7-635 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |

状态码：404

表 7-636 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |

状态码：500

表 7-637 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |

## 请求示例

同步执行工作流。

```
POST /{project_id}/fgs/workflows/{workflow_id}/sync-executions
```

```
{
 "headers": {
 "tenantId": "user1"
 },
 "input": {
 "arg1": "argument_1",
 "arg2": "argument_2"
 }
}
```

## 响应示例

状态码: 200

Success

```
{
 "execution_id": "48aff955-ea10-4ffd-b426-190bc192dc14",
 "status": "success",
 "output": {
 "result": 200,
 "message": "success"
 },
 "errors": [{
 "node_id": "48aff955-ea10-4ffd-b426-190bc192dc14",
 "error_message": "Exception happened",
 "begin_time": "2021-01-01T00:00:00Z",
 "end_time": "2021-01-01T00:00:00Z"
 }],
 "begin_time": "2021-01-01T00:00:00Z",
 "end_time": "2021-01-01T00:00:00Z"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

同步执行工作流。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class StartSyncWorkflowExecutionSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 StartSyncWorkflowExecutionRequest request = new StartSyncWorkflowExecutionRequest();
 FlowExecuteBody body = new FlowExecuteBody();
 body.withInput("{\"arg2\":\"argument_2\",\"arg1\":\"argument_1\"}");
 body.withHeaders("{\"tenantId\":\"user1\"}");
 request.withBody(body);
 try {
 StartSyncWorkflowExecutionResponse response = client.startSyncWorkflowExecution(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

同步执行工作流。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
```

example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = StartSyncWorkflowExecutionRequest()
 request.body = FlowExecuteBody(
 input="{\"arg2\": \"argument_2\", \"arg1\": \"argument_1\"}",
 headers="{\"tenantId\": \"user1\"}"
)
 response = client.start_sync_workflow_execution(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

同步执行工作流。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.StartSyncWorkflowExecutionRequest{}
 var inputFlowExecuteBody interface{} = "{\"arg2\": \"argument_2\", \"arg1\": \"argument_1\"}"
 var headersFlowExecuteBody interface{} = "{\"tenantId\": \"user1\"}"
 request.Body = &model.FlowExecuteBody{
 Input: &inputFlowExecuteBody,
 Headers: &headersFlowExecuteBody,
 }
 response, err := client.StartSyncWorkflowExecution(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 }
}
```



```
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | Success               |
| 400 | Bad Request           |
| 404 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.2 异步执行工作流

### 功能介绍

以异步执行方式启动工作流，暂不支持国际站局点调用

### 调用方法

请参见[如何调用API](#)。

## URI

POST /v2/{project\_id}/fgs/workflows/{workflow\_id}/executions

表 7-638 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                                         |
|------------|------|--------|------------------------------------------------------------|
| project_id | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |

| 参数          | 是否必选 | 参数类型   | 描述                            |
|-------------|------|--------|-------------------------------|
| workflow_id | 是    | String | 工作流定义ID<br>最小长度：1<br>最大长度：512 |

## 请求参数

表 7-639 请求 Header 参数

| 参数                              | 是否必选 | 参数类型   | 描述                                                                                      |
|---------------------------------|------|--------|-----------------------------------------------------------------------------------------|
| X-Create-Time                   | 否    | String | workflowRun task create time                                                            |
| X-WorkflowRun-ID                | 否    | String | workflowRun id                                                                          |
| X-WorkflowRun-MergeFnParameters | 否    | String | Combines the output of the previous node with the input of the next node into an input. |

表 7-640 请求 Body 参数

| 参数      | 是否必选 | 参数类型   | 描述                                |
|---------|------|--------|-----------------------------------|
| headers | 否    | Object | 执行时需要的Header                      |
| input   | 是    | Object | 定义执行时的入参，支持使用JSONPATH进行映射，以及指定默认值 |

## 响应参数

状态码：200

表 7-641 响应 Body 参数

| 参数           | 参数类型   | 描述                       |
|--------------|--------|--------------------------|
| execution_id | String | 错误码<br>最小长度：1<br>最大长度：64 |

**状态码：400**

**表 7-642** 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |

**状态码：404**

**表 7-643** 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |

**状态码：500**

**表 7-644** 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |

## 请求示例

开始执行 workflow。

```
POST /v2/{project_id}/fgs/workflows/{workflow_id}/executions

{
 "headers": {
 "tenantId": "user1"
 },
 "input": {
 "arg1": "argument_1",
 "arg2": "argument_2"
 }
}
```

## 响应示例

**状态码： 200**

Success

```
{
 "execution_id": "48aff955-ea10-4ffd-b426-190bc192dc14"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

开始执行工作流。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class StartWorkflowExecutionSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 StartWorkflowExecutionRequest request = new StartWorkflowExecutionRequest();
 FlowExecuteBody body = new FlowExecuteBody();
 body.withInput("{\"arg2\":\"argument_2\",\"arg1\":\"argument_1\"}");
 body.withHeaders("{\"tenantId\":\"user1\"}");
 request.withBody(body);
 }
}
```

```
try {
 StartWorkflowExecutionResponse response = client.startWorkflowExecution(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

开始执行工作流。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = StartWorkflowExecutionRequest()
 request.body = FlowExecuteBody(
 input="{\"arg2\": \"argument_2\", \"arg1\": \"argument_1\"}",
 headers="{\"tenantId\": \"user1\"}"
)
 response = client.start_workflow_execution(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

开始执行工作流。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
```

```

"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.StartWorkflowExecutionRequest{}
 var inputFlowExecuteBody interface{} = "{\"arg2\":\"argument_2\",\"arg1\":\"argument_1\"}"
 var headersFlowExecuteBody interface{} = "{\"tenantId\":\"user1\"}"
 request.Body = &model.FlowExecuteBody{
 Input: &inputFlowExecuteBody,
 Headers: &headersFlowExecuteBody,
 }
 response, err := client.StartWorkflowExecution(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | Success               |
| 400 | Bad Request           |
| 404 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.3 删除函数流

### 功能介绍

删除函数流

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/fgs/workflows

表 7-645 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                                         |
|------------|------|--------|------------------------------------------------------------|
| project_id | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |

### 请求参数

表 7-646 请求 Body 参数

| 参数            | 是否必选 | 参数类型             | 描述                       |
|---------------|------|------------------|--------------------------|
| workflow_urns | 是    | Array of strings | 函数流URN列表<br>数组长度：0 - 200 |

### 响应参数

状态码：200

表 7-647 响应 Body 参数

| 参数      | 参数类型                                               | 描述                        |
|---------|----------------------------------------------------|---------------------------|
| success | Array of strings                                   | 成功流程URN列表<br>数组长度：0 - 100 |
| fail    | Array of <a href="#">OperateError Info</a> objects | 错误流程详情<br>数组长度：0 - 200    |

表 7-648 OperateErrorInfo

| 参数           | 参数类型   | 描述                                    |
|--------------|--------|---------------------------------------|
| workflow_urn | String | 唯一标识ID, 流程URN<br>最小长度: 1<br>最大长度: 128 |
| error_detail | String | 错误详情<br>最小长度: 1<br>最大长度: 64           |

状态码: 400

表 7-649 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

状态码: 500

表 7-650 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |



| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| request_id | String | 请求ID<br>最小长度： 2<br>最大长度： 512 |

## 请求示例

删除urn为urn:fss:  
{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow1:latest和  
urn:fss:  
{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow2:latest的函  
数流。

```
DELETE /v2/{project_id}/fgs/workflows
```

```
{
 "workflow_urns" : ["urn:fss:{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow1:latest",
 "urn:fss:{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow2:latest"]
}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

删除urn为urn:fss:  
{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow1:latest和  
urn:fss:  
{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow2:latest的函  
数流。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class BatchDeleteWorkflowsSolution {
 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 }
}
```

```
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
BatchDeleteWorkflowsRequest request = new BatchDeleteWorkflowsRequest();
WorkflowDeleteBody body = new WorkflowDeleteBody();
List<String> listbodyWorkflowUrns = new ArrayList<>();
listbodyWorkflowUrns.add("urn:fss:
{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow1:latest");
listbodyWorkflowUrns.add("urn:fss:
{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow2:latest");
body.withWorkflowUrns(listbodyWorkflowUrns);
request.withBody(body);
try {
 BatchDeleteWorkflowsResponse response = client.batchDeleteWorkflows(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

删除urn为urn:fss:  
{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow1:latest和  
urn:fss:  
{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow2:latest的函  
数流。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
```

```
.build()

try:
 request = BatchDeleteWorkflowsRequest()
 listWorkflowUrnsbody = [
 "urn:fss:{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow1:latest",
 "urn:fss:{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow2:latest"
]
 request.body = WorkflowDeleteBody(
 workflow_urns=listWorkflowUrnsbody
)
 response = client.batch_delete_workflows(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

删除urn为urn:fss:  
{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow1:latest和  
urn:fss:  
{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow2:latest的函  
数流。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.BatchDeleteWorkflowsRequest{}
 var listWorkflowUrnsbody = []string{
 "urn:fss:{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow1:latest",
 "urn:fss:{region}:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow2:latest",
 }
 request.Body = &model.WorkflowDeleteBody{
 WorkflowUrns: listWorkflowUrnsbody,
 }
 response, err := client.BatchDeleteWorkflows(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 }
}
```

```
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | Success               |
| 400 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.4 查询函数流

### 功能介绍

查询函数流

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/workflows

表 7-651 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                                         |
|------------|------|--------|------------------------------------------------------------|
| project_id | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |

表 7-652 Query 参数

| 参数                 | 是否必选 | 参数类型    | 描述                                                      |
|--------------------|------|---------|---------------------------------------------------------|
| workflow_name      | 否    | String  | 函数流名称<br>最小长度: 0<br>最大长度: 64                            |
| limit              | 否    | Integer | 分页查询, 每页显示的条目数量, 默认值为200 limit大于200时, 按照200处理<br>最小值: 1 |
| offset             | 否    | Integer | 分页查询, 分页的偏移量, 默认值为0 offset小于0时, 按照0处理<br>最小值: 0         |
| enterprise_project | 否    | String  | 企业项目ID                                                  |
| mode               | 否    | String  | 函数流模式 "NORMAL"标准函数流 "EXPRESS"快速函数流                      |

## 请求参数

无

## 响应参数

状态码: 200

表 7-653 响应 Body 参数

| 参数        | 参数类型                                                | 描述                                      |
|-----------|-----------------------------------------------------|-----------------------------------------|
| total     | Long                                                | 返回所有满足条件的对象个数<br>最小值: 0<br>最大值: 1000000 |
| size      | Integer                                             | 返回对象的大小<br>最小值: 0<br>最大值: 200           |
| workflows | Array of <a href="#">WorkflowSimpleInfo</a> objects | 返回的实体对象<br>数组长度: 0 - 200                |

表 7-654 WorkflowSimpleInfo

| 参数                     | 参数类型    | 描述                                                                                                                                          |
|------------------------|---------|---------------------------------------------------------------------------------------------------------------------------------------------|
| id                     | String  | 唯一标识ID, 流程定义ID<br>最小长度: 1<br>最大长度: 64                                                                                                       |
| workflow_urn           | String  | 函数工作流URN, 格式为:<br>urn:fss:<region_id>:<project_id>:workflow:<package>:<workflow_name>:<version>注意:<br>package当前只支持defaultversion当前只支持latest |
| name                   | String  | 流程定义名称<br>最小长度: 1<br>最大长度: 64                                                                                                               |
| description            | String  | 流程定义描述<br>最小长度: 0<br>最大长度: 512                                                                                                              |
| created_time           | String  | 流程创建时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC时间<br>最小长度: 0<br>最大长度: 64                                                                              |
| updated_time           | String  | 流程修改时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC时间<br>最小长度: 0<br>最大长度: 64                                                                              |
| created_by             | String  | 流程创建者<br>最小长度: 0<br>最大长度: 64                                                                                                                |
| enable_stream_response | Boolean | 是否返回流数据<br>缺省值: <b>false</b>                                                                                                                |

状态码: 400

表 7-655 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36 |

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

状态码: 500

表 7-656 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

## 请求示例

查询函数流列表。

```
GET /v2/{project_id}/fgs/workflows
```

## 响应示例

状态码: 200

函数流查询返回体

```
{
 "total": 1,
 "size": 1,
 "workflows": [{
 "id": "48aff955-ea10-4ffd-b426-190bc192dc14",
 "workflow_urn": "urn:fss:cn-north-7:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow:latest",
 "name": "test-flow",
 "description": "desc",
 "created_time": "2021-01-01T00:00:00Z",
 "updated_time": "2021-01-01T00:00:00Z",
 "created_by": "admin"
 }]
}
```

```
 }
 }
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListWorkflowSolution {
 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListWorkflowRequest request = new ListWorkflowRequest();
 request.withWorkflowName("<workflow_name>");
 request.withLimit(<limit>);
 request.withOffset(<offset>);
 request.withEnterpriseProject("<enterprise_project>");
 request.withMode("<mode>");
 try {
 ListWorkflowResponse response = client.listWorkflow(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8
```



```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListWorkflowRequest()
 request.workflow_name = "<workflow_name>"
 request.limit = <limit>
 request.offset = <offset>
 request.enterprise_project = "<enterprise_project>"
 request.mode = "<mode>"
 response = client.list_workflow(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListWorkflowRequest{}
```

```
workflowNameRequest:= "<workflow_name>"
request.WorkflowName = &workflowNameRequest
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
offsetRequest:= int32(<offset>)
request.Offset = &offsetRequest
enterpriseProjectRequest:= "<enterprise_project>"
request.EnterpriseProject = &enterpriseProjectRequest
modeRequest:= "<mode>"
request.Mode = &modeRequest
response, err := client.ListWorkflow(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | 函数流查询返回体              |
| 400 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.5 创建函数流

### 功能介绍

创建函数流

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/fgs/workflows

表 7-657 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                                         |
|------------|------|--------|------------------------------------------------------------|
| project_id | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |

## 请求参数

表 7-658 请求 Body 参数

| 参数          | 是否必选 | 参数类型                                            | 描述                                                                                       |
|-------------|------|-------------------------------------------------|------------------------------------------------------------------------------------------|
| name        | 是    | String                                          | 函数流名称<br>最小长度：1<br>最大长度：64                                                               |
| description | 否    | String                                          | 函数流描述<br>最小长度：1<br>最大长度：512                                                              |
| start       | 是    | String                                          | 流程开始节点ID<br>最小长度：1<br>最大长度：256                                                           |
| triggers    | 否    | Array of <a href="#">Trigger</a> objects        | 触发器列表<br>数组长度：0 - 200                                                                    |
| functions   | 是    | Array of <a href="#">Function</a> objects       | 函数列表<br>数组长度：0 - 100                                                                     |
| states      | 是    | Array of <a href="#">OperationState</a> objects | 函数流节点清单，定义参考 <a href="#">SleepState</a> 和 <a href="#">OperationState</a><br>数组长度：1 - 100 |
| constants   | 是    | Object                                          | 函数流中的常量                                                                                  |
| retries     | 是    | Array of <a href="#">Retry</a> objects          | 重试策略清单<br>数组长度：0 - 200                                                                   |

| 参数                     | 是否必选 | 参数类型                        | 描述                                                                                                                                                                                                                                                                                                     |
|------------------------|------|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| mode                   | 否    | String                      | 函数流模式，当前支持两种模式<br>NORMAL: 标准模式，普通模式<br>面向普通的业务场景，支持长时间任务，支持执行历史持久化和查询，只支持异步调用<br>EXPRESS: 快速模式，快速模式<br>面向业务执行时长较短，需要极致性能的场景，只支持流程执行时长低于5分钟的场景，不支持执行历史持久化，支持同步和异步调用默认为标准模式<br>缺省值: <b>NORMAL</b><br>枚举值:<br><ul style="list-style-type: none"> <li>• <b>NORMAL</b></li> <li>• <b>EXPRESS</b></li> </ul> |
| express_config         | 否    | <b>ExpressConfig</b> object | 快速函数流模式配置，当mode为EXPRESS时需要设置                                                                                                                                                                                                                                                                           |
| enterprise_project_id  | 否    | String                      | 企业项目ID<br>最小长度: <b>0</b><br>最大长度: <b>64</b>                                                                                                                                                                                                                                                            |
| enable_stream_response | 否    | Boolean                     | 是否返回流数据<br>缺省值: <b>false</b>                                                                                                                                                                                                                                                                           |

表 7-659 Trigger

| 参数           | 是否必选 | 参数类型   | 描述                                          |
|--------------|------|--------|---------------------------------------------|
| trigger_name | 是    | String | 触发器名称<br>最小长度: <b>1</b><br>最大长度: <b>256</b> |

| 参数             | 是否必选 | 参数类型                           | 描述                                                                                                                                                                                                                                                                                   |
|----------------|------|--------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| trigger_type   | 是    | String                         | 触发器类型 FLOWTIMER: 定时触发器 SMN: SMN触发器<br>APIG: APIG触发器(共享版)<br>APIG_DE: APIG触发器(专享版)<br>OBS: OBS触发器<br>(当前OBS触发器暂不支持中国站。)<br>最小长度: 1<br>最大长度: 32<br>枚举值: <ul style="list-style-type: none"> <li>• FLOWTIMER</li> <li>• SMN</li> <li>• APIG</li> <li>• APIG_DE</li> <li>• OBS</li> </ul> |
| enabled        | 否    | Boolean                        | 是否启用触发器<br>缺省值: <b>true</b><br>枚举值: <ul style="list-style-type: none"> <li>• <b>true</b></li> <li>• <b>false</b></li> </ul>                                                                                                                                                          |
| trigger_config | 否    | <b>OBSTriggerConfig</b> object | OBS触发器配置, 当 trigger_type为OBS时需要设置                                                                                                                                                                                                                                                    |

表 7-660 OBSTriggerConfig

| 参数     | 是否必选 | 参数类型             | 描述                                                    |
|--------|------|------------------|-------------------------------------------------------|
| bucket | 是    | String           | OBS桶名 ( trigger_type为OBS时配置)<br>最小长度: 1<br>最大长度: 64   |
| events | 是    | Array of strings | OBS事件列表 ( trigger_type为OBS时配置)<br>数组长度: 1 - 64        |
| prefix | 是    | String           | 对象名前缀 ( trigger_type为OBS时配置)<br>最小长度: 1<br>最大长度: 1024 |

| 参数     | 是否必选 | 参数类型   | 描述                                                |
|--------|------|--------|---------------------------------------------------|
| suffix | 是    | String | 对象名后缀（trigger_type为OBS时配置）<br>最小长度：1<br>最大长度：1024 |

表 7-661 Function

| 参数        | 是否必选 | 参数类型   | 描述                                       |
|-----------|------|--------|------------------------------------------|
| name      | 是    | String | 函数名称，在单个流程中，名称需要唯一<br>最小长度：0<br>最大长度：256 |
| operation | 是    | String | 函数调用URN<br>最小长度：0<br>最大长度：256            |
| metadata  | 否    | Object | 函数扩展属性，由用户自己定制                           |

表 7-662 OperationState

| 参数          | 是否必选 | 参数类型                     | 描述                                                                                                                                             |
|-------------|------|--------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| action_mode | 否    | String                   | Action执行模式，支持串行，并行两种模式，默认串行<br>最小长度：1<br>最大长度：32<br>枚举值：<br><ul style="list-style-type: none"> <li>• sequential</li> <li>• parallel</li> </ul> |
| actions     | 是    | Array of Action objects  | 节点中要执行的操作列表<br>数组长度：1 - 10                                                                                                                     |
| on_errors   | 否    | Array of OnError objects | 错误处理策略<br>数组长度：0 - 200                                                                                                                         |
| id          | 是    | String                   | 节点ID，需要在当前函数流中唯一<br>最小长度：0<br>最大长度：64                                                                                                          |

| 参数                | 是否必选 | 参数类型                          | 描述                                                                                                                                 |
|-------------------|------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| name              | 是    | String                        | 节点名称<br>最小长度：1<br>最大长度：256                                                                                                         |
| type              | 是    | String                        | 节点类型<br>最小长度：1<br>最大长度：32<br>枚举值：<br><ul style="list-style-type: none"> <li>• Operation</li> <li>• Sleep</li> <li>• End</li> </ul> |
| end               | 是    | Boolean                       | 是否是结束节点                                                                                                                            |
| transition        | 是    | String                        | 下一步骤节点ID<br>最小长度：0<br>最大长度：256                                                                                                     |
| state_data_filter | 否    | <b>StateDataFilter</b> object | 过滤表达式                                                                                                                              |
| duration          | 否    | Long                          | 时间等待节点等待时间（秒），节点类型为Sleep时为必填，节点类型不为Sleep时无效<br>最小值：0.001<br>最大值：86400                                                              |

表 7-663 Action

| 参数           | 是否必选 | 参数类型                      | 描述     |
|--------------|------|---------------------------|--------|
| function_ref | 否    | <b>FunctionRef</b> object | 函数调用信息 |

表 7-664 FunctionRef

| 参数       | 是否必选 | 参数类型   | 描述                                                      |
|----------|------|--------|---------------------------------------------------------|
| ref_name | 是    | String | 函数引用名称，需要和外层 functions 中的 name 对应<br>最小长度：0<br>最大长度：256 |

| 参数          | 是否必选 | 参数类型   | 描述                                                                                                                                                                                  |
|-------------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| invoke_mode | 否    | String | 函数调用模式，目前只支持同步调用<br>最小长度：1<br>最大长度：32<br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>synchronize</b></li> <li>• <b>asynchronous</b></li> </ul>                              |
| arguments   | 是    | Object | 函数执行时的入参，支持引用 constants 中的常量 定义方式：<br>参数路径   常量值/常量路径 参数路径指输入参数的 JsonPath 路径，如 \$.a.b[0].c 常量值可以为数字类型，字符串类型(需要用单引号括起来)，布尔类型 常量路径为常量的 JsonPath 路径，但是根节点需要用 \$CONST 表示，示例：\$CONST.a.b |

表 7-665 OnError

| 参数         | 是否必选 | 参数类型   | 描述                                        |
|------------|------|--------|-------------------------------------------|
| error      | 否    | String | 错误匹配表达式，用来过滤需要处理的异常<br>最小长度：0<br>最大长度：256 |
| transition | 否    | String | 下一步骤节点ID<br>最小长度：0<br>最大长度：256            |
| retry_ref  | 否    | String | 重试策略名称<br>最小长度：0<br>最大长度：256              |

表 7-666 StateDataFilter

| 参数    | 是否必选 | 参数类型   | 描述                                         |
|-------|------|--------|--------------------------------------------|
| input | 否    | String | 输入过滤表达式 ( JsonPath )<br>最小长度：0<br>最大长度：256 |



| 参数     | 是否必选 | 参数类型   | 描述                                                       |
|--------|------|--------|----------------------------------------------------------|
| output | 否    | String | 输出过滤表达式 (JsonPath)<br>最小长度: <b>0</b><br>最大长度: <b>256</b> |

表 7-667 Retry

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| name         | 是    | String | 重试策略名称, 在单个流程中, 名称需要唯一<br>最小长度: <b>1</b><br>最大长度: <b>256</b> |
| delay        | 否    | Long   | 重试间隔, 单位: 秒。若不传, 默认为1<br>最小值: <b>1</b><br>最大值: <b>30</b>     |
| max_attempts | 否    | Long   | 最大重试次数, 。若不传, 默认为3<br>最小值: <b>1</b><br>最大值: <b>8</b>         |

表 7-668 ExpressConfig

| 参数        | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                 |
|-----------|------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| log_level | 否    | String | 快速模式相关配置, 仅在mode配置为EXPRESS时生效 快速模式下流程的执行日志级别, 当前支持: ALL: 记录所有节点的执行日志 ERROR: 仅记录异常节点执行日志 NONE: 不记录日志 注意: 当配置为ALL和ERROR级别时租户需要开启LTS相关权限<br>最小长度: <b>1</b><br>最大长度: <b>5</b><br>枚举值:<br><ul style="list-style-type: none"> <li>● <b>ALL</b></li> <li>● <b>ERROR</b></li> <li>● <b>NONE</b></li> </ul> |

## 响应参数

状态码： 201

表 7-669 响应 Body 参数

| 参数                     | 参数类型    | 描述                                                                                                                                                     |
|------------------------|---------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| id                     | String  | 唯一标识ID, 流程定义ID<br>最小长度: 1<br>最大长度: 64                                                                                                                  |
| workflow_urn           | String  | 函数 workflow URN, 格式为:<br>urn:fss:<region_id>:<project_id>:workflow:<package>:<workflow_name>:<version>注意:<br>package 当前只支持 defaultversion 当前只支持 latest |
| name                   | String  | 流程定义名称<br>最小长度: 1<br>最大长度: 64                                                                                                                          |
| description            | String  | 流程定义描述<br>最小长度: 0<br>最大长度: 512                                                                                                                         |
| created_time           | String  | 流程创建时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC 时间<br>最小长度: 0<br>最大长度: 64                                                                                        |
| updated_time           | String  | 流程修改时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC 时间<br>最小长度: 0<br>最大长度: 64                                                                                        |
| created_by             | String  | 流程创建者<br>最小长度: 0<br>最大长度: 64                                                                                                                           |
| enable_stream_response | Boolean | 是否返回流数据<br>缺省值: <b>false</b>                                                                                                                           |

状态码： 400

表 7-670 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |
| request_id | String | 请求ID<br>最小长度：2<br>最大长度：512 |

状态码：500

表 7-671 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |
| request_id | String | 请求ID<br>最小长度：2<br>最大长度：512 |

## 请求示例

创建名为workflow\_test，开始节点的名称为test的函数流。

POST /v2/{project\_id}/fgs/workflows

```
{
 "name": "workflow_test",
 "start": "test",
 "functions": [{
 "name": "test",
 "operation": "urn:fss:{region}:c53626012ba84727b938ca8bf03108ef:test:latest"
 }],
 "constants": { },
 "states": [{
 "id": "test",
 "name": "test",
```

```
"type" : "Operation",
"end" : false,
"transition" : "end",
"actions" : [{
 "function_ref" : {
 "ref_name" : "test",
 "arguments" : {
 "execution_name" : "$.test"
 }
 }
}
], {
 "id" : "end",
 "name" : "end",
 "type" : "End",
 "end" : true,
 "transition" : "",
 "state_data_filter" : {
 "input" : "",
 "output" : ""
 },
 "actions" : [],
 "action_mode" : ""
}],
"retries" : [{
 "name" : "test"
}]
}
```

## 响应示例

状态码： 201

Created

```
{
 "id" : "48aff955-ea10-4ffd-b426-190bc192dc14",
 "workflow_urn" : "urn:fss:cn-north-7:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow:latest",
 "name" : "test-flow",
 "description" : "desc",
 "created_time" : "2021-01-01T00:00:00Z",
 "updated_time" : "2021-01-01T00:00:00Z",
 "created_by" : "admin"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建名为workflow\_test，开始节点的名称为test的函数流。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.List;
import java.util.ArrayList;
```

```
public class CreateWorkflowSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateWorkflowRequest request = new CreateWorkflowRequest();
 WorkflowCreateBody body = new WorkflowCreateBody();
 List<Retry> listbodyRetries = new ArrayList<>();
 listbodyRetries.add(
 new Retry()
 .withName("test")
);
 StateDataFilter stateDataFilterStates = new StateDataFilter();
 stateDataFilterStates.withInput("")
 .withOutput("");
 FunctionRef functionRefActions = new FunctionRef();
 functionRefActions.withRefName("test")
 .withArguments("{\"execution_name\":\"$.test\"}");
 List<Action> listStatesActions = new ArrayList<>();
 listStatesActions.add(
 new Action()
 .withFunctionRef(functionRefActions)
);
 List<OperationState> listbodyStates = new ArrayList<>();
 listbodyStates.add(
 new OperationState()
 .withActions(listStatesActions)
 .withId("test")
 .withName("test")
 .withType(OperationState.TypeEnum.fromValue("Operation"))
 .withEnd(false)
 .withTransition("end")
);
 listbodyStates.add(
 new OperationState()
 .withActionMode(OperationState.ActionModeEnum.fromValue(""))
 .withActions()
 .withId("end")
 .withName("end")
 .withType(OperationState.TypeEnum.fromValue("End"))
 .withEnd(true)
 .withTransition("")
 .withStateDataFilter(stateDataFilterStates)
);
 List<Function> listbodyFunctions = new ArrayList<>();
 listbodyFunctions.add(
 new Function()
 .withName("test")
 .withOperation("urn:fss:{region}:c53626012ba84727b938ca8bf03108ef:test:latest")
);
 body.withRetries(listbodyRetries);
 body.withConstants(new Object());
 body.withStates(listbodyStates);
 body.withFunctions(listbodyFunctions);
 }
}
```

```
body.withStart("test");
body.withName("workflow_test");
request.withBody(body);
try {
 CreateWorkflowResponse response = client.createWorkflow(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

创建名为workflow\_test，开始节点的名称为test的函数流。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateWorkflowRequest()
 listRetriesbody = [
 Retry(
 name="test"
)
]
 stateDataFilterStates = StateDataFilter(
 input="",
 output=""
)
 functionRefActions = FunctionRef(
 ref_name="test",
 arguments="{\"execution_name\": \"$.test\"}"
)
 listActionsStates = [
 Action(
 function_ref=functionRefActions
)
]
 listStatesbody = [
 OperationState(
```

```
 actions=listActionsStates,
 id="test",
 name="test",
 type="Operation",
 end=False,
 transition="end"
),
 OperationState(
 action_mode="",
 id="end",
 name="end",
 type="End",
 end=True,
 transition="",
 state_data_filter=stateDataFilterStates
)
]
listFunctionsbody = [
 Function(
 name="test",
 operation="urn:fss:{region}:c53626012ba84727b938ca8bf03108ef:test:latest"
)
]
request.body = WorkflowCreateBody(
 retries=listRetriesbody,
 constants={},
 states=listStatesbody,
 functions=listFunctionsbody,
 start="test",
 name="workflow_test"
)
response = client.create_workflow(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

创建名为workflow\_test，开始节点的名称为test的函数流。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
```

```
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build()

request := &model.CreateWorkflowRequest{}
var listRetriesbody = []model.Retry{
 {
 Name: "test",
 },
}
inputStateDataFilter:= ""
outputStateDataFilter:= ""
stateDataFilterStates := &model.StateDataFilter{
 Input: &inputStateDataFilter,
 Output: &outputStateDataFilter,
}
var argumentsFunctionRef interface{} = "{}execution_name\\":\\"$.test\\"}"
functionRefActions := &model.FunctionRef{
 RefName: "test",
 Arguments: &argumentsFunctionRef,
}
var listActionsStates = []model.Action{
 {
 FunctionRef: functionRefActions,
 },
}
actionModeStates:= model.GetOperationStateActionModeEnum().EMPTY
var listStatesbody = []model.OperationState{
 {
 Actions: listActionsStates,
 Id: "test",
 Name: "test",
 Type: model.GetOperationStateTypeEnum().OPERATION,
 End: false,
 Transition: "end",
 },
 {
 ActionMode: &actionModeStates,
 Id: "end",
 Name: "end",
 Type: model.GetOperationStateTypeEnum().END,
 End: true,
 Transition: "",
 StateDataFilter: stateDataFilterStates,
 },
}
var listFunctionsbody = []model.Function{
 {
 Name: "test",
 Operation: "urn:fss:{regin}:c53626012ba84727b938ca8bf03108ef:test:latest",
 },
}
var constantsWorkflowCreateBody interface{} = make(map[string]string)
request.Body = &model.WorkflowCreateBody{
 Retries: listRetriesbody,
 Constants: &constantsWorkflowCreateBody,
 States: listStatesbody,
 Functions: listFunctionsbody,
 Start: "test",
 Name: "workflow_test",
}
response, err := client.CreateWorkflow(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```



## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 201 | Created               |
| 400 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.6 获取指定函数流执行实例列表

### 功能介绍

获取指定函数流执行实例列表

### 调用方法

请参见[如何调用API](#)。

## URI

GET /v2/{project\_id}/fgs/workflows/{workflow\_id}/executions

表 7-672 路径参数

| 参数          | 是否必选 | 参数类型   | 描述                                                         |
|-------------|------|--------|------------------------------------------------------------|
| project_id  | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |
| workflow_id | 是    | String | 函数工作流ID<br>最小长度：1<br>最大长度：512                              |

表 7-673 Query 参数

| 参数         | 是否必选 | 参数类型    | 描述                                                                                                                                                                              |
|------------|------|---------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| limit      | 否    | Integer | 分页查询，每页显示的条目数量，最大数量200，超过200后只返回200<br>最小值：0<br>最大值：500                                                                                                                         |
| status     | 否    | String  | 需要过滤的流程实例状态<br>最小长度：0<br>最大长度：64<br>枚举值：<br><ul style="list-style-type: none"> <li>• success</li> <li>• fail</li> <li>• running</li> <li>• timeout</li> <li>• cancel</li> </ul> |
| start_time | 否    | String  | 查询开始时间，UTC时间。若起始时间未填写，以终止时间前推3天为起始时间                                                                                                                                            |
| end_time   | 否    | String  | 查询结束时间，UTC时间。若终止时间未填写，以起始时间后退3天未终止时间。若均未填写，默认查询最近3天数据。                                                                                                                          |

### 请求参数

无

### 响应参数

状态码： 200

表 7-674 响应 Body 参数

| 参数         | 参数类型                                                | 描述                       |
|------------|-----------------------------------------------------|--------------------------|
| executions | Array of <a href="#">FlowExecutionBrief</a> objects | 流程执行信息列表<br>数组长度：0 - 999 |

表 7-675 FlowExecutionBrief

| 参数               | 参数类型   | 描述                                                                                                                                                                  |
|------------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| workflow_id      | String | 流程定义ID<br>最小长度：1<br>最大长度：64                                                                                                                                         |
| workflow_urn     | String | 函数工作流URN, 格式为：<br>urn:fss:<region_id>:<project_id>:workflow:<package>:<workflow_name>:<version>注意：<br>package当前只支持defaultversion当前只支持latest                         |
| execution_id     | String | 流程执行实例ID<br>最小长度：1<br>最大长度：64                                                                                                                                       |
| status           | String | 流程实例执行状态<br>最小长度：1<br>最大长度：32<br>枚举值： <ul style="list-style-type: none"><li>• success</li><li>• fail</li><li>• running</li><li>• timeout</li><li>• cancel</li></ul> |
| begin_time       | String | 流程实例创建时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC时间<br>最小长度：0<br>最大长度：64                                                                                                      |
| end_time         | String | 流程实例结束时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC时间<br>最小长度：0<br>最大长度：64                                                                                                      |
| last_update_time | String | 流程实例上次更新时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC时间<br>最小长度：0<br>最大长度：64                                                                                                    |
| created_by       | String | 流程实例创建者<br>最小长度：1<br>最大长度：32                                                                                                                                        |

状态码：400

表 7-676 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

状态码: 500

表 7-677 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

## 请求示例

获取函数流id为{workflow\_id}的函数流的执行实例列表。

```
GET /v2/{project_id}/fgs/workflows/{workflow_id}/executions
```

## 响应示例

状态码: 200

函数流执行查询返回体

```
{
 "executions": [{
 "workflow_id": "48aff955-ea10-4ffd-b426-190bc192dc14",
 "workflow_urn": "urn:fss:cn-north-7:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow:latest",
```

```
"execution_id" : "48aff955-ea10-4ffd-b426-190bc192dc14",
"status" : "running",
"begin_time" : "2021-01-01T00:00:00Z",
"end_time" : "2021-01-01T00:00:00Z",
"last_update_time" : "2021-01-01T00:00:00Z",
"created_by" : "admin"
}]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListWorkflowExecutionsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListWorkflowExecutionsRequest request = new ListWorkflowExecutionsRequest();
 request.withLimit(<limit>);
 request.withStatus(ListWorkflowExecutionsRequest.StatusEnum.fromValue("<status>"));
 request.withStartTime("<start_time>");
 request.withEndTime("<end_time>");
 try {
 ListWorkflowExecutionsResponse response = client.listWorkflowExecutions(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListWorkflowExecutionsRequest()
 request.limit = <limit>
 request.status = "<status>"
 request.start_time = "<start_time>"
 request.end_time = "<end_time>"
 response = client.list_workflow_executions(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
```

```
Build()

request := &model.ListWorkflowExecutionsRequest{}
limitRequest:= int32(<limit>)
request.Limit = &limitRequest
statusRequest:= model.GetListWorkflowExecutionsRequestStatusEnum().<STATUS>
request.Status = &statusRequest
startTimeRequest:= "<start_time>"
request.StartTime = &startTimeRequest
endTimeRequest:= "<end_time>"
request.EndTime = &endTimeRequest
response, err := client.ListWorkflowExecutions(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | 函数流执行查询返回体            |
| 400 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.7 获取指定函数流执行实例

### 功能介绍

获取指定函数流执行实例

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/workflows/{workflow\_id}/executions/{execution\_id}

表 7-678 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                                            |
|--------------|------|--------|---------------------------------------------------------------|
| project_id   | 是    | String | 项目ID, 获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度: 1<br>最大长度: 64 |
| workflow_id  | 是    | String | 函数工作流ID<br>最小长度: 1<br>最大长度: 512                               |
| execution_id | 是    | String | 函数流执行实例ID<br>最小长度: 1<br>最大长度: 64                              |

## 请求参数

表 7-679 请求 Header 参数

| 参数                               | 是否必选 | 参数类型    | 描述             |
|----------------------------------|------|---------|----------------|
| X-Get-Workflow-Full-History-Data | 否    | Boolean | 获取函数流执行详情完整输出值 |

## 响应参数

状态码: 200

表 7-680 响应 Body 参数

| 参数           | 参数类型   | 描述                                                                                                                                          |
|--------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------|
| workflow_id  | String | 流程定义ID<br>最小长度: 1<br>最大长度: 64                                                                                                               |
| workflow_urn | String | 函数工作流URN, 格式为:<br>urn:fss:<region_id>:<project_id>:workflow:<package>:<workflow_name>:<version>注意:<br>package当前只支持defaultversion当前只支持latest |



| 参数                     | 参数类型                                                 | 描述                                                                                                                                                                        |
|------------------------|------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| execution_id           | String                                               | 流程执行实例ID<br>最小长度：1<br>最大长度：64                                                                                                                                             |
| status                 | String                                               | 流程实例执行状态<br>最小长度：1<br>最大长度：32<br>枚举值： <ul style="list-style-type: none"> <li>• success</li> <li>• fail</li> <li>• running</li> <li>• timeout</li> <li>• cancel</li> </ul> |
| headers                | Object                                               | 函数执行时需要的Header                                                                                                                                                            |
| input                  | Object                                               | 函数执行时的入参                                                                                                                                                                  |
| output                 | Object                                               | 函数执行结果                                                                                                                                                                    |
| begin_time             | String                                               | 流程实例创建时间，格式：yyyy-MM-ddTHH:mm:ssZ，UTC时间<br>最小长度：0<br>最大长度：64                                                                                                               |
| end_time               | String                                               | 流程实例结束时间，格式：yyyy-MM-ddTHH:mm:ssZ，UTC时间<br>最小长度：0<br>最大长度：64                                                                                                               |
| last_update_time       | String                                               | 流程实例上次更新时间，格式：yyyy-MM-ddTHH:mm:ssZ，UTC时间<br>最小长度：0<br>最大长度：64                                                                                                             |
| created_by             | String                                               | 流程实例创建者<br>最小长度：1<br>最大长度：32                                                                                                                                              |
| node_execution_details | Array of <a href="#">NodeExecutionDetail</a> objects | 节点执行信息<br>数组长度：0 - 200                                                                                                                                                    |

表 7-681 NodeExecutionDetail

| 参数           | 参数类型                                           | 描述                          |
|--------------|------------------------------------------------|-----------------------------|
| node_id      | String                                         | 流程节点ID<br>最小长度：1<br>最大长度：64 |
| node_name    | String                                         | 流程节点名称                      |
| execution_id | String                                         | 流程节点执行ID                    |
| executions   | Array of <a href="#">NodeExecution</a> objects | 节点执行记录<br>数组长度：0 - 200      |

表 7-682 NodeExecution

| 参数            | 参数类型   | 描述                                                                                                                                                                  |
|---------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| status        | String | 流程节点执行状态<br>最小长度：1<br>最大长度：32<br>枚举值： <ul style="list-style-type: none"><li>• success</li><li>• fail</li><li>• running</li><li>• timeout</li><li>• cancel</li></ul> |
| input         | Object | 函数执行时的入参                                                                                                                                                            |
| output        | Object | 函数执行结果                                                                                                                                                              |
| begin_time    | Long   | 节点启动时间，UTC毫秒时间戳格式<br>最小值：0<br>最大值：9999999999999999                                                                                                                  |
| end_time      | Long   | 节点结束时间，UTC毫秒时间戳格式<br>最小值：0<br>最大值：9999999999999999                                                                                                                  |
| error_message | Object | 节点错误信息，仅在节点出错时非空                                                                                                                                                    |
| request_id    | String | 流程节点请求ID                                                                                                                                                            |

状态码：400

表 7-683 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

状态码: 404

表 7-684 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |

状态码: 500

表 7-685 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| request_id | String | 请求ID<br>最小长度：2<br>最大长度：512 |

## 请求示例

获取函数流id为{workflow\_id}并且执行实例id为{execution\_id}的函数流执行实例详情。

```
GET /v2/{project_id}/fgs/workflows/{workflow_id}/executions/{execution_id}
```

## 响应示例

状态码：200

OK

```
{
 "workflow_id": "48aff955-ea10-4ffd-b426-190bc192dc14",
 "workflow_urn": "urn:fss:cn-north-7:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow:latest",
 "execution_id": "48aff955-ea10-4ffd-b426-190bc192dc14",
 "status": "running",
 "headers": {
 "tenantId": "user1"
 },
 "input": {
 "arg1": "argument_1",
 "arg2": "argument_2"
 },
 "output": {
 "arg1": "argument_1",
 "arg2": "argument_2"
 },
 "begin_time": "2021-01-01T00:00:00Z",
 "end_time": "2021-01-01T00:00:00Z",
 "last_update_time": "2021-01-01T00:00:00Z",
 "created_by": "admin",
 "node_execution_details": [{
 "node_id": "48aff955-ea10-4ffd-b426-190bc192dc14",
 "node_name": "string",
 "execution_id": "string",
 "executions": [{
 "status": "running",
 "input": {
 "arg1": "argument_1",
 "arg2": "argument_2"
 },
 "output": {
 "arg1": "argument_1",
 "arg2": "argument_2"
 },
 "begin_time": 1605119892050,
 "end_time": 1605119892050,
 "error_message": {
 "detail:"Execution failed\"": null
 },
 "request_id": "string"
 }]
 }]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowWorkflowExecutionSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR_REGION>"))
 .build();
 ShowWorkflowExecutionRequest request = new ShowWorkflowExecutionRequest();
 try {
 ShowWorkflowExecutionResponse response = client.showWorkflowExecution(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
```

```
In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = ShowWorkflowExecutionRequest()
 response = client.show_workflow_execution(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowWorkflowExecutionRequest{}
 response, err := client.ShowWorkflowExecution(request)
 if err == nil {
 fmt.Printf("%v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | OK                    |
| 400 | Bad Request           |
| 404 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.8 获取指定函数流实例的元数据

### 功能介绍

获取指定函数流实例的元数据

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/workflows/{workflow\_id}

表 7-686 路径参数

| 参数          | 是否必选 | 参数类型   | 描述                                                         |
|-------------|------|--------|------------------------------------------------------------|
| project_id  | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |
| workflow_id | 是    | String | 函数工作流ID<br>最小长度：1<br>最大长度：512                              |

### 请求参数

无

### 响应参数

状态码： 200

表 7-687 响应 Body 参数

| 参数            | 参数类型                                | 描述                                                                                                                                          |
|---------------|-------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------|
| id            | String                              | 唯一标识ID, 流程定义ID<br>最小长度: 1<br>最大长度: 64                                                                                                       |
| workflow_urn  | String                              | 函数工作流URN, 格式为:<br>urn:fss:<region_id>:<project_id>:workflow:<package>:<workflow_name>:<version>注意:<br>package当前只支持defaultversion当前只支持latest |
| created_time  | String                              | 流程创建时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC时间<br>最小长度: 0<br>最大长度: 64                                                                              |
| updated_time  | String                              | 流程修改时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC时间<br>最小长度: 0<br>最大长度: 64                                                                              |
| created_by    | String                              | 流程创建者<br>最小长度: 0<br>最大长度: 64                                                                                                                |
| lts_group_id  | String                              | 快速函数流日志组ID, 仅快速模式函数流且日志级别不为NONE时                                                                                                            |
| lts_stream_id | String                              | 快速函数流日志流ID, 仅快速模式函数流且日志级别不为NONE时返回。                                                                                                         |
| definition    | <b>WorkflowCreateBody</b><br>object | 创建函数流的body体                                                                                                                                 |

表 7-688 WorkflowCreateBody

| 参数          | 参数类型   | 描述                            |
|-------------|--------|-------------------------------|
| name        | String | 函数流名称<br>最小长度: 1<br>最大长度: 64  |
| description | String | 函数流描述<br>最小长度: 1<br>最大长度: 512 |



| 参数                     | 参数类型                                   | 描述                                                                                                                                                                                                                                                                                               |
|------------------------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| start                  | String                                 | 流程开始节点ID<br>最小长度: 1<br>最大长度: 256                                                                                                                                                                                                                                                                 |
| triggers               | Array of <b>Trigger</b> objects        | 触发器列表<br>数组长度: 0 - 200                                                                                                                                                                                                                                                                           |
| functions              | Array of <b>Function</b> objects       | 函数列表<br>数组长度: 0 - 100                                                                                                                                                                                                                                                                            |
| states                 | Array of <b>OperationState</b> objects | 函数流节点清单, 定义参考SleepState和OperationState<br>数组长度: 1 - 100                                                                                                                                                                                                                                          |
| constants              | Object                                 | 函数流中的常量                                                                                                                                                                                                                                                                                          |
| retries                | Array of <b>Retry</b> objects          | 重试策略清单<br>数组长度: 0 - 200                                                                                                                                                                                                                                                                          |
| mode                   | String                                 | 函数流模式, 当前支持两种模式NORMAL: 标准模式, 普通模式面向普通的业务场景, 支持长时间任务, 支持执行历史持久化和查询, 只支持异步调用EXPRESS: 快速模式, 快速模式面向业务执行时长较短, 需要极致性能的场景, 只支持流程执行时长低于5分钟的场景, 不支持执行历史持久化, 支持同步和异步调用默认为标准模式<br>缺省值: <b>NORMAL</b><br>枚举值:<br><ul style="list-style-type: none"> <li>• <b>NORMAL</b></li> <li>• <b>EXPRESS</b></li> </ul> |
| express_config         | <b>ExpressConfig</b> object            | 快速函数流模式配置, 当mode为EXPRESS时需要设置                                                                                                                                                                                                                                                                    |
| enterprise_project_id  | String                                 | 企业项目ID<br>最小长度: 0<br>最大长度: 64                                                                                                                                                                                                                                                                    |
| enable_stream_response | Boolean                                | 是否返回流数据<br>缺省值: <b>false</b>                                                                                                                                                                                                                                                                     |

表 7-689 Trigger

| 参数             | 参数类型                           | 描述                                                                                                                                                                                                                                                             |
|----------------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| trigger_name   | String                         | 触发器名称<br>最小长度：1<br>最大长度：256                                                                                                                                                                                                                                    |
| trigger_type   | String                         | 触发器类型 FLOWTIMER：定时触发器 SMN：SMN触发器 APIG：APIG触发器(共享版) APIG_DE：APIG触发器(专享版) OBS：OBS触发器<br>(当前OBS触发器暂不支持中国站。)<br>最小长度：1<br>最大长度：32<br>枚举值： <ul style="list-style-type: none"><li>• FLOWTIMER</li><li>• SMN</li><li>• APIG</li><li>• APIG_DE</li><li>• OBS</li></ul> |
| enabled        | Boolean                        | 是否启用触发器<br>缺省值：true<br>枚举值： <ul style="list-style-type: none"><li>• true</li><li>• false</li></ul>                                                                                                                                                             |
| trigger_config | <b>OBSTriggerConfig</b> object | OBS触发器配置，当trigger_type为OBS时需要设置                                                                                                                                                                                                                                |

表 7-690 OBSTriggerConfig

| 参数     | 参数类型             | 描述                                                   |
|--------|------------------|------------------------------------------------------|
| bucket | String           | OBS桶名 ( trigger_type为OBS时配置 )<br>最小长度：1<br>最大长度：64   |
| events | Array of strings | OBS事件列表 ( trigger_type为OBS时配置 )<br>数组长度：1 - 64       |
| prefix | String           | 对象名前缀 ( trigger_type为OBS时配置 )<br>最小长度：1<br>最大长度：1024 |

| 参数     | 参数类型   | 描述                                                     |
|--------|--------|--------------------------------------------------------|
| suffix | String | 对象名后缀 ( trigger_type为OBS时配置 )<br>最小长度: 1<br>最大长度: 1024 |

表 7-691 Function

| 参数        | 参数类型   | 描述                                           |
|-----------|--------|----------------------------------------------|
| name      | String | 函数名称, 在单个流程中, 名称需要唯一<br>最小长度: 0<br>最大长度: 256 |
| operation | String | 函数调用URN<br>最小长度: 0<br>最大长度: 256              |
| metadata  | Object | 函数扩展属性, 由用户自己定制                              |

表 7-692 OperationState

| 参数          | 参数类型                                     | 描述                                                                                                                                                  |
|-------------|------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------|
| action_mode | String                                   | Action执行模式, 支持串行, 并行两种模式, 默认串行<br>最小长度: 1<br>最大长度: 32<br>枚举值:<br><ul style="list-style-type: none"> <li>• sequential</li> <li>• parallel</li> </ul> |
| actions     | Array of <a href="#">Action</a> objects  | 节点中要执行的操作列表<br>数组长度: 1 - 10                                                                                                                         |
| on_errors   | Array of <a href="#">OnError</a> objects | 错误处理策略<br>数组长度: 0 - 200                                                                                                                             |
| id          | String                                   | 节点ID, 需要在当前函数流中唯一<br>最小长度: 0<br>最大长度: 64                                                                                                            |

| 参数                | 参数类型                          | 描述                                                                                                                                 |
|-------------------|-------------------------------|------------------------------------------------------------------------------------------------------------------------------------|
| name              | String                        | 节点名称<br>最小长度：1<br>最大长度：256                                                                                                         |
| type              | String                        | 节点类型<br>最小长度：1<br>最大长度：32<br>枚举值：<br><ul style="list-style-type: none"> <li>• Operation</li> <li>• Sleep</li> <li>• End</li> </ul> |
| end               | Boolean                       | 是否是结束节点                                                                                                                            |
| transition        | String                        | 下一步骤节点ID<br>最小长度：0<br>最大长度：256                                                                                                     |
| state_data_filter | <b>StateDataFilter</b> object | 过滤表达式                                                                                                                              |
| duration          | Long                          | 时间等待节点等待时间（秒），节点类型为Sleep时为必填，节点类型不为Sleep时无效<br>最小值：0.001<br>最大值：86400                                                              |

表 7-693 Action

| 参数           | 参数类型                      | 描述     |
|--------------|---------------------------|--------|
| function_ref | <b>FunctionRef</b> object | 函数调用信息 |

表 7-694 FunctionRef

| 参数       | 参数类型   | 描述                                                  |
|----------|--------|-----------------------------------------------------|
| ref_name | String | 函数引用名称，需要和外层functions中的name对应<br>最小长度：0<br>最大长度：256 |

| 参数          | 参数类型   | 描述                                                                                                                                                                            |
|-------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| invoke_mode | String | 函数调用模式，目前只支持同步调用<br>最小长度：1<br>最大长度：32<br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>synchronize</b></li> <li>• <b>asynchronous</b></li> </ul>                        |
| arguments   | Object | 函数执行时的入参，支持引用constants中的常量<br>定义方式：参数路径   常量值/常量路径 参数路径指输入参数的JsonPath路径，如\$.a.b[0].c 常量值可以为数字类型，字符串类型(需要用单引号括起来)，布尔类型 常量路径为常量的JsonPath路径，但是根节点需要用\$CONST表示，示例：<br>\$CONST.a.b |

表 7-695 OnError

| 参数         | 参数类型   | 描述                                        |
|------------|--------|-------------------------------------------|
| error      | String | 错误匹配表达式，用来过滤需要处理的异常<br>最小长度：0<br>最大长度：256 |
| transition | String | 下一步骤节点ID<br>最小长度：0<br>最大长度：256            |
| retry_ref  | String | 重试策略名称<br>最小长度：0<br>最大长度：256              |

表 7-696 StateDataFilter

| 参数     | 参数类型   | 描述                                         |
|--------|--------|--------------------------------------------|
| input  | String | 输入过滤表达式 ( JsonPath )<br>最小长度：0<br>最大长度：256 |
| output | String | 输出过滤表达式 ( JsonPath )<br>最小长度：0<br>最大长度：256 |

表 7-697 Retry

| 参数           | 参数类型   | 描述                                         |
|--------------|--------|--------------------------------------------|
| name         | String | 重试策略名称，在单个流程中，名称需要唯一<br>最小长度：1<br>最大长度：256 |
| delay        | Long   | 重试间隔，单位：秒。若不传，默认为1<br>最小值：1<br>最大值：30      |
| max_attempts | Long   | 最大重试次数，。若不传，默认为3<br>最小值：1<br>最大值：8         |

表 7-698 ExpressConfig

| 参数        | 参数类型   | 描述                                                                                                                                                                                                                                                         |
|-----------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| log_level | String | 快速模式相关配置，仅在mode配置为EXPRESS时生效 快速模式下流程的执行日志级别，当前支持： ALL: 记录所有节点的执行日志 ERROR: 仅记录异常节点执行日志 NONE: 不记录日志 注意：当配置为ALL和ERROR级别时租户需要开启LTS相关权限<br>最小长度：1<br>最大长度：5<br>枚举值：<br><ul style="list-style-type: none"> <li>● ALL</li> <li>● ERROR</li> <li>● NONE</li> </ul> |

状态码： 400

表 7-699 响应 Body 参数

| 参数         | 参数类型   | 描述                       |
|------------|--------|--------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36 |

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |
| request_id | String | 请求ID<br>最小长度：2<br>最大长度：512 |

**状态码： 500**

**表 7-700 响应 Body 参数**

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |
| request_id | String | 请求ID<br>最小长度：2<br>最大长度：512 |

## 请求示例

获取函数流id为{workflow\_id}的函数流元数据。

```
GET /v2/{project_id}/fgs/workflows/{workflow_id}
```

## 响应示例

**状态码： 200**

OK

```
{
 "id": "fa466a62-73a2-49b9-9261-358dfe873b6e",
 "created_time": "2022-09-01T01:29:46Z",
 "updated_time": "2022-09-13T07:27:02Z",
 "workflow_urn": "urn:fs:cn-north-7:c53626012ba84727b938ca8bf03108ef:workflow:default:flow_nxy:latest",
 "lts_group_id": "124cef5b-d494-490d-af94-166c4bef0e0d",
 "lts_stream_id": "07e89ef7-d038-4018-97de-167b255de61f"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowWorkFlowSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowWorkFlowRequest request = new ShowWorkFlowRequest();
 try {
 ShowWorkFlowResponse response = client.showWorkFlow(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
```



```
In this example, AK and SK are stored in environment variables for authentication. Before running this
example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = ShowWorkFlowRequest()
 response = client.show_work_flow(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowWorkFlowRequest{}
 response, err := client.ShowWorkFlow(request)
 if err == nil {
 fmt.Printf("%v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | OK                    |
| 400 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.9 修改指定函数流实例的元数据

### 功能介绍

修改指定函数流实例的元数据

### 调用方法

请参见[如何调用API](#)。

## URI

PUT /v2/{project\_id}/fgs/workflows/{workflow\_id}

表 7-701 路径参数

| 参数          | 是否必选 | 参数类型   | 描述                                                         |
|-------------|------|--------|------------------------------------------------------------|
| project_id  | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |
| workflow_id | 是    | String | 函数 workflow ID<br>最小长度：1<br>最大长度：512                       |

## 请求参数

表 7-702 请求 Body 参数

| 参数          | 是否必选 | 参数类型                                   | 描述                                                                                                                                                                                                                                                                       |
|-------------|------|----------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| name        | 是    | String                                 | 函数流名称<br>最小长度：1<br>最大长度：64                                                                                                                                                                                                                                               |
| description | 否    | String                                 | 函数流描述<br>最小长度：1<br>最大长度：512                                                                                                                                                                                                                                              |
| start       | 是    | String                                 | 流程开始节点ID<br>最小长度：1<br>最大长度：256                                                                                                                                                                                                                                           |
| triggers    | 否    | Array of <b>Trigger</b> objects        | 触发器列表<br>数组长度：0 - 200                                                                                                                                                                                                                                                    |
| functions   | 是    | Array of <b>Function</b> objects       | 函数列表<br>数组长度：0 - 100                                                                                                                                                                                                                                                     |
| states      | 是    | Array of <b>OperationState</b> objects | 函数流节点清单，定义参考 SleepState和OperationState<br>数组长度：1 - 100                                                                                                                                                                                                                   |
| constants   | 是    | Object                                 | 函数流中的常量                                                                                                                                                                                                                                                                  |
| retries     | 是    | Array of <b>Retry</b> objects          | 重试策略清单<br>数组长度：0 - 200                                                                                                                                                                                                                                                   |
| mode        | 否    | String                                 | 函数流模式，当前支持两种模式<br>NORMAL: 标准模式，普通模式面向普通的业务场景，支持长时间任务，支持执行历史持久化和查询，只支持异步调用<br>EXPRESS: 快速模式，快速模式面向业务执行时长较短，需要极致性能的场景，只支持流程执行时长低于5分钟的场景，不支持执行历史持久化，支持同步和异步调用默认为标准模式<br>缺省值：NORMAL<br>枚举值：<br><ul style="list-style-type: none"> <li>● NORMAL</li> <li>● EXPRESS</li> </ul> |

| 参数                     | 是否必选 | 参数类型                        | 描述                                          |
|------------------------|------|-----------------------------|---------------------------------------------|
| express_config         | 否    | <b>ExpressConfig</b> object | 快速函数流模式配置，当mode为EXPRESS时需要设置                |
| enterprise_project_id  | 否    | String                      | 企业项目ID<br>最小长度： <b>0</b><br>最大长度： <b>64</b> |
| enable_stream_response | 否    | Boolean                     | 是否返回流数据<br>缺省值： <b>false</b>                |

表 7-703 Trigger

| 参数             | 是否必选 | 参数类型                           | 描述                                                                                                                                                                                                                                                                                                                         |
|----------------|------|--------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| trigger_name   | 是    | String                         | 触发器名称<br>最小长度： <b>1</b><br>最大长度： <b>256</b>                                                                                                                                                                                                                                                                                |
| trigger_type   | 是    | String                         | 触发器类型 FLOWTIMER：定时触发器 SMN：SMN触发器 APIG：APIG触发器(共享版) APIG_DE：APIG触发器(专享版) OBS：OBS触发器<br>(当前OBS触发器暂不支持中国站。)<br>最小长度： <b>1</b><br>最大长度： <b>32</b><br>枚举值：<br><ul style="list-style-type: none"> <li>● <b>FLOWTIMER</b></li> <li>● <b>SMN</b></li> <li>● <b>APIG</b></li> <li>● <b>APIG_DE</b></li> <li>● <b>OBS</b></li> </ul> |
| enabled        | 否    | Boolean                        | 是否启用触发器<br>缺省值： <b>true</b><br>枚举值：<br><ul style="list-style-type: none"> <li>● <b>true</b></li> <li>● <b>false</b></li> </ul>                                                                                                                                                                                             |
| trigger_config | 否    | <b>OBSTriggerConfig</b> object | OBS触发器配置，当trigger_type为OBS时需要设置                                                                                                                                                                                                                                                                                            |

表 7-704 OBSTriggerConfig

| 参数     | 是否必选 | 参数类型             | 描述                                                |
|--------|------|------------------|---------------------------------------------------|
| bucket | 是    | String           | OBS桶名（trigger_type为OBS时配置）<br>最小长度：1<br>最大长度：64   |
| events | 是    | Array of strings | OBS事件列表（trigger_type为OBS时配置）<br>数组长度：1 - 64       |
| prefix | 是    | String           | 对象名前缀（trigger_type为OBS时配置）<br>最小长度：1<br>最大长度：1024 |
| suffix | 是    | String           | 对象名后缀（trigger_type为OBS时配置）<br>最小长度：1<br>最大长度：1024 |

表 7-705 Function

| 参数        | 是否必选 | 参数类型   | 描述                                       |
|-----------|------|--------|------------------------------------------|
| name      | 是    | String | 函数名称，在单个流程中，名称需要唯一<br>最小长度：0<br>最大长度：256 |
| operation | 是    | String | 函数调用URN<br>最小长度：0<br>最大长度：256            |
| metadata  | 否    | Object | 函数扩展属性，由用户自己定制                           |

表 7-706 OperationState

| 参数                | 是否必选 | 参数类型                            | 描述                                                                                                                                             |
|-------------------|------|---------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------|
| action_mode       | 否    | String                          | Action执行模式，支持串行，并行两种模式，默认串行<br>最小长度：1<br>最大长度：32<br>枚举值：<br><ul style="list-style-type: none"> <li>• sequential</li> <li>• parallel</li> </ul> |
| actions           | 是    | Array of <b>Action</b> objects  | 节点中要执行的操作列表<br>数组长度：1 - 10                                                                                                                     |
| on_errors         | 否    | Array of <b>OnError</b> objects | 错误处理策略<br>数组长度：0 - 200                                                                                                                         |
| id                | 是    | String                          | 节点ID，需要在当前函数流中唯一<br>最小长度：0<br>最大长度：64                                                                                                          |
| name              | 是    | String                          | 节点名称<br>最小长度：1<br>最大长度：256                                                                                                                     |
| type              | 是    | String                          | 节点类型<br>最小长度：1<br>最大长度：32<br>枚举值：<br><ul style="list-style-type: none"> <li>• Operation</li> <li>• Sleep</li> <li>• End</li> </ul>             |
| end               | 是    | Boolean                         | 是否是结束节点                                                                                                                                        |
| transition        | 是    | String                          | 下一步骤节点ID<br>最小长度：0<br>最大长度：256                                                                                                                 |
| state_data_filter | 否    | <b>StateDataFilter</b> object   | 过滤表达式                                                                                                                                          |

| 参数       | 是否必选 | 参数类型 | 描述                                                                                    |
|----------|------|------|---------------------------------------------------------------------------------------|
| duration | 否    | Long | 时间等待节点等待时间（秒），节点类型为Sleep时为必填，节点类型不为Sleep时无效<br>最小值： <b>0.001</b><br>最大值： <b>86400</b> |

表 7-707 Action

| 参数           | 是否必选 | 参数类型                         | 描述     |
|--------------|------|------------------------------|--------|
| function_ref | 否    | <b>FunctionRef</b><br>object | 函数调用信息 |

表 7-708 FunctionRef

| 参数          | 是否必选 | 参数类型   | 描述                                                                                                                                                                                  |
|-------------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ref_name    | 是    | String | 函数引用名称，需要和外层 functions 中的 name 对应<br>最小长度： <b>0</b><br>最大长度： <b>256</b>                                                                                                             |
| invoke_mode | 否    | String | 函数调用模式，目前只支持同步调用<br>最小长度： <b>1</b><br>最大长度： <b>32</b><br>枚举值：<br><ul style="list-style-type: none"> <li>• <b>synchronize</b></li> <li>• <b>asynchronous</b></li> </ul>              |
| arguments   | 是    | Object | 函数执行时的入参，支持引用 constants 中的常量 定义方式：<br>参数路径   常量值/常量路径 参数路径指输入参数的 JsonPath 路径，如 \$.a.b[0].c 常量值可以为数字类型，字符串类型(需要用单引号括起来)，布尔类型 常量路径为常量的 JsonPath 路径，但是根节点需要用 \$CONST 表示，示例：\$CONST.a.b |

表 7-709 OnError

| 参数         | 是否必选 | 参数类型   | 描述                                        |
|------------|------|--------|-------------------------------------------|
| error      | 否    | String | 错误匹配表达式，用来过滤需要处理的异常<br>最小长度：0<br>最大长度：256 |
| transition | 否    | String | 下一步骤节点ID<br>最小长度：0<br>最大长度：256            |
| retry_ref  | 否    | String | 重试策略名称<br>最小长度：0<br>最大长度：256              |

表 7-710 StateDataFilter

| 参数     | 是否必选 | 参数类型   | 描述                                         |
|--------|------|--------|--------------------------------------------|
| input  | 否    | String | 输入过滤表达式 ( JsonPath )<br>最小长度：0<br>最大长度：256 |
| output | 否    | String | 输出过滤表达式 ( JsonPath )<br>最小长度：0<br>最大长度：256 |

表 7-711 Retry

| 参数    | 是否必选 | 参数类型   | 描述                                         |
|-------|------|--------|--------------------------------------------|
| name  | 是    | String | 重试策略名称，在单个流程中，名称需要唯一<br>最小长度：1<br>最大长度：256 |
| delay | 否    | Long   | 重试间隔，单位：秒。若不传，默认为1<br>最小值：1<br>最大值：30      |



| 参数           | 是否必选 | 参数类型 | 描述                                 |
|--------------|------|------|------------------------------------|
| max_attempts | 否    | Long | 最大重试次数，。若不传，默认为3<br>最小值：1<br>最大值：8 |

表 7-712 ExpressConfig

| 参数        | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                         |
|-----------|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| log_level | 否    | String | 快速模式相关配置，仅在mode配置为EXPRESS时生效 快速模式下流程的执行日志级别，当前支持： ALL: 记录所有节点的执行日志 ERROR: 仅记录异常节点执行日志 NONE: 不记录日志 注意：当配置为ALL和ERROR级别时租户需要开启LTS相关权限<br>最小长度：1<br>最大长度：5<br>枚举值：<br><ul style="list-style-type: none"> <li>• ALL</li> <li>• ERROR</li> <li>• NONE</li> </ul> |

## 响应参数

状态码： 200

表 7-713 响应 Body 参数

| 参数           | 参数类型   | 描述                                                                                                                                          |
|--------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------|
| id           | String | 唯一标识ID，流程定义ID<br>最小长度：1<br>最大长度：64                                                                                                          |
| workflow_urn | String | 函数工作流URN, 格式为：<br>urn:fss:<region_id>:<project_id>:workflow:<package>:<workflow_name>:<version>注意：<br>package当前只支持defaultversion当前只支持latest |

| 参数                     | 参数类型    | 描述                                                             |
|------------------------|---------|----------------------------------------------------------------|
| name                   | String  | 流程定义名称<br>最小长度: 1<br>最大长度: 64                                  |
| description            | String  | 流程定义描述<br>最小长度: 0<br>最大长度: 512                                 |
| created_time           | String  | 流程创建时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC时间<br>最小长度: 0<br>最大长度: 64 |
| updated_time           | String  | 流程修改时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC时间<br>最小长度: 0<br>最大长度: 64 |
| created_by             | String  | 流程创建者<br>最小长度: 0<br>最大长度: 64                                   |
| enable_stream_response | Boolean | 是否返回流数据<br>缺省值: <b>false</b>                                   |

状态码: 400

表 7-714 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

状态码: 500

表 7-715 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |
| request_id | String | 请求ID<br>最小长度：2<br>最大长度：512 |

## 请求示例

修改函数流id为{workflow\_id}的函数流元数据。

PUT /v2/{project\_id}/fgs/workflows/{workflow\_id}

```
{
 "name": "workflow_test",
 "start": "test",
 "functions": [{
 "name": "test",
 "operation": "urn:fss:{region}:c53626012ba84727b938ca8bf03108ef:test:latest"
 }],
 "constants": { },
 "states": [{
 "id": "test",
 "name": "test",
 "type": "Operation",
 "end": false,
 "transition": "end",
 "actions": [{
 "function_ref": {
 "ref_name": "test",
 "arguments": {
 "execution_name": "$.test"
 }
 }
 }
]
}, {
 "id": "end",
 "name": "end",
 "type": "End",
 "end": true,
 "transition": "",
 "state_data_filter": {
 "input": "",
 "output": ""
 },
 "actions": [],
 "action_mode": ""
}],
 "retries": [{
 "name": "test"
 }]
}
```

## 响应示例

状态码： 200

Success

```
{
 "id": "48aff955-ea10-4ffd-b426-190bc192dc14",
 "workflow_urn": "urn:fss:cn-north-7:73d69ae0cfcf460190522d060f05ad:workflow:default:testflow:latest",
 "name": "test-flow",
 "description": "desc",
 "created_time": "2021-01-01T00:00:00Z",
 "updated_time": "2021-01-01T00:00:00Z",
 "created_by": "admin"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

修改函数流id为{workflow\_id}的函数流元数据。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.List;
import java.util.ArrayList;

public class UpdateWorkFlowSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 UpdateWorkFlowRequest request = new UpdateWorkFlowRequest();
 WorkflowCreateBody body = new WorkflowCreateBody();
 List<Retry> listbodyRetries = new ArrayList<>();
 listbodyRetries.add(
 new Retry()
 .withName("test")
);
 StateDataFilter stateDataFilterStates = new StateDataFilter();
 stateDataFilterStates.withInput("")
 .withOutput("");
 }
}
```

```
FunctionRef functionRefActions = new FunctionRef();
functionRefActions.withRefName("test")
 .withArguments("{\"execution_name\":\"$.test\"}");
List<Action> listStatesActions = new ArrayList<>();
listStatesActions.add(
 new Action()
 .withFunctionRef(functionRefActions)
);
List<OperationState> listbodyStates = new ArrayList<>();
listbodyStates.add(
 new OperationState()
 .withActions(listStatesActions)
 .withId("test")
 .withName("test")
 .withType(OperationState.TypeEnum.fromValue("Operation"))
 .withEnd(false)
 .withTransition("end")
);
listbodyStates.add(
 new OperationState()
 .withActionMode(OperationState.ActionModeEnum.fromValue(""))
 .withActions()
 .withId("end")
 .withName("end")
 .withType(OperationState.TypeEnum.fromValue("End"))
 .withEnd(true)
 .withTransition("")
 .withStateDataFilter(stateDataFilterStates)
);
List<Function> listbodyFunctions = new ArrayList<>();
listbodyFunctions.add(
 new Function()
 .withName("test")
 .withOperation("urn:fss:{regin}:c53626012ba84727b938ca8bf03108ef:test:latest")
);
body.withRetries(listbodyRetries);
body.withConstants(new Object());
body.withStates(listbodyStates);
body.withFunctions(listbodyFunctions);
body.withStart("test");
body.withName("workflow_test");
request.withBody(body);
try {
 UpdateWorkFlowResponse response = client.updateWorkFlow(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

修改函数流id为{workflow\_id}的函数流元数据。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *
```

```
if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = UpdateWorkFlowRequest()
 listRetriesbody = [
 Retry(
 name="test"
)
]
 stateDataFilterStates = StateDataFilter(
 input="",
 output=""
)
 functionRefActions = FunctionRef(
 ref_name="test",
 arguments="{\"execution_name\": \"$.test\"}"
)
 listActionsStates = [
 Action(
 function_ref=functionRefActions
)
]
 listStatesbody = [
 OperationState(
 actions=listActionsStates,
 id="test",
 name="test",
 type="Operation",
 end=False,
 transition="end"
),
 OperationState(
 action_mode="",
 id="end",
 name="end",
 type="End",
 end=True,
 transition="",
 state_data_filter=stateDataFilterStates
)
]
 listFunctionsbody = [
 Function(
 name="test",
 operation="urn:fss:{region}:c53626012ba84727b938ca8bf03108ef:test:latest"
)
]
 request.body = WorkflowCreateBody(
 retries=listRetriesbody,
 constants={},
 states=listStatesbody,
 functions=listFunctionsbody,
 start="test",
 name="workflow_test"
```

```
)
response = client.update_work_flow(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

修改函数流id为{workflow\_id}的函数流元数据。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.UpdateWorkFlowRequest{}
 var listRetriesbody = []model.Retry{
 {
 Name: "test",
 },
 }
 inputStateDataFilter:= ""
 outputStateDataFilter:= ""
 stateDataFilterStates := &model.StateDataFilter{
 Input: &inputStateDataFilter,
 Output: &outputStateDataFilter,
 }
 var argumentsFunctionRef interface{} = "{\"execution_name\": \"$.test\"}"
 functionRefActions := &model.FunctionRef{
 RefName: "test",
 Arguments: &argumentsFunctionRef,
 }
 var listActionsStates = []model.Action{
 {
 FunctionRef: functionRefActions,
 },
 }
 actionModeStates:= model.GetOperationStateActionModeEnum().EMPTY
 var listStatesbody = []model.OperationState{
 {
 Actions: listActionsStates,
```

```
 Id: "test",
 Name: "test",
 Type: model.GetOperationStateTypeEnum().OPERATION,
 End: false,
 Transition: "end",
 },
 {
 ActionMode: &actionModeStates,
 Id: "end",
 Name: "end",
 Type: model.GetOperationStateTypeEnum().END,
 End: true,
 Transition: "",
 StateDataFilter: stateDataFilterStates,
 },
}
var listFunctionsbody = []model.Function{
 {
 Name: "test",
 Operation: "urn:fss:{regin}:c53626012ba84727b938ca8bf03108ef:test:latest",
 },
}
var constantsWorkflowCreateBody interface{} = make(map[string]string)
request.Body = &model.WorkflowCreateBody{
 Retries: listRetriesbody,
 Constants: &constantsWorkflowCreateBody,
 States: listStatesbody,
 Functions: listFunctionsbody,
 Start: "test",
 Name: "workflow_test",
}
response, err := client.UpdateWorkFlow(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | Success               |
| 400 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。



## 7.14.10 获取函数流指标

### 功能介绍

获取函数流指标

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/workflow-statistic

表 7-716 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                                         |
|------------|------|--------|------------------------------------------------------------|
| project_id | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |

表 7-717 Query 参数

| 参数          | 是否必选 | 参数类型   | 描述                                                                                                                |
|-------------|------|--------|-------------------------------------------------------------------------------------------------------------------|
| period      | 否    | String | 时间段，单位为分钟                                                                                                         |
| start_time  | 否    | String | 开始时间，精确到ms的时间戳                                                                                                    |
| end_time    | 否    | String | 结束时间，精确到ms的时间戳                                                                                                    |
| metric_type | 否    | String | 指标类型，为空或不在取值范围内时，查询所有指标。取值范围：totalCount 调用次数；errorCount 错误次数；averageDuration 运行时间；running 运行中个数；rejectCount 拒绝个数。 |

### 请求参数

无

### 响应参数

状态码： 200

表 7-718 响应 Body 参数

| 参数            | 参数类型                                             | 描述         |
|---------------|--------------------------------------------------|------------|
| count         | Array of <a href="#">SlaReportsValue</a> objects | 执行次数       |
| duration      | Array of <a href="#">SlaReportsValue</a> objects | 平均时延, 单位毫秒 |
| fail_count    | Array of <a href="#">SlaReportsValue</a> objects | 错误次数       |
| running_count | Array of <a href="#">SlaReportsValue</a> objects | 运行中数量      |

表 7-719 SlaReportsValue

| 参数        | 参数类型   | 描述  |
|-----------|--------|-----|
| timestamp | Long   | 时间戳 |
| value     | Double | 值   |

状态码: 400

表 7-720 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

状态码: 500

表 7-721 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

## 请求示例

获取函数流指标。

```
GET /v2/{projectId}/fgs/workflow-statistic
```

## 响应示例

状态码: 200

函数流统计信息

```
{
 "count": [{
 "timestamp": 0,
 "value": 0
 }],
 "duration": [{
 "timestamp": 0,
 "value": 0
 }],
 "fail_count": [{
 "timestamp": 0,
 "value": 0
 }],
 "running_count": [{
 "timestamp": 0,
 "value": 0
 }]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
```

```
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowTenantMetricSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR_REGION>"))
 .build();
 ShowTenantMetricRequest request = new ShowTenantMetricRequest();
 request.withPeriod("<period>");
 request.withStartTime("<start_time>");
 request.withEndTime("<end_time>");
 request.withMetricType("<metric_type>");
 try {
 ShowTenantMetricResponse response = client.showTenantMetric(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \
```

```
client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = ShowTenantMetricRequest()
 request.period = "<period>"
 request.start_time = "<start_time>"
 request.end_time = "<end_time>"
 request.metric_type = "<metric_type>"
 response = client.show_tenant_metric(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowTenantMetricRequest{
 periodRequest:= "<period>"
 request.Period = &periodRequest
 startTimeRequest:= "<start_time>"
 request.StartTime = &startTimeRequest
 endTimeRequest:= "<end_time>"
 request.EndTime = &endTimeRequest
 metricTypeRequest:= "<metric_type>"
 request.MetricType = &metricTypeRequest
 }
 response, err := client.ShowTenantMetric(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | 函数流统计信息               |
| 400 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

### 7.14.11 获取指定函数流指标

#### 功能介绍

获取指定函数流指标

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/fgs/workflow-statistic/{workflow\_urn}

表 7-722 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                                                                                                                                                           |
|--------------|------|--------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| project_id   | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64                                                                                                                   |
| workflow_urn | 是    | String | 函数工作流URN，格式为：<br>urn:fss:<region_id>:<project_id>:<br>workflow:<package>:<workflow_name>:<br><version>注意：<br>package当前只支持<br>defaultversion当前只支持latest<br>最小长度：1<br>最大长度：128 |

表 7-723 Query 参数

| 参数         | 是否必选 | 参数类型   | 描述              |
|------------|------|--------|-----------------|
| period     | 否    | String | 时间段, 单位为分钟      |
| start_time | 否    | String | 开始时间, 精确到ms的时间戳 |
| end_time   | 否    | String | 结束时间, 精确到ms的时间戳 |

### 请求参数

无

### 响应参数

状态码: 200

表 7-724 响应 Body 参数

| 参数            | 参数类型                                             | 描述         |
|---------------|--------------------------------------------------|------------|
| count         | Array of <a href="#">SlaReportsValue</a> objects | 执行次数       |
| duration      | Array of <a href="#">SlaReportsValue</a> objects | 平均时延, 单位毫秒 |
| fail_count    | Array of <a href="#">SlaReportsValue</a> objects | 错误次数       |
| running_count | Array of <a href="#">SlaReportsValue</a> objects | 运行中数量      |

表 7-725 SlaReportsValue

| 参数        | 参数类型   | 描述  |
|-----------|--------|-----|
| timestamp | Long   | 时间戳 |
| value     | Double | 值   |

状态码: 400

表 7-726 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

状态码: 500

表 7-727 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

## 请求示例

获取指定函数流指标。

```
GET /v2/{project_id}/fgs/workflow-statistic/{workflow_urn}
```

## 响应示例

状态码: 200

函数流统计信息

```
{
 "count": [{
 "timestamp": 0,
 "value": 0
 }
]
```



```
 }],
 "duration" : [{
 "timestamp" : 0,
 "value" : 0
 }],
 "fail_count" : [{
 "timestamp" : 0,
 "value" : 0
 }],
 "running_count" : [{
 "timestamp" : 0,
 "value" : 0
 }]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowWorkFlowMetricSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowWorkFlowMetricRequest request = new ShowWorkFlowMetricRequest();
 request.withPeriod("<period>");
 request.withStartTime("<start_time>");
 request.withEndTime("<end_time>");
 try {
 ShowWorkFlowMetricResponse response = client.showWorkFlowMetric(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 }
 }
}
```

```
 System.out.println(e.getErrorMsg());
 }
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowWorkFlowMetricRequest()
 request.period = "<period>"
 request.start_time = "<start_time>"
 request.end_time = "<end_time>"
 response = client.show_work_flow_metric(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()
```

```

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.ShowWorkFlowMetricRequest{
 periodRequest:= "<period>"
 request.Period = &periodRequest
 startTimeRequest:= "<start_time>"
 request.StartTime = &startTimeRequest
 endTimeRequest:= "<end_time>"
 request.EndTime = &endTimeRequest
 response, err := client.ShowWorkFlowMetric(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | 函数流统计信息               |
| 400 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.12 重试函数流

### 功能介绍

重试函数流(执行标准函数流失败时可使用执行id调用此接口重试，快速工作流不支持重试)

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/fgs/workflows/{workflow\_id}/executions/{execution\_id}/retry

表 7-728 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                                         |
|--------------|------|--------|------------------------------------------------------------|
| project_id   | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |
| workflow_id  | 是    | String | 函数工作流ID<br>最小长度：1<br>最大长度：512                              |
| execution_id | 是    | String | 函数流执行实例ID<br>最小长度：1<br>最大长度：64                             |

### 请求参数

无

### 响应参数

状态码：400

表 7-729 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |
| request_id | String | 请求ID<br>最小长度：2<br>最大长度：512 |

状态码：404

表 7-730 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |

状态码: 500

表 7-731 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |
| request_id | String | 请求ID<br>最小长度: 2<br>最大长度: 512 |

## 请求示例

重试执行id为{execution\_id}的函数流执行实例。

```
POST /v2/{project_id}/fgs/workflows/{workflow_id}/executions/{execution_id}/retry
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;
import com.huaweicloud.sdk.core.auth.ICredential;
```

```
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class RetryWorkFlowSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 RetryWorkFlowRequest request = new RetryWorkFlowRequest();
 try {
 RetryWorkFlowResponse response = client.retryWorkFlow(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
```

```

 .build()

 try:
 request = RetryWorkFlowRequest()
 response = client.retry_work_flow(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)

```

## Go

```

package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.RetryWorkFlowRequest{}
 response, err := client.RetryWorkFlow(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | OK          |
| 400 | Bad Request |

| 状态码 | 描述                    |
|-----|-----------------------|
| 404 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.13 停止函数流

### 功能介绍

停止函数流

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/fgs/workflows/{workflow\_id}/executions/{execution\_id}/  
terminate

表 7-732 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                                         |
|--------------|------|--------|------------------------------------------------------------|
| project_id   | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |
| workflow_id  | 是    | String | 函数 workflow ID<br>最小长度：1<br>最大长度：512                       |
| execution_id | 是    | String | 函数流执行实例ID<br>最小长度：1<br>最大长度：64                             |

### 请求参数

无



## 响应参数

状态码： 400

表 7-733 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度： 1<br>最大长度： 36   |
| error_msg  | String | 错误描述<br>最小长度： 2<br>最大长度： 512 |
| request_id | String | 请求ID<br>最小长度： 2<br>最大长度： 512 |

状态码： 404

表 7-734 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度： 1<br>最大长度： 36   |
| error_msg  | String | 错误描述<br>最小长度： 2<br>最大长度： 512 |

状态码： 500

表 7-735 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度： 1<br>最大长度： 36 |

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |
| request_id | String | 请求ID<br>最小长度：2<br>最大长度：512 |

## 请求示例

停止执行id{execution\_id}为函数流执行实例。

```
POST /v2/{project_id}/fgs/workflows/{workflow_id}/executions/{execution_id}/terminate
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class StopWorkFlowSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 StopWorkFlowRequest request = new StopWorkFlowRequest();
 try {
```

```
 StopWorkFlowResponse response = client.stopWorkFlow(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = StopWorkFlowRequest()
 response = client.stop_work_flow(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
ak := os.Getenv("CLOUD_SDK_AK")
sk := os.Getenv("CLOUD_SDK_SK")

auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.StopWorkFlowRequest{}
response, err := client.StopWorkFlow(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | OK                    |
| 400 | Bad Request           |
| 404 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.14 分页查询指定函数流记录

### 功能介绍

分页查询指定函数流记录

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/workflows/{workflow\_id}/executions-history

表 7-736 路径参数

| 参数          | 是否必选 | 参数类型   | 描述                                                         |
|-------------|------|--------|------------------------------------------------------------|
| project_id  | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |
| workflow_id | 是    | String | 函数工作流ID<br>最小长度：1<br>最大长度：512                              |

表 7-737 Query 参数

| 参数         | 是否必选 | 参数类型    | 描述                                                                            |
|------------|------|---------|-------------------------------------------------------------------------------|
| offset     | 是    | Integer | 偏移量，表示从此偏移量开始查询，offset大于等于0<br>最小值：0                                          |
| limit      | 是    | Integer | 分页查询，每页查询数据条数，取值范围：1,2,3...100<br>最小值：1<br>最大值：100                            |
| start_time | 否    | String  | 查询开始时间，UTC时间，格式：YYYY-MM-DD hh:mm:ss。若起始时间未填写，以终止时间前推3天为起始时间。                  |
| end_time   | 否    | String  | 查询结束时间，UTC时间，格式：YYYY-MM-DD hh:mm:ss。若终止时间未填写，以起始时间后退3天为终止时间。若均未填写，默认查询最近3天数据。 |

### 请求参数

无

### 响应参数

状态码： 200

表 7-738 响应 Body 参数

| 参数         | 参数类型                                                | 描述                        |
|------------|-----------------------------------------------------|---------------------------|
| executions | Array of <a href="#">FlowExecutionBrief</a> objects | 流程执行信息列表<br>数组长度: 0 - 999 |
| total      | Long                                                | 总条数                       |
| size       | Integer                                             | 每页查询的条数                   |

表 7-739 FlowExecutionBrief

| 参数           | 参数类型   | 描述                                                                                                                                                                             |
|--------------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| workflow_id  | String | 流程定义ID<br>最小长度: 1<br>最大长度: 64                                                                                                                                                  |
| workflow_urn | String | 函数工作流URN, 格式为:<br>urn:fss:<region_id>:<project_id>:workflow:<package>:<workflow_name>:<version>注意:<br>package当前只支持defaultversion当前只支持latest                                    |
| execution_id | String | 流程执行实例ID<br>最小长度: 1<br>最大长度: 64                                                                                                                                                |
| status       | String | 流程实例执行状态<br>最小长度: 1<br>最大长度: 32<br>枚举值:<br><ul style="list-style-type: none"> <li>• success</li> <li>• fail</li> <li>• running</li> <li>• timeout</li> <li>• cancel</li> </ul> |
| begin_time   | String | 流程实例创建时间, 格式: yyyy-MM-ddTHH:mm:ssZ, UTC时间<br>最小长度: 0<br>最大长度: 64                                                                                                               |

| 参数               | 参数类型   | 描述                                                            |
|------------------|--------|---------------------------------------------------------------|
| end_time         | String | 流程实例结束时间，格式：yyyy-MM-ddTHH:mm:ssZ，UTC时间<br>最小长度：0<br>最大长度：64   |
| last_update_time | String | 流程实例上次更新时间，格式：yyyy-MM-ddTHH:mm:ssZ，UTC时间<br>最小长度：0<br>最大长度：64 |
| created_by       | String | 流程实例创建者<br>最小长度：1<br>最大长度：32                                  |

状态码：400

表 7-740 响应 Body 参数

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36   |
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |
| request_id | String | 请求ID<br>最小长度：2<br>最大长度：512 |

状态码：500

表 7-741 响应 Body 参数

| 参数         | 参数类型   | 描述                       |
|------------|--------|--------------------------|
| error_code | String | 错误码<br>最小长度：1<br>最大长度：36 |

| 参数         | 参数类型   | 描述                         |
|------------|--------|----------------------------|
| error_msg  | String | 错误描述<br>最小长度：2<br>最大长度：512 |
| request_id | String | 请求ID<br>最小长度：2<br>最大长度：512 |

## 请求示例

分页获取函数流id为{workflow\_id}的函数流执行实例列表。

```
GET /v2/{project_id}/fgs/workflows/{workflow_id}/executions-history
```

## 响应示例

**状态码：200**

函数流执行查询返回体

```
{
 "executions": [{
 "workflow_id": "c0aecc37-8b78-4fc5-81af-14af495dc053",
 "execution_id": "41c25eec-a1ba-4789-9851-eeead00cd6b9",
 "status": "success",
 "begin_time": "2022-10-11T06:30:45Z",
 "end_time": "2022-10-11T06:30:47Z",
 "last_update_time": "2022-10-11T06:30:45Z",
 "created_by": "c0aecc37-8b78-4fc5-81af-14af495dc053",
 "workflow_urn": "urn:fss:cn-
north-7:c53626012ba84727b938ca8bf03108ef:workflow:default:flow_nxy_tp:latest"
 }, {
 "workflow_id": "c0aecc37-8b78-4fc5-81af-14af495dc053",
 "execution_id": "74a29a25-2d6d-4b50-a02e-63b7c5b8ab93",
 "status": "success",
 "begin_time": "2022-10-11T06:30:24Z",
 "end_time": "2022-10-11T06:30:26Z",
 "last_update_time": "2022-10-11T06:30:25Z",
 "created_by": "c0aecc37-8b78-4fc5-81af-14af495dc053",
 "workflow_urn": "urn:fss:cn-
north-7:c53626012ba84727b938ca8bf03108ef:workflow:default:flow_nxy_tp:latest"
 }],
 "total": 100,
 "size": 10
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
```



```
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowWorkflowExecutionForPageSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowWorkflowExecutionForPageRequest request = new ShowWorkflowExecutionForPageRequest();
 request.withOffset(<offset>);
 request.withLimit(<limit>);
 request.withStartTime("<start_time>");
 request.withEndTime("<end_time>");
 try {
 ShowWorkflowExecutionForPageResponse response =
 client.showWorkflowExecutionForPage(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \
```

```
client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = ShowWorkflowExecutionForPageRequest()
 request.offset = <offset>
 request.limit = <limit>
 request.start_time = "<start_time>"
 request.end_time = "<end_time>"
 response = client.show_workflow_execution_for_page(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowWorkflowExecutionForPageRequest{}
 request.Offset = int32(<offset>)
 request.Limit = int32(<limit>)
 startTimeRequest := "<start_time>"
 request.StartTime = &startTimeRequest
 endTimeRequest := "<end_time>"
 request.EndTime = &endTimeRequest
 response, err := client.ShowWorkflowExecutionForPage(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | 函数流执行查询返回体            |
| 400 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.14.15 回调工作流

### 功能介绍

回调工作流

### 调用方法

请参见[如何调用API](#)。

## URI

POST /v2/{project\_id}/fgs/workflows/{workflow\_id}/callback

表 7-742 路径参数

| 参数          | 是否必选 | 参数类型   | 描述                                                         |
|-------------|------|--------|------------------------------------------------------------|
| project_id  | 是    | String | 项目ID，获取方式请参见 <a href="#">获取项目ID</a> 。<br>最小长度：1<br>最大长度：64 |
| workflow_id | 是    | String | 工作流定义ID<br>最小长度：1<br>最大长度：512                              |

## 请求参数

表 7-743 请求 Header 参数

| 参数                  | 是否必选 | 参数类型   | 描述                |
|---------------------|------|--------|-------------------|
| X-Workflow-Run-Id   | 是    | String | workflow run id   |
| X-Workflow-State-Id | 是    | String | workflow state id |

表 7-744 请求 Body 参数

| 参数     | 是否必选 | 参数类型   | 描述                                                                                                               |
|--------|------|--------|------------------------------------------------------------------------------------------------------------------|
| result | 是    | String | 执行结果<br>最小长度：1<br>最大长度：64<br>枚举值：<br><ul style="list-style-type: none"> <li>• success</li> <li>• fail</li> </ul> |
| error  | 否    | String | 错误信息<br>最小长度：1<br>最大长度：64                                                                                        |
| output | 是    | Object | 工作流的执行结果，JSON格式，仅在status为success时有值                                                                              |

## 响应参数

状态码：200

表 7-745 响应 Body 参数

| 参数           | 参数类型   | 描述                       |
|--------------|--------|--------------------------|
| execution_id | String | 错误码<br>最小长度：1<br>最大长度：64 |

状态码：400

表 7-746 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |

状态码: 404

表 7-747 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |

状态码: 500

表 7-748 响应 Body 参数

| 参数         | 参数类型   | 描述                           |
|------------|--------|------------------------------|
| error_code | String | 错误码<br>最小长度: 1<br>最大长度: 36   |
| error_msg  | String | 错误描述<br>最小长度: 2<br>最大长度: 512 |

## 请求示例

回调 workflow

```
POST /v2/{project_id}/fgs/workflows/{workflow_id}/callback
{
```

```
"result": "success",
"output": {}
}
```

## 响应示例

**状态码： 200**

Success

```
{
 "execution_id" : "48aff955-ea10-4ffd-b426-190bc192dc13"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CreateCallbackWorkflowSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CreateCallbackWorkflowRequest request = new CreateCallbackWorkflowRequest();
 CallbackWorkflowRequestBody body = new CallbackWorkflowRequestBody();
 request.withBody(body);
 try {
 CreateCallbackWorkflowResponse response = client.createCallbackWorkflow(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

```
}
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateCallbackWorkflowRequest()
 request.body = CallbackWorkflowRequestBody(
)
 response = client.create_callback_workflow(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
```

```
WithRegion(region.ValueOf("<YOUR REGION>")).
WithCredential(auth).
Build()

request := &model.CreateCallbackWorkflowRequest{
request.Body = &model.CallbackWorkflowRequestBody{
}
response, err := client.CreateCallbackWorkflow(request)
if err == nil {
 fmt.Printf("%v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                    |
|-----|-----------------------|
| 200 | Success               |
| 400 | Bad Request           |
| 404 | Bad Request           |
| 500 | Internal Server Error |

## 错误码

请参见[错误码](#)。

## 7.15 函数异步配置

### 7.15.1 获取函数异步配置信息

#### 功能介绍

获取指定函数某一版本的异步配置信息。

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/async-invoke-config



表 7-749 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

## 请求参数

表 7-750 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-751 响应 Body 参数

| 参数                             | 参数类型                                              | 描述                               |
|--------------------------------|---------------------------------------------------|----------------------------------|
| func_urn                       | String                                            | 函数URN。<br>最小长度： 1<br>最大长度： 269   |
| max_async_event_age_in_seconds | Integer                                           | 消息最大存活时长，取值范围[1, 86400]。单位：秒。    |
| max_async_retry_attempts       | Integer                                           | 异步调用失败后的最大重试次数，默认值为3。取值范围[0, 8]。 |
| destination_config             | <a href="#">FuncAsyncDestinationConfig</a> object | 异步调用目标。                          |
| created_time                   | String                                            | 异步调用配置的创建时间。                     |
| last_modified                  | String                                            | 异步调用配置的最后更改时间。                   |

| 参数                      | 参数类型    | 描述          |
|-------------------------|---------|-------------|
| enable_async_status_log | Boolean | 开启异步调用状态持久化 |

表 7-752 FuncAsyncDestinationConfig

| 参数         | 参数类型                                         | 描述                                     |
|------------|----------------------------------------------|----------------------------------------|
| on_success | <a href="#">FuncDestinationConfig</a> object | 当函数执行成功时，函数计算将调用该配置对应的目标。              |
| on_failure | <a href="#">FuncDestinationConfig</a> object | 当函数执行失败（系统错误或函数内部错误）时，函数计算将调用该配置对应的目标。 |

表 7-753 FuncDestinationConfig

| 参数          | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                |
|-------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| destination | String | 目标类型。<br><ul style="list-style-type: none"> <li>● OBS：通知到OBS服务。</li> <li>● SMN：通知到SMN服务。</li> <li>● DIS：通知到DIS服务。</li> <li>● FunctionGraph：通知到函数服务。</li> </ul> 枚举值：<br><ul style="list-style-type: none"> <li>● <b>OBS</b></li> <li>● <b>SMN</b></li> <li>● <b>DIS</b></li> <li>● <b>FunctionGraph</b></li> </ul> |
| param       | String | 通知目标服务对应参数,json字符串。<br><ul style="list-style-type: none"> <li>● OBS：包含bucket桶，对象目录前缀prefix，对象默认expires过期时间[0~365]天，0默认不过期。</li> <li>● SMN：包含smn 主题topic_urn。</li> <li>● DIS：包含DIS 通道名stream_name。</li> <li>● FunctionGraph：包含func_urn，函数urn</li> </ul>                                                            |

状态码： 404

表 7-754 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

获取函数异步配置信息。

```
GET /v2/{project_id}/fgs/functions/{function_urn}/async-invoke-config
```

## 响应示例

**状态码： 200**

Ok

```
{
 "func_urn" : "urn:fss:xxxxxxxx:7aad83af3e8d42e99ac194xxxxxxxx:function:default:test:latest",
 "max_async_event_age_in_seconds" : 60,
 "max_async_retry_attempts" : 1,
 "destination_config" : {
 "on_success" : {
 "destination" : "FunctionGraph",
 "param" : "{\\"func_urn\\":\\"urn:fss:
{region}:5691ba790e2b46ceb38316xxxxxxxx:function:default:testPython:latest\\"}"
 },
 "on_failure" : {
 "destination" : "FunctionGraph",
 "param" : "{\\"func_urn\\":\\"urn:fss:
{region}:5691ba790e2b46ceb38316xxxxxxxx:function:default:testPython:latest\\"}"
 }
 },
 "created_time" : "2021-03-04T14:50:02+08:00",
 "last_modified" : "2021-03-04 14:50:02"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowFunctionAsyncInvokeConfigSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
```

security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

// In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
ShowFunctionAsyncInvokeConfigRequest request = new ShowFunctionAsyncInvokeConfigRequest();
try {
 ShowFunctionAsyncInvokeConfigResponse response =
client.showFunctionAsyncInvokeConfig(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

```
coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *
```

```
if __name__ == "__main__":
```

```
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.
```

```
 # In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
```

```
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")
```

```
 credentials = BasicCredentials(ak, sk) \
```

```
 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()
```

```
 try:
```

```
 request = ShowFunctionAsyncInvokeConfigRequest()
 response = client.show_function_async_invoke_config(request)
 print(response)
```

```
 except exceptions.ClientRequestException as e:
```

```
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowFunctionAsyncInvokeConfigRequest{}
 response, err := client.ShowFunctionAsyncInvokeConfig(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述        |
|-----|-----------|
| 200 | Ok        |
| 404 | Not Found |

## 错误码

请参见[错误码](#)。

## 7.15.2 删除函数异步配置信息

### 功能介绍

删除函数异步配置信息。

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/fgs/functions/{function\_urn}/async-invoke-config

表 7-755 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见 FunctionGraph函数模型的描述。       |

### 请求参数

表 7-756 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                            |
|--------------|------|--------|---------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中 X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                    |

### 响应参数

状态码： 404

表 7-757 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

删除函数异步配置信息。

```
DELETE /v2/{project_id}/fgs/functions/{function_urn}/async-invoke-config
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class DeleteFunctionAsyncInvokeConfigSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 DeleteFunctionAsyncInvokeConfigRequest request = new DeleteFunctionAsyncInvokeConfigRequest();
 try {
 DeleteFunctionAsyncInvokeConfigResponse response =
 client.deleteFunctionAsyncInvokeConfig(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

```
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = DeleteFunctionAsyncInvokeConfigRequest()
 response = client.delete_function_async_invoke_config(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())
```



```
request := &model.DeleteFunctionAsyncInvokeConfigRequest{}
response, err := client.DeleteFunctionAsyncInvokeConfig(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述         |
|-----|------------|
| 204 | No Content |
| 404 | Not Found  |

## 错误码

请参见[错误码](#)。

## 7.15.3 设置函数异步配置信息

### 功能介绍

设置函数异步配置信息。

### 调用方法

请参见[如何调用API](#)。

### URI

PUT /v2/{project\_id}/fgs/functions/{function\_urn}/async-invoke-config

表 7-758 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

## 请求参数

表 7-759 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

表 7-760 请求 Body 参数

| 参数                             | 是否必选 | 参数类型                              | 描述                                                           |
|--------------------------------|------|-----------------------------------|--------------------------------------------------------------|
| max_async_event_age_in_seconds | 否    | Integer                           | 消息最大存活时长，取值范围[1, 86400]，单位：秒，默认值为3600。<br>最小值：1<br>最大值：86400 |
| max_async_retry_attempts       | 否    | Integer                           | 异步调用失败后的最大重试次数，默认值为1。取值范围[0, 3]。<br>最小值：0<br>最大值：3           |
| destination_config             | 否    | FuncAsyncDestinationConfig object | 异步调用目标。                                                      |
| enable_async_status_log        | 否    | Boolean                           | 开启异步调用状态持久化                                                  |

表 7-761 FuncAsyncDestinationConfig

| 参数         | 是否必选 | 参数类型                         | 描述                                     |
|------------|------|------------------------------|----------------------------------------|
| on_success | 否    | FuncDestinationConfig object | 当函数执行成功时，函数计算将调用该配置对应的目标。              |
| on_failure | 否    | FuncDestinationConfig object | 当函数执行失败（系统错误或函数内部错误）时，函数计算将调用该配置对应的目标。 |

表 7-762 FuncDestinationConfig

| 参数          | 是否必选 | 参数类型   | 描述                                                                                                                                                                                                                                                                                                                            |
|-------------|------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| destination | 否    | String | <p>目标类型。</p> <ul style="list-style-type: none"> <li>• OBS: 通知到OBS服务。</li> <li>• SMN: 通知到SMN服务。</li> <li>• DIS: 通知到DIS服务。</li> <li>• FunctionGraph: 通知到函数服务。</li> </ul> <p>枚举值:</p> <ul style="list-style-type: none"> <li>• <b>OBS</b></li> <li>• <b>SMN</b></li> <li>• <b>DIS</b></li> <li>• <b>FunctionGraph</b></li> </ul> |
| param       | 否    | String | <p>通知目标服务对应参数,json字符串。</p> <ul style="list-style-type: none"> <li>• OBS: 包含bucket桶, 对象目录前缀prefix, 对象默认expires过期时间[0~365]天, 0默认不过期。</li> <li>• SMN: 包含smn 主题topic_urn。</li> <li>• DIS: 包含DIS 通道名stream_name。</li> <li>• FunctionGraph: 包含func_urn, 函数urn</li> </ul>                                                            |

## 响应参数

状态码: 200

表 7-763 响应 Body 参数

| 参数                             | 参数类型    | 描述                                            |
|--------------------------------|---------|-----------------------------------------------|
| func_urn                       | String  | <p>函数URN。</p> <p>最小长度: 1</p> <p>最大长度: 269</p> |
| max_async_event_age_in_seconds | Integer | <p>消息最大存活时长, 取值范围[60, 86400]。单位: 秒。</p>       |
| max_async_retry_attempts       | Integer | <p>异步调用失败后的最大重试次数, 默认值为3。取值范围[0, 8]。</p>      |

| 参数                      | 参数类型                                     | 描述             |
|-------------------------|------------------------------------------|----------------|
| destination_config      | <b>FuncAsyncDestinationConfig</b> object | 异步调用目标。        |
| created_time            | String                                   | 异步调用配置的创建时间。   |
| last_modified           | String                                   | 异步调用配置的最后更改时间。 |
| enable_async_status_log | Boolean                                  | 开启异步调用状态持久化    |

表 7-764 FuncAsyncDestinationConfig

| 参数         | 参数类型                                | 描述                                     |
|------------|-------------------------------------|----------------------------------------|
| on_success | <b>FuncDestinationConfig</b> object | 当函数执行成功时，函数计算将调用该配置对应的目标。              |
| on_failure | <b>FuncDestinationConfig</b> object | 当函数执行失败（系统错误或函数内部错误）时，函数计算将调用该配置对应的目标。 |

表 7-765 FuncDestinationConfig

| 参数          | 参数类型   | 描述                                                                                                                                                                                                                                                                                                |
|-------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| destination | String | 目标类型。 <ul style="list-style-type: none"><li>• OBS：通知到OBS服务。</li><li>• SMN：通知到SMN服务。</li><li>• DIS：通知到DIS服务。</li><li>• FunctionGraph：通知到函数服务。</li></ul> 枚举值： <ul style="list-style-type: none"><li>• <b>OBS</b></li><li>• <b>SMN</b></li><li>• <b>DIS</b></li><li>• <b>FunctionGraph</b></li></ul> |

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                                             |
|-------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| param | String | 通知目标服务对应参数,json字符串。<br><ul style="list-style-type: none"> <li>• OBS: 包含bucket桶, 对象目录前缀prefix, 对象默认expires过期时间[0~365]天, 0默认不过期。</li> <li>• SMN: 包含smn 主题topic_urn。</li> <li>• DIS: 包含DIS 通道名stream_name。</li> <li>• FunctionGraph: 包含func_urn, 函数urn</li> </ul> |

状态码: 400

表 7-766 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码: 404

表 7-767 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

设置函数异步配置信息, 消息最大存活时间设置为10s, 最大重试次数为3次, 开启异步调用状态持久化。

```
PUT /v2/{project_id}/fgs/functions/{function_urn}/async-invoke-config
{
 "max_async_event_age_in_seconds": 10,
 "max_async_retry_attempts": 3,
 "enable_async_status_log": true
}
```

## 响应示例

状态码: 200

Ok

```
{
 "func_urn": "urn:fss:xxxxxxxx:7aad83af3e8d42e99ac194xxxxxxxx:function:default:test:latest",
```

```
"max_async_event_age_in_seconds" : 60,
"max_async_retry_attempts" : 1,
"destination_config" : {
 "on_success" : {
 "destination" : "FunctionGraph",
 "param" : "{\"func_urn\":\"urn:fss:
{region}:5691ba790e2b46ceb38316xxxxxxxx:funcion:default:testPython:latest\"}"
 },
 "on_failure" : {
 "destination" : "FunctionGraph",
 "param" : "{\"func_urn\":\"urn:fss:
{region}:5691ba790e2b46ceb38316xxxxxxxx:funcion:default:testPython:latest\"}"
 }
},
"created_time" : "2021-03-04T14:50:02+08:00",
"last_modified" : "2021-03-04 14:50:02"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

设置函数异步配置信息，消息最大存活时间设置为10s，最大重试次数为3次，开启异步调用状态持久化。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class UpdateFunctionAsyncInvokeConfigSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 UpdateFunctionAsyncInvokeConfigRequest request = new UpdateFunctionAsyncInvokeConfigRequest();
 UpdateFunctionAsyncInvokeConfigRequestBody body = new
 UpdateFunctionAsyncInvokeConfigRequestBody();
 body.withEnableAsyncStatusLog(true);
 body.withMaxAsyncRetryAttempts(3);
 body.withMaxAsyncEventAgeInSeconds(10);
 request.withBody(body);
 try {
 UpdateFunctionAsyncInvokeConfigResponse response =
```

```
client.updateFunctionAsyncInvokeConfig(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
}
```

## Python

设置函数异步配置信息，消息最大存活时间设置为10s，最大重试次数为3次，开启异步调用状态持久化。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = UpdateFunctionAsyncInvokeConfigRequest()
 request.body = UpdateFunctionAsyncInvokeConfigRequestBody(
 enable_async_status_log=True,
 max_async_retry_attempts=3,
 max_async_event_age_in_seconds=10
)
 response = client.update_function_async_invoke_config(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

设置函数异步配置信息，消息最大存活时间设置为10s，最大重试次数为3次，开启异步调用状态持久化。

```
package main

import (
```

```
"fmt"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.UpdateFunctionAsyncInvokeConfigRequest{
 enableAsyncStatusLogUpdateFunctionAsyncInvokeConfigRequestBody:= true
 maxAsyncRetryAttemptsUpdateFunctionAsyncInvokeConfigRequestBody:= int32(3)
 maxAsyncEventAgeInSecondsUpdateFunctionAsyncInvokeConfigRequestBody:= int32(10)
 request.Body = &model.UpdateFunctionAsyncInvokeConfigRequestBody{
 EnableAsyncStatusLog: &enableAsyncStatusLogUpdateFunctionAsyncInvokeConfigRequestBody,
 MaxAsyncRetryAttempts: &maxAsyncRetryAttemptsUpdateFunctionAsyncInvokeConfigRequestBody,
 MaxAsyncEventAgeInSeconds:
 &maxAsyncEventAgeInSecondsUpdateFunctionAsyncInvokeConfigRequestBody,
 }
 }
 response, err := client.UpdateFunctionAsyncInvokeConfig(request)
 if err == nil {
 fmt.Printf("%v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述          |
|-----|-------------|
| 200 | Ok          |
| 400 | Bad Request |
| 404 | Not Found   |

## 错误码

请参见[错误码](#)。



## 7.15.4 获取函数异步配置列表

### 功能介绍

获取指定函数所有版本的异步配置列表。。

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/async-invoke-configs

表 7-768 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见 FunctionGraph函数模型的描述。       |

表 7-769 Query 参数

| 参数     | 是否必选 | 参数类型   | 描述                                                                                                                                               |
|--------|------|--------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| marker | 否    | String | 上一次查询到的最后的记录位置。<br>最大长度： <b>64</b>                                                                                                               |
| limit  | 否    | String | 每页显示的条目数量。 <ul style="list-style-type: none"><li>如果不提供该值或者提供的值等于0，则使用默认值：10，最大值100，大于100取值100。</li><li>如果该值小于0，则返回参数错误。</li></ul> 缺省值： <b>10</b> |

## 请求参数

表 7-770 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-771 响应 Body 参数

| 参数                   | 参数类型                                                                 | 描述              |
|----------------------|----------------------------------------------------------------------|-----------------|
| async_invoke_configs | Array of <a href="#">ListFunctionAsyncInvokeConfigResult</a> objects | 函数异步配置列表。       |
| count                | Long                                                                 | 列表总数。<br>最小值： 0 |
| page_info            | <a href="#">PageInfo</a> object                                      |                 |

表 7-772 ListFunctionAsyncInvokeConfigResult

| 参数                             | 参数类型    | 描述                               |
|--------------------------------|---------|----------------------------------|
| func_urn                       | String  | 函数URN。<br>最小长度： 1<br>最大长度： 269   |
| max_async_event_age_in_seconds | Integer | 消息最大存活时长，取值范围[60, 86400]。单位：秒。   |
| max_async_retry_attempts       | Integer | 异步调用失败后的最大重试次数，默认值为3。取值范围[0, 8]。 |

| 参数                      | 参数类型                                     | 描述             |
|-------------------------|------------------------------------------|----------------|
| destination_config      | <b>FuncAsyncDestinationConfig</b> object | 异步调用目标。        |
| created_time            | String                                   | 异步调用配置的创建时间。   |
| last_modified           | String                                   | 异步调用配置的最后更改时间。 |
| enable_async_status_log | Boolean                                  | 开启异步调用状态持久化    |

表 7-773 FuncAsyncDestinationConfig

| 参数         | 参数类型                                | 描述                                     |
|------------|-------------------------------------|----------------------------------------|
| on_success | <b>FuncDestinationConfig</b> object | 当函数执行成功时，函数计算将调用该配置对应的目标。              |
| on_failure | <b>FuncDestinationConfig</b> object | 当函数执行失败（系统错误或函数内部错误）时，函数计算将调用该配置对应的目标。 |

表 7-774 FuncDestinationConfig

| 参数          | 参数类型   | 描述                                                                                                                                                                                                                                                                                                |
|-------------|--------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| destination | String | 目标类型。 <ul style="list-style-type: none"><li>● OBS：通知到OBS服务。</li><li>● SMN：通知到SMN服务。</li><li>● DIS：通知到DIS服务。</li><li>● FunctionGraph：通知到函数服务。</li></ul> 枚举值： <ul style="list-style-type: none"><li>● <b>OBS</b></li><li>● <b>SMN</b></li><li>● <b>DIS</b></li><li>● <b>FunctionGraph</b></li></ul> |

| 参数    | 参数类型   | 描述                                                                                                                                                                                                                                                             |
|-------|--------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| param | String | 通知目标服务对应参数,json字符串。<br><ul style="list-style-type: none"> <li>• OBS: 包含bucket桶, 对象目录前缀prefix, 对象默认expires过期时间[0~365]天, 0默认不过期。</li> <li>• SMN: 包含smn 主题topic_urn。</li> <li>• DIS: 包含DIS 通道名stream_name。</li> <li>• FunctionGraph: 包含func_urn, 函数urn</li> </ul> |

表 7-775 PageInfo

| 参数                  | 参数类型 | 描述      |
|---------------------|------|---------|
| next_marker         | Long | 下一次读取位置 |
| previous_mar<br>ker | Long | 上一次读取位置 |
| current_count       | Long | 当前页总数   |

## 请求示例

获取函数异步配置列表。

```
GET /v2/{project_id}/fgs/functions/{function_urn}/async-invoke-configs
```

## 响应示例

状态码: 200

Ok

```
{
 "async_invoke_configs": [{
 "func_urn": "urn:fss:{region}:5691ba790e2b46ceb38316xxxxxxxx:function:default:test_async:latest",
 "max_async_event_age_in_seconds": 60,
 "max_async_retry_attempts": 1,
 "destination_config": {
 "on_success": {
 "destination": "FunctionGraph",
 "param": "{\"func_urn\":\"urn:fss:
{region}:5691ba790e2b46ceb38316xxxxxxxx:function:default:testPython:latest\"}"
 },
 "on_failure": {
 "destination": "FunctionGraph",
 "param": "{\"func_urn\":\"urn:fss:
{region}:5691ba790e2b46ceb38316xxxxxxxx:function:default:testPython:latest\"}"
 }
 },
 "created_time": "2021-03-04T14:50:02+08:00",
 "last_modified": "2021-03-04 14:50:02"
 }],
 "count": 2,
 "page_info": {
```

```
"next_marker" : 2,
"previous_marker" : 0,
"current_count" : 2
}
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListFunctionAsyncInvokeConfigSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListFunctionAsyncInvokeConfigRequest request = new ListFunctionAsyncInvokeConfigRequest();
 request.withMarker("<marker>");
 request.withLimit("<limit>");
 try {
 ListFunctionAsyncInvokeConfigResponse response = client.listFunctionAsyncInvokeConfig(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8
```

```
from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListFunctionAsyncInvokeConfigRequest()
 request.marker = "<marker>"
 request.limit = "<limit>"
 response = client.list_function_async_invoke_config(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListFunctionAsyncInvokeConfigRequest{}
 markerRequest := "<marker>"
 request.Marker = &markerRequest
 limitRequest := "<limit>"
```

```
request.Limit = &limitRequest
response, err := client.ListFunctionAsyncInvokeConfig(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述 |
|-----|----|
| 200 | Ok |

## 错误码

请参见[错误码](#)。

## 7.15.5 获取函数异步调用请求列表

### 功能介绍

获取函数异步调用请求列表

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/async-inocations

表 7-776 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

表 7-777 Query 参数

| 参数               | 是否必选 | 参数类型   | 描述                                                                                                                        |
|------------------|------|--------|---------------------------------------------------------------------------------------------------------------------------|
| request_id       | 否    | String | 需要查询的异步请求ID。如果不指定，默认查询所有异步调用记录<br>最小长度：0<br>最大长度：64                                                                       |
| marker           | 否    | String | 本次查询起始位置，默认值0<br>缺省值：0<br>最小长度：0<br>最大长度：64                                                                               |
| limit            | 否    | String | 本次查询最大返回的数据条数，最大值500，默认值100<br>缺省值：100<br>最小长度：0<br>最大长度：64                                                               |
| status           | 否    | String | 本次查询指定的异步调用状态，支持5种状态，如果不指定，则查询所有状态的调用记录 WAIT: 等待 RUNNING: 执行中 SUCCESS: 执行成功 FAIL: 执行失败 DISCARD: 请求丢弃<br>最小长度：0<br>最大长度：64 |
| query_begin_time | 否    | String | 搜索起始时间（格式为YYYY-MM-DD'T'HH:mm:ss,UTC时间）。如果不指定默认为当前时间前1小时                                                                   |
| query_end_time   | 否    | String | 搜索结束时间（格式为YYYY-MM-DD'T'HH:mm:ss,UTC时间）。如果不指定默认为当前时间                                                                       |



## 请求参数

表 7-778 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-779 响应 Body 参数

| 参数          | 参数类型                                                       | 描述         |
|-------------|------------------------------------------------------------|------------|
| invocations | Array of <b>ListFunctionAsyncInvocationsResult</b> objects | 异步调用记录列表。  |
| count       | Integer                                                    | 查询数据总条数    |
| next_marker | Integer                                                    | 查询下一页的起始位置 |

表 7-780 ListFunctionAsyncInvocationsResult

| 参数         | 参数类型   | 描述                                                                                                                                                                                                                      |
|------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| request_id | String | 异步调用请求ID                                                                                                                                                                                                                |
| status     | String | 异步调用状态，支持5种状态 WAIT: 等待 RUNNING: 执行中 SUCCESS: 执行成功 FAIL: 执行失败 DISCARD: 请求丢弃<br>枚举值：<br><ul style="list-style-type: none"> <li>● WAIT</li> <li>● RUNNING</li> <li>● SUCCESS</li> <li>● FAIL</li> <li>● DISCARD</li> </ul> |

| 参数            | 参数类型    | 描述                                        |
|---------------|---------|-------------------------------------------|
| error_message | String  | 异步调用错误信息，如果执行成功，则返回空                      |
| error_code    | Integer | 异步调用错误码，如果执行成功，则返回0                       |
| start_time    | String  | 异步调用开始时间（格式为YYYY-MM-DD'T'HH:mm:ss,UTC时间）。 |
| end_time      | String  | 异步调用结束时间（格式为YYYY-MM-DD'T'HH:mm:ss,UTC时间）。 |

**状态码： 403**

**表 7-781 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 404**

**表 7-782 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码： 500**

**表 7-783 响应 Body 参数**

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

获取函数异步调用请求列表。

```
GET /v2/{project_id}/fgs/functions/{function_urn}/async-invocations
```

## 响应示例

### 状态码： 200

OK

```
{
 "invocations": [{
 "request_id": "403fcbd6-ec41-401f-9fa7-386f3d3d****",
 "status": "SUCCESS",
 "error_message": "",
 "start_time": "2019-10-25T15:37:27",
 "end_time": "2019-10-25T15:37:27",
 "error_code": 0
 }]
}
```

### 状态码： 403

FORBIDDEN

```
{
 "error_code": "FSS.0403",
 "error_msg": "invalid token"
}
```

### 状态码： 404

Not Found

```
{
 "error_code": "FSS.0404",
 "error_msg": "can not find function"
}
```

### 状态码： 500

内部错误

```
{
 "error_code": "FSS.0500",
 "error_msg": "xxx"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.time.OffsetDateTime;
import java.time.format.DateTimeFormatter;

public class ListAsyncInvocationsSolution {

 public static void main(String[] args) {
```

```
// The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
environment variables and decrypted during use to ensure security.
// In this example, AK and SK are stored in environment variables for authentication. Before running
this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
String ak = System.getenv("CLOUD_SDK_AK");
String sk = System.getenv("CLOUD_SDK_SK");

ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();

ListAsyncInvocationsRequest request = new ListAsyncInvocationsRequest();
request.setRequestId("<request_id>");
request.withMarker("<marker>");
request.withLimit("<limit>");
request.withStatus("<status>");
request.withQueryBeginTime(OffsetDateTime.parse("<query_begin_time>",
DateTimeFormatter.ISO_OFFSET_DATE_TIME));
request.withQueryEndTime(OffsetDateTime.parse("<query_end_time>",
DateTimeFormatter.ISO_OFFSET_DATE_TIME));
try {
 ListAsyncInvocationsResponse response = client.listAsyncInvocations(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListAsyncInvocationsRequest()
```

```
request.request_id = "<request_id>"
request.marker = "<marker>"
request.limit = "<limit>"
request.status = "<status>"
request.query_begin_time = "<query_begin_time>"
request.query_end_time = "<query_end_time>"
response = client.list_async_invocations(request)
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "time"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/sdktime"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListAsyncInvocationsRequest{}
 requestIdRequest := "<request_id>"
 request.RequestId = &requestIdRequest
 markerRequest := "<marker>"
 request.Marker = &markerRequest
 limitRequest := "<limit>"
 request.Limit = &limitRequest
 statusRequest := "<status>"
 request.Status = &statusRequest
 dateTimeQueryBeginTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "<query_begin_time>",
 time.UTC)
 queryBeginTimeRequest := sdktime.SdkTime(dateTimeQueryBeginTime)
 request.QueryBeginTime = &queryBeginTimeRequest
 dateTimeQueryEndTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "<query_end_time>",
 time.UTC)
 queryEndTimeRequest := sdktime.SdkTime(dateTimeQueryEndTime)
 request.QueryEndTime = &queryEndTimeRequest
 response, err := client.ListAsyncInvocations(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
```

```
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述        |
|-----|-----------|
| 200 | OK        |
| 403 | FORBIDDEN |
| 404 | Not Found |
| 500 | 内部错误      |

## 错误码

请参见[错误码](#)。

## 7.15.6 获取函数活跃异步调用请求列表

### 功能介绍

获取函数异步调用活跃请求列表

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/functions/{function\_urn}/active-async-invocations

表 7-784 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

表 7-785 Query 参数

| 参数               | 是否必选 | 参数类型   | 描述                                                                                                                             |
|------------------|------|--------|--------------------------------------------------------------------------------------------------------------------------------|
| requests         | 否    | String | 需要查询的异步请求ID, 多个请求id使用','分隔, 最大支持10个请求id查询。如果不指定, 默认查询所有异步调用记录<br>最小长度: 0<br>最大长度: 640                                          |
| marker           | 否    | String | 本次查询起始位置, 默认值0<br>缺省值: 0<br>最小长度: 0<br>最大长度: 64                                                                                |
| limit            | 否    | String | 本次查询最大返回的数据条数, 最大值500, 默认值100<br>缺省值: 100<br>最小长度: 0<br>最大长度: 64                                                               |
| status           | 否    | String | 本次查询指定的异步调用状态, 支持5种状态, 如果不指定, 则查询所有状态的调用记录 WAIT: 等待 RUNNING: 执行中 SUCCESS: 执行成功 FAIL: 执行失败 DISCARD: 请求丢弃<br>最小长度: 0<br>最大长度: 64 |
| query_begin_time | 否    | String | 搜索起始时间 (格式为YYYY-MM-DD'T'HH:mm:ss,UTC时间)。如果不指定默认为当前时间前1小时                                                                       |
| query_end_time   | 否    | String | 搜索结束时间 (格式为YYYY-MM-DD'T'HH:mm:ss,UTC时间)。如果不指定默认为当前时间                                                                           |

## 请求参数

表 7-786 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-787 响应 Body 参数

| 参数          | 参数类型                                                       | 描述         |
|-------------|------------------------------------------------------------|------------|
| invocations | Array of <b>ListFunctionAsyncInvocationsResult</b> objects | 异步调用记录列表。  |
| count       | Integer                                                    | 查询数据总条数    |
| next_marker | Integer                                                    | 查询下一页的起始位置 |

表 7-788 ListFunctionAsyncInvocationsResult

| 参数         | 参数类型   | 描述                                                                                                                                                                                                                      |
|------------|--------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| request_id | String | 异步调用请求ID                                                                                                                                                                                                                |
| status     | String | 异步调用状态，支持5种状态 WAIT: 等待 RUNNING: 执行中 SUCCESS: 执行成功 FAIL: 执行失败 DISCARD: 请求丢弃<br>枚举值：<br><ul style="list-style-type: none"> <li>● WAIT</li> <li>● RUNNING</li> <li>● SUCCESS</li> <li>● FAIL</li> <li>● DISCARD</li> </ul> |



| 参数            | 参数类型    | 描述                                        |
|---------------|---------|-------------------------------------------|
| error_message | String  | 异步调用错误信息，如果执行成功，则返回空                      |
| error_code    | Integer | 异步调用错误码，如果执行成功，则返回0                       |
| start_time    | String  | 异步调用开始时间（格式为YYYY-MM-DD'T'HH:mm:ss,UTC时间）。 |
| end_time      | String  | 异步调用结束时间（格式为YYYY-MM-DD'T'HH:mm:ss,UTC时间）。 |

状态码： 403

表 7-789 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-790 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-791 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

获取函数活跃异步调用请求列表。

```
GET /v2/{project_id}/fgs/functions/{function_urn}/active-async-invocations
```

## 响应示例

### 状态码： 200

OK

```
{
 "invocations": [{
 "request_id": "403fcbd6-ec41-401f-9fa7-386f3d3d****",
 "status": "SUCCESS",
 "error_message": "",
 "start_time": "2019-10-25T15:37:27",
 "end_time": "2019-10-25T15:37:27",
 "error_code": 0
 }],
 "next_marker": 1,
 "count": 10
}
```

### 状态码： 403

FORBIDDEN

```
{
 "error_code": "FSS.0403",
 "error_msg": "invalid token"
}
```

### 状态码： 404

Not Found

```
{
 "error_code": "FSS.0404",
 "error_msg": "can not find function"
}
```

### 状态码： 500

内部错误

```
{
 "error_code": "FSS.0500",
 "error_msg": "xxx"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.time.OffsetDateTime;
import java.time.format.DateTimeFormatter;

public class ListActiveAsyncInvocationsSolution {
```

```
public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListActiveAsyncInvocationsRequest request = new ListActiveAsyncInvocationsRequest();
 request.withRequests("<requests>");
 request.withMarker("<marker>");
 request.withLimit("<limit>");
 request.withStatus("<status>");
 request.withQueryBeginTime(OffsetDateTime.parse("<query_begin_time>",
 DateTimeFormatter.ISO_OFFSET_DATE_TIME));
 request.withQueryEndTime(OffsetDateTime.parse("<query_end_time>",
 DateTimeFormatter.ISO_OFFSET_DATE_TIME));
 try {
 ListActiveAsyncInvocationsResponse response = client.listActiveAsyncInvocations(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()
```

```
try:
 request = ListActiveAsyncInvocationsRequest()
 request.requests = "<requests>"
 request.marker = "<marker>"
 request.limit = "<limit>"
 request.status = "<status>"
 request.query_begin_time = "<query_begin_time>"
 request.query_end_time = "<query_end_time>"
 response = client.list_active_async_invocations(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "time"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/sdktime"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListActiveAsyncInvocationsRequest{}
 requestsRequest := "<requests>"
 request.Requests = &requestsRequest
 markerRequest := "<marker>"
 request.Marker = &markerRequest
 limitRequest := "<limit>"
 request.Limit = &limitRequest
 statusRequest := "<status>"
 request.Status = &statusRequest
 dateTimeQueryBeginTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "<query_begin_time>",
 time.UTC)
 queryBeginTimeRequest := sdktime.SdkTime(dateTimeQueryBeginTime)
 request.QueryBeginTime = &queryBeginTimeRequest
 dateTimeQueryEndTime, _ := time.ParseInLocation("2006-01-02T15:04:05Z", "<query_end_time>",
 time.UTC)
 queryEndTimeRequest := sdktime.SdkTime(dateTimeQueryEndTime)
 request.QueryEndTime = &queryEndTimeRequest
 response, err := client.ListActiveAsyncInvocations(request)
 if err == nil {
```

```
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述        |
|-----|-----------|
| 200 | OK        |
| 403 | FORBIDDEN |
| 404 | Not Found |
| 500 | 内部错误      |

## 错误码

请参见[错误码](#)。

## 7.15.7 停止函数异步调用请求

### 功能介绍

当前仅支持参数recursive为false且force为true的函数。在1:N的函数做并发异步调用的场景下调用停止异步请求接口时，同一函数实例同时在执行的其他请求也会被一并停止并返回4208 function invocation canceled

### 调用方法

请参见[如何调用API](#)。

### URI

POST /v2/{project\_id}/fgs/functions/{function\_urn}/cancel

表 7-792 路径参数

| 参数           | 是否必选 | 参数类型   | 描述                                       |
|--------------|------|--------|------------------------------------------|
| project_id   | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| function_urn | 是    | String | 函数的URN，详细解释见FunctionGraph函数模型的描述。        |

## 请求参数

表 7-793 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

表 7-794 请求 Body 参数

| 参数         | 是否必选 | 参数类型   | 描述                                                                                                                                                        |
|------------|------|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------|
| request_id | 是    | String | 被停止的请求id                                                                                                                                                  |
| type       | 否    | String | 停止的类型 支持recursive, force。recursive: 停止正在调用的子函数。force: 直接终止runtime。<br>枚举值： <ul style="list-style-type: none"> <li>• force</li> <li>• recursive</li> </ul> |

## 响应参数

状态码： 400

表 7-795 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-796 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-797 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-798 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

停止函数异步调用请求。

```
POST /v2/{project_id}/fgs/functions/{function_urn}/cancel
{
 "request_id": "xxxx"
}
```

## 响应示例

状态码： 200

OK

```
null
```

状态码： 400

Invalid RequestId

```
{
 "error_code": "FSS.0400",
 "error_msg": "Invalid RequestId"
}
```

**状态码： 403**

FORBIDDEN

```
{
 "error_code" : "FSS.0403",
 "error_msg" : "invalid token"
}
```

**状态码： 404**

Not Found

```
{
 "error_code" : "FSS.0404",
 "error_msg" : "can not find function"
}
```

**状态码： 500**

内部错误

```
{
 "error_code" : "FSS.0500",
 "error_msg" : "xxx"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

停止函数异步调用请求。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class CancelAsyncInvocationSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 CancelAsyncInvocationRequest request = new CancelAsyncInvocationRequest();
```



```
CancelAsyncInvocationRequestBody body = new CancelAsyncInvocationRequestBody();
body.withRequestId("xxxx");
request.withBody(body);
try {
 CancelAsyncInvocationResponse response = client.cancelAsyncInvocation(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

停止函数异步调用请求。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CancelAsyncInvocationRequest()
 request.body = CancelAsyncInvocationRequestBody(
 request_id="xxxx"
)
 response = client.cancel_async_invocation(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

停止函数异步调用请求。

```
package main

import (
 "fmt"
```

```
"github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
"github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.CancelAsyncInvocationRequest{}
 request.Body = &model.CancelAsyncInvocationRequestBody{
 RequestId: "xxxx",
 }
 response, err := client.CancelAsyncInvocation(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                |
|-----|-------------------|
| 200 | OK                |
| 400 | Invalid RequestId |
| 403 | FORBIDDEN         |
| 404 | Not Found         |
| 500 | 内部错误              |

## 错误码

请参见[错误码](#)。

## 7.16 函数应用中心

### 7.16.1 查询应用程序模板详情

#### 功能介绍

查询应用程序模板详情（该功能目前仅支持华东-上海一、华北-北京四）

#### 调用方法

请参见[如何调用API](#)。

#### URI

GET /v2/{project\_id}/fgs/application/templates/{id}

表 7-799 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| id         | 是    | String | 应用模板ID。<br>最小长度：1<br>最大长度：64             |

#### 请求参数

表 7-800 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

#### 响应参数

状态码：200

表 7-801 响应 Body 参数

| 参数             | 参数类型                                                       | 描述                  |
|----------------|------------------------------------------------------------|---------------------|
| name           | String                                                     | 模板名称                |
| description    | String                                                     | 模板描述                |
| runtime        | String                                                     | 模板执行运行时             |
| category       | String                                                     | 模板使用场景              |
| params         | String                                                     | 模板参数                |
| image          | String                                                     | 模板镜像文件 ( base64编码 ) |
| deploy_count   | Long                                                       | 模板部署次数              |
| version        | Long                                                       | 模板版本                |
| template_guide | String                                                     | 模板指南                |
| create_time    | Long                                                       | 模板创建时间              |
| update_time    | Long                                                       | 模板更新时间              |
| resources      | Array of <a href="#">AppTemplateResourceDetail</a> objects | 模板资源                |

表 7-802 AppTemplateResourceDetail

| 参数            | 参数类型   | 描述                |
|---------------|--------|-------------------|
| resource_name | String | 资源名称              |
| icon          | String | 图标文件 ( base64编码 ) |
| href          | String | 超链接地址             |

状态码： 400

表 7-803 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码：401****表 7-804** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码：403****表 7-805** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码：404****表 7-806** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**状态码：500****表 7-807** 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

**请求示例**

查询函数应用程序模板详情。

```
GET /v2/{project_id}/fgs/application/templates/{id}
```

## 响应示例

状态码： 200

get template successfully

```
{
 "name": "AI Drawing Stable-Diffusion",
 "description": "AI Drawing",
 "runtime": "http",
 "category": "AI",
 "params": "",
 "image": "iVBORw0KGgxxxxxx",
 "deploy_count": 0,
 "version": 20231103,
 "template_guide": "Stable Diffusion AI Drawing...",
 "create_time": 1699328406899,
 "update_time": 1699328406899,
 "resources": [{
 "resource_name": "Object Storage Service (OBS)",
 "icon": "iVBORw0KGgoAAAANSUHEUgAAAGAAAABgCAYAAADixxxx",
 "href": "/console/?region=%s&locale=zh-cn#/obs/manage/%s/object/list"
 }]
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowAppTemplateSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowAppTemplateRequest request = new ShowAppTemplateRequest();
 try {
 ShowAppTemplateResponse response = client.showAppTemplate(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
```

```
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
}
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowAppTemplateRequest()
 response = client.show_app_template(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")
```

```
auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

request := &model.ShowAppTemplateRequest{}
response, err := client.ShowAppTemplate(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | get template successfully     |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.16.2 查询应用程序模板列表

### 功能介绍

查询应用程序模板列表（该功能目前仅支持华东-上海一、华北-北京四）

### 调用方法

请参见[如何调用API](#)。



## URI

GET /v2/{project\_id}/fgs/application/templates

表 7-808 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

表 7-809 Query 参数

| 参数       | 是否必选 | 参数类型   | 描述                                                                                |
|----------|------|--------|-----------------------------------------------------------------------------------|
| marker   | 否    | String | 本次查询起始位置，默认值0<br>缺省值： <b>0</b><br>最小长度： <b>0</b><br>最大长度： <b>64</b>               |
| maxitems | 否    | String | 本次查询最大返回的数据条数，最大值500，默认值100<br>缺省值： <b>0</b><br>最小长度： <b>0</b><br>最大长度： <b>64</b> |
| runtime  | 否    | String | 模板执行运行时<br>缺省值： <b>all</b>                                                        |
| category | 否    | String | 模板类别<br>缺省值： <b>all</b>                                                           |

## 请求参数

表 7-810 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Language   | 否    | String | 模板语言<br>缺省值： <b>en-us</b>                                    |
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-811 响应 Body 参数

| 参数          | 参数类型                                                    | 描述         |
|-------------|---------------------------------------------------------|------------|
| templates   | Array of <a href="#">ListAppTemplatesResult</a> objects | 函数应用程序模板列表 |
| next_marker | Long                                                    | 下次读取位置     |
| count       | Long                                                    | 应用程序模板总数   |

表 7-812 ListAppTemplatesResult

| 参数          | 参数类型   | 描述                  |
|-------------|--------|---------------------|
| id          | String | 模板id                |
| name        | String | 模板名称                |
| runtime     | String | 模板执行运行时             |
| category    | String | 模板使用场景              |
| description | String | 模板描述                |
| image       | String | 模板镜像文件 ( base64编码 ) |

状态码： 400

表 7-813 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-814 响应 Body 参数

| 参数         | 参数类型   | 描述  |
|------------|--------|-----|
| error_code | String | 错误码 |

| 参数        | 参数类型   | 描述   |
|-----------|--------|------|
| error_msg | String | 错误信息 |

状态码： 403

表 7-815 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-816 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-817 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询函数应用程序模板列表。

```
GET /v2/{project_id}/fgs/application/templates
```

## 响应示例

状态码： 200

list templates successfully

```
{
 "templates": [{
 "id": "5351xxxx-xxxx-xxxx-xxxx-xxxxxxfa2d27",
```

```
"name" : "AI Drawing Stable-Diffusion",
"runtime" : "http",
"category" : "AI",
"description" : "AI Drawing",
"image" : "iVBORw0KGgxxxxx"
}],
"next_marker" : 11,
"count" : 11
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListAppTemplatesSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListAppTemplatesRequest request = new ListAppTemplatesRequest();
 request.withMarker("<marker>");
 request.withMaxItems("<maxitems>");
 request.withRuntime("<runtime>");
 request.withCategory("<category>");
 try {
 ListAppTemplatesResponse response = client.listAppTemplates(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ListAppTemplatesRequest()
 request.marker = "<marker>"
 request.maxitems = "<maxitems>"
 request.runtime = "<runtime>"
 request.category = "<category>"
 response = client.list_app_templates(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
```

```

Build()

request := &model.ListAppTemplatesRequest{}
markerRequest:= "<marker>"
request.Marker = &markerRequest
maxitemsRequest:= "<maxitems>"
request.Maxitems = &maxitemsRequest
runtimeRequest:= "<runtime>"
request.Runtime = &runtimeRequest
categoryRequest:= "<category>"
request.Category = &categoryRequest
response, err := client.ListAppTemplates(request)
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}

```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | list templates successfully   |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.16.3 查询应用程序列表

### 功能介绍

查询应用程序列表（该功能目前仅支持华东-上海一、华北-北京四）

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/applications

表 7-818 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

表 7-819 Query 参数

| 参数     | 是否必选 | 参数类型   | 描述                                                                                  |
|--------|------|--------|-------------------------------------------------------------------------------------|
| limit  | 否    | String | 本次查询最大返回的数据条数，最大值500，默认值100<br>缺省值： <b>100</b><br>最小长度： <b>1</b><br>最大长度： <b>64</b> |
| marker | 否    | String | 本次查询起始位置，默认值0<br>缺省值： <b>0</b><br>最小长度： <b>1</b><br>最大长度： <b>64</b>                 |

## 请求参数

表 7-820 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-821 响应 Body 参数

| 参数           | 参数类型                                                           | 描述     |
|--------------|----------------------------------------------------------------|--------|
| applications | Array of <a href="#">ListFunctionApplicationResult</a> objects | 函数应用列表 |
| next_marker  | Long                                                           | 下次读取位置 |
| count        | Long                                                           | 应用程序总数 |

表 7-822 ListFunctionApplicationResult

| 参数                 | 参数类型   | 描述     |
|--------------------|--------|--------|
| id                 | String | 应用id   |
| name               | String | 应用名称   |
| status             | String | 应用状态   |
| last_modified_time | Long   | 最后修改时间 |
| description        | String | 应用描述   |

状态码： 400

表 7-823 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-824 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403



表 7-825 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-826 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-827 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询函数应用程序列表。

```
GET /v2/{project_id}/fgs/applications
```

## 响应示例

状态码： 200

list applications successfully

```
{
 "applications": [{
 "id": "a2685d64-9bf6-481d-8845-0482271f99cc",
 "name": "testApplication",
 "status": "repoFail",
 "last_modified_time": 1699320954910,
 "description": "test function application"
 }],
 "next_marker": 1,
 "count": 1
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ListFunctionApplicationsSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ListFunctionApplicationsRequest request = new ListFunctionApplicationsRequest();
 request.withLimit("<limit>");
 request.withMarker("<marker>");
 try {
 ListFunctionApplicationsResponse response = client.listFunctionApplications(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

### Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
```

risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment variables and decrypted during use to ensure security.

# In this example, AK and SK are stored in environment variables for authentication. Before running this example, set environment variables CLOUD\_SDK\_AK and CLOUD\_SDK\_SK in the local environment

```
ak = __import__('os').getenv("CLOUD_SDK_AK")
sk = __import__('os').getenv("CLOUD_SDK_SK")

credentials = BasicCredentials(ak, sk) \

client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

try:
 request = ListFunctionApplicationsRequest()
 request.limit = "<limit>"
 request.marker = "<marker>"
 response = client.list_function_applications(request)
 print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ListFunctionApplicationsRequest{}
 limitRequest := "<limit>"
 request.Limit = &limitRequest
 markerRequest := "<marker>"
 request.Marker = &markerRequest
 response, err := client.ListFunctionApplications(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

```
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                             |
|-----|--------------------------------|
| 200 | list applications successfully |
| 400 | Bad Request 请求错误。              |
| 401 | Unauthorized 鉴权失败。             |
| 403 | Forbidden 没有操作权限。              |
| 404 | Not Found 找不到资源。               |
| 500 | Internal Server Error 服务内部错误。  |

## 错误码

请参见[错误码](#)。

## 7.16.4 创建应用程序

### 功能介绍

创建应用程序（该功能目前仅支持华东-上海一、华北-北京四）

### 调用方法

请参见[如何调用API](#)。

## URI

POST /v2/{project\_id}/fgs/applications

表 7-828 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |

## 请求参数

表 7-829 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

表 7-830 请求 Body 参数

| 参数                    | 是否必选 | 参数类型               | 描述        |
|-----------------------|------|--------------------|-----------|
| name                  | 否    | String             | 应用名称      |
| template_id           | 否    | String             | 应用使用的模板ID |
| description           | 否    | String             | 应用描述      |
| enterprise_project_id | 否    | String             | 企业项目ID    |
| agency_name           | 否    | String             | 委托名称      |
| params                | 否    | Map<String,String> | 模板参数      |

## 响应参数

状态码： 200

表 7-831 响应 Body 参数

| 参数             | 参数类型   | 描述   |
|----------------|--------|------|
| application_id | String | 应用id |

状态码： 400

表 7-832 响应 Body 参数

| 参数         | 参数类型   | 描述  |
|------------|--------|-----|
| error_code | String | 错误码 |

| 参数        | 参数类型   | 描述   |
|-----------|--------|------|
| error_msg | String | 错误信息 |

状态码： 401

表 7-833 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-834 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-835 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-836 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

创建函数应用程序。

```
POST /v2/{project_id}/fgs/applications
{
 "name" : "appTest",
 "template_id" : "xxx",
 "params" : { },
 "description" : "应用程序测试",
 "enterprise_project_id" : "0",
 "agency_name" : ""
}
```

## 响应示例

状态码： 200

create application successfully

```
{
 "application_id" : "cca14689-2a40-425a-86f8-14556a0fff9c"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

创建函数应用程序。

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

import java.util.Map;
import java.util.HashMap;

public class CreateFunctionAppSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 }
}
```

```
CreateFunctionAppRequest request = new CreateFunctionAppRequest();
CreateFunctionAppRequestBody body = new CreateFunctionAppRequestBody();
body.withAgencyName("");
body.withEnterpriseProjectId("0");
body.withDescription("应用程序测试");
body.withTemplateId("xxx");
body.withName("appTest");
request.withBody(body);
try {
 CreateFunctionAppResponse response = client.createFunctionApp(request);
 System.out.println(response.toString());
} catch (ConnectionException e) {
 e.printStackTrace();
} catch (RequestTimeoutException e) {
 e.printStackTrace();
} catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
}
}
```

## Python

创建函数应用程序。

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = CreateFunctionAppRequest()
 request.body = CreateFunctionAppRequestBody(
 agency_name="",
 enterprise_project_id="0",
 description="应用程序测试",
 template_id="xxx",
 name="appTest"
)
 response = client.create_function_app(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```



## Go

创建函数应用程序。

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.CreateFunctionAppRequest{}
 agencyNameCreateFunctionAppRequestBody := ""
 enterpriseProjectIdCreateFunctionAppRequestBody := "0"
 descriptionCreateFunctionAppRequestBody := "应用程序测试"
 templateIdCreateFunctionAppRequestBody := "xxx"
 nameCreateFunctionAppRequestBody := "appTest"
 request.Body = &model.CreateFunctionAppRequestBody{
 AgencyName: &agencyNameCreateFunctionAppRequestBody,
 EnterpriseProjectId: &enterpriseProjectIdCreateFunctionAppRequestBody,
 Description: &descriptionCreateFunctionAppRequestBody,
 TemplateId: &templateIdCreateFunctionAppRequestBody,
 Name: &nameCreateFunctionAppRequestBody,
 }
 response, err := client.CreateFunctionApp(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                              |
|-----|---------------------------------|
| 200 | create application successfully |

| 状态码 | 描述                            |
|-----|-------------------------------|
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.16.5 查询应用程序详情

### 功能介绍

查询应用程序详情（该功能目前仅支持华东-上海一、华北-北京四）

### 调用方法

请参见[如何调用API](#)。

### URI

GET /v2/{project\_id}/fgs/applications/{id}

表 7-837 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |
| id         | 是    | String | 应用ID。<br>最小长度：1<br>最大长度：64               |

## 请求参数

表 7-838 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码： 200

表 7-839 响应 Body 参数

| 参数                 | 参数类型                                 | 描述     |
|--------------------|--------------------------------------|--------|
| name               | String                               | 应用名称   |
| last_modified_time | Long                                 | 最后修改时间 |
| stack_resources    | <a href="#">StackResource object</a> | 资源栈    |
| status             | String                               | 应用状态   |
| stack_name         | String                               | 资源栈名称  |
| stack_id           | String                               | 资源栈id  |
| repo_name          | String                               | 存储库名称  |
| description        | String                               | 应用描述   |
| repo               | <a href="#">RepoInfo object</a>      | 存储库信息  |
| pipeline_id        | String                               | 管道id   |
| project_id         | String                               | 项目id   |
| apig_url           | String                               | 调用URL  |

表 7-840 StackResource

| 参数                     | 参数类型   | 描述     |
|------------------------|--------|--------|
| physical_resource_id   | String | 物理资源id |
| physical_resource_name | String | 物理资源名称 |
| logical_resource_name  | String | 逻辑资源名称 |
| logical_resource_type  | String | 逻辑资源类型 |
| resource_status        | String | 资源状态   |
| status_message         | String | 状态信息   |
| href                   | String | 超链接地址  |
| display_name           | String | 云服务名称  |

表 7-841 RepoInfo

| 参数          | 参数类型   | 描述     |
|-------------|--------|--------|
| https_url   | String | http地址 |
| web_url     | String | 存储库链接  |
| repo_status | String | 存储库状态  |
| error_msg   | String | 报错信息   |
| project_id  | String | 项目id   |

状态码： 400

表 7-842 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 401

表 7-843 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 403

表 7-844 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-845 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-846 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

查询指定函数应用详情。

```
GET /v2/{project_id}/fgs/applications/{id}
```

## 响应示例

状态码： 200

```
get application successfully
```

```
{
 "name": "testApplication",
 "last_modified_time": 1699356705515,
 "stack_resources": null,
 "status": "repoFail",
 "stack_name": "",
 "repo_name": "",
 "description": "test for function application.",
 "repo": { },
 "project_id": "c53626012ba84727b938ca8bf03108ef"
}
```

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class ShowFunctionAppSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 ShowFunctionAppRequest request = new ShowFunctionAppRequest();
 try {
 ShowFunctionAppResponse response = client.showFunctionApp(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = ShowFunctionAppRequest()
 response = client.show_function_app(request)
 print(response)
 except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.ShowFunctionAppRequest{}
 response, err := client.ShowFunctionApp(request)
```

```
if err == nil {
 fmt.Printf("%+v\n", response)
} else {
 fmt.Println(err)
}
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                            |
|-----|-------------------------------|
| 200 | get application successfully  |
| 400 | Bad Request 请求错误。             |
| 401 | Unauthorized 鉴权失败。            |
| 403 | Forbidden 没有操作权限。             |
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

## 7.16.6 删除应用程序

### 功能介绍

删除应用程序（该功能目前仅支持华东-上海一、华北-北京四）

### 调用方法

请参见[如何调用API](#)。

### URI

DELETE /v2/{project\_id}/fgs/applications/{id}

表 7-847 路径参数

| 参数         | 是否必选 | 参数类型   | 描述                                       |
|------------|------|--------|------------------------------------------|
| project_id | 是    | String | 租户项目 ID，获取方式请参见 <a href="#">获取项目ID</a> 。 |



| 参数 | 是否必选 | 参数类型   | 描述                         |
|----|------|--------|----------------------------|
| id | 是    | String | 应用ID。<br>最小长度：1<br>最大长度：64 |

## 请求参数

表 7-848 请求 Header 参数

| 参数           | 是否必选 | 参数类型   | 描述                                                           |
|--------------|------|--------|--------------------------------------------------------------|
| X-Auth-Token | 是    | String | 用户Token。<br>通过调用IAM服务获取用户Token接口获取（响应消息头中X-Subject-Token的值）。 |
| Content-Type | 是    | String | 消息体的类型（格式）                                                   |

## 响应参数

状态码：400

表 7-849 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码：401

表 7-850 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码：403

表 7-851 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 404

表 7-852 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

状态码： 500

表 7-853 响应 Body 参数

| 参数         | 参数类型   | 描述   |
|------------|--------|------|
| error_code | String | 错误码  |
| error_msg  | String | 错误信息 |

## 请求示例

删除执行函数应用。

```
DELETE /v2/{project_id}/fgs/applications/{id}
```

## 响应示例

无

## SDK 代码示例

SDK代码示例如下。

### Java

```
package com.huaweicloud.sdk.test;

import com.huaweicloud.sdk.core.auth.ICredential;
import com.huaweicloud.sdk.core.auth.BasicCredentials;
import com.huaweicloud.sdk.core.exception.ConnectionException;
import com.huaweicloud.sdk.core.exception.RequestTimeoutException;
import com.huaweicloud.sdk.core.exception.ServiceResponseException;
import com.huaweicloud.sdk.functiongraph.v2.region.FunctionGraphRegion;
```

```
import com.huaweicloud.sdk.functiongraph.v2.*;
import com.huaweicloud.sdk.functiongraph.v2.model.*;

public class DeleteFunctionAppSolution {

 public static void main(String[] args) {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great
 // security risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or
 // environment variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running
 // this example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 String ak = System.getenv("CLOUD_SDK_AK");
 String sk = System.getenv("CLOUD_SDK_SK");

 ICredential auth = new BasicCredentials()
 .withAk(ak)
 .withSk(sk);

 FunctionGraphClient client = FunctionGraphClient.newBuilder()
 .withCredential(auth)
 .withRegion(FunctionGraphRegion.valueOf("<YOUR REGION>"))
 .build();
 DeleteFunctionAppRequest request = new DeleteFunctionAppRequest();
 try {
 DeleteFunctionAppResponse response = client.deleteFunctionApp(request);
 System.out.println(response.toString());
 } catch (ConnectionException e) {
 e.printStackTrace();
 } catch (RequestTimeoutException e) {
 e.printStackTrace();
 } catch (ServiceResponseException e) {
 e.printStackTrace();
 System.out.println(e.getHttpStatusCode());
 System.out.println(e.getRequestId());
 System.out.println(e.getErrorCode());
 System.out.println(e.getErrorMsg());
 }
 }
}
```

## Python

```
coding: utf-8

from huaweicloudsdkcore.auth.credentials import BasicCredentials
from huaweicloudsdkfunctiongraph.v2.region.functiongraph_region import FunctionGraphRegion
from huaweicloudsdkcore.exceptions import exceptions
from huaweicloudsdkfunctiongraph.v2 import *

if __name__ == "__main__":
 # The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 # risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 # variables and decrypted during use to ensure security.
 # In this example, AK and SK are stored in environment variables for authentication. Before running this
 # example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak = __import__('os').getenv("CLOUD_SDK_AK")
 sk = __import__('os').getenv("CLOUD_SDK_SK")

 credentials = BasicCredentials(ak, sk) \

 client = FunctionGraphClient.new_builder() \
 .with_credentials(credentials) \
 .with_region(FunctionGraphRegion.value_of("<YOUR REGION>")) \
 .build()

 try:
 request = DeleteFunctionAppRequest()
 response = client.delete_function_app(request)
```

```
print(response)
except exceptions.ClientRequestException as e:
 print(e.status_code)
 print(e.request_id)
 print(e.error_code)
 print(e.error_msg)
```

## Go

```
package main

import (
 "fmt"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/core/auth/basic"
 functiongraph "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2"
 "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/model"
 region "github.com/huaweicloud/huaweicloud-sdk-go-v3/services/functiongraph/v2/region"
)

func main() {
 // The AK and SK used for authentication are hard-coded or stored in plaintext, which has great security
 // risks. It is recommended that the AK and SK be stored in ciphertext in configuration files or environment
 // variables and decrypted during use to ensure security.
 // In this example, AK and SK are stored in environment variables for authentication. Before running this
 // example, set environment variables CLOUD_SDK_AK and CLOUD_SDK_SK in the local environment
 ak := os.Getenv("CLOUD_SDK_AK")
 sk := os.Getenv("CLOUD_SDK_SK")

 auth := basic.NewCredentialsBuilder().
 WithAk(ak).
 WithSk(sk).
 Build()

 client := functiongraph.NewFunctionGraphClient(
 functiongraph.FunctionGraphClientBuilder().
 WithRegion(region.ValueOf("<YOUR REGION>")).
 WithCredential(auth).
 Build())

 request := &model.DeleteFunctionAppRequest{}
 response, err := client.DeleteFunctionApp(request)
 if err == nil {
 fmt.Printf("%+v\n", response)
 } else {
 fmt.Println(err)
 }
}
```

## 更多

更多编程语言的SDK代码示例，请参见[API Explorer](#)的代码示例页签，可生成自动对应的SDK代码示例。

## 状态码

| 状态码 | 描述                              |
|-----|---------------------------------|
| 200 | delete application successfully |
| 400 | Bad Request 请求错误。               |
| 401 | Unauthorized 鉴权失败。              |
| 403 | Forbidden 没有操作权限。               |

| 状态码 | 描述                            |
|-----|-------------------------------|
| 404 | Not Found 找不到资源。              |
| 500 | Internal Server Error 服务内部错误。 |

## 错误码

请参见[错误码](#)。

# 8 权限及授权项说明

如果您需要对您所拥有的FunctionGraph进行精细的权限管理，您可以使用统一身份认证服务（Identity and Access Management，简称IAM），如果公有云账号已经能满足您的要求，不需要创建独立的IAM用户，您可以跳过本章节，不影响您使用FunctionGraph服务的其它功能。

默认情况下，新建的IAM用户没有任何权限，您需要将其加入用户组，并给用户组授予策略，才能使用户组中的用户获得策略定义的权限，这一过程称为授权。授权后，用户就可以基于策略对云服务进行操作。

权限根据授权的精细程度，分为角色和策略。角色以服务为粒度，是IAM最初提供了一种根据用户的工作职能定义权限的粗粒度授权机制。策略以API接口为粒度进行权限拆分，授权更加精细，可以精确到某个操作、资源和条件，能够满足企业对权限最小化的安全管控要求。

## 📖 说明

如果您要允许或是禁止某个接口的操作权限，请使用策略。

账号具备所有接口的调用权限，如果使用账号下的IAM用户发起API请求时，该IAM用户必须具备调用该接口所需的权限，否则，API请求将调用失败。每个接口所需要的权限，与各个接口所对应的授权项相对应，只有发起请求的用户被授予授权项所对应的策略，该用户才能成功调用该接口。例如，用户要调用接口来获取函数列表，那么这个IAM用户被授予的策略中必须包含允许“FunctionGraph:function:list”的授权项，该接口才能调用成功。

## 支持的授权项

策略包含系统策略和自定义策略，如果系统策略不满足授权要求，管理员可以创建自定义策略，并通过给用户组授予自定义策略来进行精细的访问控制。策略支持的操作与API相对应，授权项列表说明如下：

- 权限：自定义策略中授权项定义的内容即为权限。
- 对应API接口：自定义策略实际调用的API接口。
- 授权项：自定义策略中支持的Action，在自定义策略中的Action中写入授权项，可以实现授权项对应的权限功能。
- IAM项目(Project)/企业项目(Enterprise Project)：自定义策略的授权范围，包括IAM项目与企业项目。授权范围如果同时支持IAM项目和企业项目，表示此授权项对应的自定义策略，可以在IAM和企业管理两个服务中给用户组授权并生效。如

果仅支持IAM项目，不支持企业项目，表示仅能在IAM中给用户组授权并生效，如果在企业管理中授权，则该自定义策略不生效。关于IAM项目与企业项目的区别，详情请参见：[IAM与企业管理的区别](#)。

 说明

“√”表示支持，“x”表示暂不支持。

FunctionGraph服务支持的自定义策略授权项如[表8-1](#)所示。

**表 8-1** FunctionGraph 授权项明细

| 权限                | 对应API接口                                                     | 授权项                                  | IAM项目 (Project) | 企业项目 (Enterprise Project) |
|-------------------|-------------------------------------------------------------|--------------------------------------|-----------------|---------------------------|
| 获取函数列表            | GET /v2/{project_id}/fgs/functions                          | FunctionGraph:function:list          | √               | √                         |
| 获取函数的 metadata    | GET /v2/{project_id}/fgs/functions/{function_urn}/config    | FunctionGraph:function:getConfig     | √               | √                         |
| 获取指定函数代码          | GET /v2/{project_id}/fgs/functions/{function_urn}/code      | FunctionGraph:function:getCode       | √               | √                         |
| 创建函数              | POST /v2/{project_id}/fgs/functions                         | FunctionGraph:function:create        | √               | √                         |
| 删除函数/版本           | DELETE /v2/{project_id}/fgs/functions/{function_urn}        | FunctionGraph:function:delete        | √               | √                         |
| 修改函数代码            | PUT /v2/{project_id}/fgs/functions/{function_urn}/code      | FunctionGraph:function:updateCode    | √               | √                         |
| 修改函数的 metadata 信息 | PUT /v2/{project_id}/fgs/functions/{function_urn}/config    | FunctionGraph:function:updateConfig  | √               | √                         |
| 发布函数版本            | POST /v2/{project_id}/fgs/functions/{function_urn}/versions | FunctionGraph:function:createVersion | √               | √                         |

| 权限             | 对应API接口                                                                           | 授权项                                                   | IAM项目 (Project) | 企业项目 (Enterprise Project) |
|----------------|-----------------------------------------------------------------------------------|-------------------------------------------------------|-----------------|---------------------------|
| 获取指定函数的版本列表    | GET /v2/{project_id}/fgs/functions/{function_urn}/versions                        | FunctionGraph:function:listVersion                    | √               | √                         |
| 创建函数版本别名       | POST /v2/{project_id}/fgs/functions/{function_urn}/aliases                        | FunctionGraph:function:createAlias                    | √               | √                         |
| 修改函数版本别名信息     | PUT /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}            | FunctionGraph:function:updateAlias                    | √               | √                         |
| 删除函数版本别名       | DELETE /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}         | FunctionGraph:function:deleteAlias                    | √               | √                         |
| 获取函数版本的指定别名信息  | GET /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}            | FunctionGraph:function:getAlias                       | √               | √                         |
| 获取指定函数所有版本别名列表 | GET /v2/{project_id}/fgs/functions/{function_urn}/aliases                         | FunctionGraph:function:listAlias                      | √               | √                         |
| 获取指定函数的所有触发器   | GET /v2/{project_id}/fgs/triggers/{function_urn}                                  | FunctionGraph:trigger:listSpecifiedFunctionTriggers   | √               | √                         |
| 获取指定触发器的信息     | GET /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id} | FunctionGraph:trigger:get                             | √               | √                         |
| 删除指定函数的所有触发器   | DELETE /v2/{project_id}/fgs/triggers/{function_urn}                               | FunctionGraph:trigger:deleteSpecifiedFunctionTriggers | √               | √                         |



| 权限     | 对应API接口                                                                                                  | 授权项                                        | IAM项目<br>(Project) | 企业项目<br>(Enterprise Project) |
|--------|----------------------------------------------------------------------------------------------------------|--------------------------------------------|--------------------|------------------------------|
| 创建触发器  | POST /v2/<br>{project_id}/fgs/<br>triggers/<br>{function_urn}                                            | FunctionGraph:tr<br>igger:create           | √                  | √                            |
| 删除触发器  | DELETE /v2/<br>{project_id}/fgs/<br>triggers/<br>{function_urn}/<br>{trigger_type_code}/<br>{trigger_id} | FunctionGraph:tr<br>igger:delete           | √                  | √                            |
| 同步执行函数 | POST /v2/<br>{project_id}/fgs/<br>functions/<br>{function_urn}/<br>invocations                           | FunctionGraph:f<br>unction:invoke          | √                  | √                            |
| 异步执行函数 | POST /v2/<br>{project_id}/fgs/<br>functions/<br>{function_urn}/<br>invocations-async                     | FunctionGraph:f<br>unction:invokeA<br>sync | √                  | √                            |

# 9 附录

## 9.1 状态码

状态码如表9-1所示

表 9-1 状态码

| 状态码 | 返回                            | 说明                           |
|-----|-------------------------------|------------------------------|
| 200 | -                             | 请求成功。                        |
| 204 | -                             | 请求成功。                        |
| 400 | Bad Request                   | 服务器未能处理请求。                   |
| 401 | Unauthorized                  | 被请求的页面需要认证。                  |
| 403 | Forbidden                     | 对被请求页面的访问被禁止。                |
| 404 | Not Found                     | 服务器无法找到被请求的页面。               |
| 405 | Method Not Allowed            | 请求中指定的方法不被允许。                |
| 406 | Not Acceptable                | 服务器生成的响应无法被客户端所接受。           |
| 407 | Proxy Authentication Required | 用户必须首先使用代理服务器进行验证，这样请求才会被处理。 |
| 408 | Request Timeout               | 请求超出了服务器的等待时间。               |
| 409 | Conflict                      | 由于冲突，请求无法被完成。                |
| 500 | Internal Server Error         | 请求未完成，服务异常。                  |
| 501 | Not Implemented               | 请求未完成，服务器不支持所请求的功能。          |
| 502 | Bad Gateway                   | 请求未完成，服务器从上游服务器收到一个无效的响应。    |

| 状态码 | 返回                  | 说明            |
|-----|---------------------|---------------|
| 503 | Service Unavailable | 请求未完成，系统暂时异常。 |
| 504 | Gateway Timeout     | 网关超时。         |

## 9.2 错误码

| 状态码 | 错误码      | 错误信息                              | 描述           | 处理措施         |
|-----|----------|-----------------------------------|--------------|--------------|
| 400 | FSS.0400 | Invalid parameter.                | 无效参数         | 检查请求body体中参数 |
| 400 | FSS.1001 | Invalid query parameter.          | 无效的查询参数      | 检查查询参数       |
| 400 | FSS.1002 | Invalid function name.            | 无效的函数名称      | 检查函数名称       |
| 400 | FSS.1003 | Invalid function handler.         | 无效的函数Handler | 检查函数执行入口     |
| 400 | FSS.1004 | Invalid Package (function app).   | 无效的函数包       | 检查函数包        |
| 400 | FSS.1005 | Invalid runtime.                  | 无效的Runtime   | 检查运行时        |
| 400 | FSS.1006 | Invalid function code entry mode. | 无效的代码类型      | 检查代码类型       |
| 400 | FSS.1007 | Invalid function memory.          | 无效的内存        | 检查函数内存       |
| 400 | FSS.1008 | Invalid function timeout.         | 无效的超时时间      | 检查函数超时时间     |
| 400 | FSS.1009 | Invalid function UserData.        | 无效的环境变量      | 检查函数环境变量     |
| 400 | FSS.1010 | Invalid URL.                      | 无效的URL       | 检查URL        |
| 400 | FSS.1011 | Invalid function code.            | 无效的函数代码      | 检查函数代码       |

| 状态码 | 错误码      | 错误信息                                                                                       | 描述                     | 处理措施               |
|-----|----------|--------------------------------------------------------------------------------------------|------------------------|--------------------|
| 400 | FSS.1012 | The function code must be configured.                                                      | 函数代码不能为空               | 检查函数代码             |
| 400 | FSS.1013 | Invalid file type.                                                                         | 无效的文件类型                | 检查文件类型             |
| 400 | FSS.1014 | Invalid function alias.                                                                    | 无效的函数版本别名              | 检查函数别名             |
| 400 | FSS.1015 | Invalid function version.                                                                  | 别名对应的版本无效              | 检查函数版本             |
| 400 | FSS.1016 | The function cannot be published because no changes have been made since last publication. | 无效的版本发布操作，与上次发布相比无任何更改 | 确认与上次版本发布相比函数进行了更改 |
| 400 | FSS.1017 | The number of environment variables exceeds the maximum allowed limit (20).                | 环境变量个数超过限制 (20)        | 检查环境变量的个数          |
| 400 | FSS.1018 | The total size of environment variables exceeds the maximum allowed limit (2 KB).          | 环境变量总大小超过限制 (2K)       | 检查环境变量的总大小         |
| 400 | FSS.1019 | The description exceeds the maximum allowed limit (512 characters).                        | 描述信息长度超过限制 (512 字符数)   | 检查描述长度             |
| 400 | FSS.1020 | Failed to validate XRole.                                                                  | 无效的委托                  | 检查请求body体中xrole参数  |

| 状态码 | 错误码      | 错误信息                                                     | 描述                           | 处理措施               |
|-----|----------|----------------------------------------------------------|------------------------------|--------------------|
| 400 | FSS.1021 | invalid service link agency.                             | 创建服务关联委托失败                   | 检查服务关联委托信息相关权限     |
| 400 | FSS.1022 | Only one YAML file is allowed.                           | 只容许有一个YAML文件                 | 确认是否只有一个YAML文件     |
| 400 | FSS.1023 | The imported file is too large.                          | 导入的文件太大了, 超过了最大限制            | 检查导入文件的大小          |
| 400 | FSS.1024 | Invalid dependency.                                      | 无效的依赖包                       | 检查依赖包              |
| 400 | FSS.1025 | Invalid YAML file.                                       | 无效的YAML文件                    | 检查YAML文件           |
| 400 | FSS.1026 | Invalid Concurrency.                                     | 无效的并发策略                      | 检查并发策略             |
| 400 | FSS.1027 | Invalid packageName (app name).                          | 无效的包名或者应用名称                  | 检查包和应用名称。          |
| 400 | FSS.1028 | The app cannot be deleted because it contains functions. | 该应用不能删除, 因为该应用下还有函数存在        | 应用删除前, 检查其下是否有函数存在 |
| 400 | FSS.1029 | The default app cannot be deleted.                       | 默认的应用不能删除                    | 默认的应用不能删除          |
| 400 | FSS.1031 | The dependency already exists.                           | 依赖包已经存在                      | 检查依赖包是否存在          |
| 400 | FSS.1032 | Invalid dependency type.                                 | 无效的依赖包类型, 当前只容许本地zip包或者OBS上传 | 检查依赖包类型            |
| 400 | FSS.1034 | Invalid image URL.                                       | 无效的镜像地址                      | 检查镜像地址             |
| 400 | FSS.1035 | The image does not exist.                                | 该镜像不存在                       | 检查镜像是否存在           |
| 400 | FSS.1036 | The VPC does not exist.                                  | 该VPC不存在                      | 检查VPC是否存在          |

| 状态码 | 错误码      | 错误信息                                                                                                                  | 描述                                         | 处理措施               |
|-----|----------|-----------------------------------------------------------------------------------------------------------------------|--------------------------------------------|--------------------|
| 400 | FSS.1037 | No subnet matches the specified ID.                                                                                   | 没有匹配的子网                                    | 检查子网是否存在           |
| 400 | FSS.1038 | The file system configuration already exists in the function.                                                         | 在该函数中已经存在相同的文件系统配置                         | 检查文件系统配置是否重复       |
| 400 | FSS.1039 | The mounting path is invalid.                                                                                         | 无效的挂载路径                                    | 检查挂载路径             |
| 400 | FSS.1058 | The app name and function name cannot be the same in the YAML file.                                                   | 在YAML文件中不允许存在相同的应用名和函数名                    | 检查YAML中应用名和函数名是否重复 |
| 400 | FSS.1074 | The event field is invalid.                                                                                           | 无效的事件参数                                    | 检查事件参数             |
| 400 | FSS.1075 | The value of UserId or GroupId must be a non-zero integer from -1 to 65534.                                           | 用户ID和用户组ID必须是从-1到65534之间的整数(0、1000、1002除外) | 检查用户ID和用户组ID       |
| 400 | FSS.1076 | The number of reserved instances exceeds the maximum allowed limit.                                                   | 函数的预留实例数超过了最大限制                            | 检查函数预留实例的个数        |
| 400 | FSS.1077 | The code and configuration of the function cannot be updated because the function has at least one reserved instance. | 该函数下存在预留实例，不能更新函数代码和配置                     | 检查函数下是否存在预留实例      |

| 状态码 | 错误码      | 错误信息                                                             | 描述           | 处理措施        |
|-----|----------|------------------------------------------------------------------|--------------|-------------|
| 400 | FSS.1081 | Invalid function async invoke config.                            | 无效的异步配置参数    | 检查异步配置参数    |
| 400 | FSS.1091 | The additional version is invalid.                               | 无效的灰度版本      | 检查灰度版本      |
| 400 | FSS.1092 | The weight of the additional version is invalid.                 | 无效的灰度版本权重值   | 检查灰度版本权重值   |
| 400 | FSS.1093 | The major version and the additional version cannot be the same. | 灰度版本不能与主版本一致 | 检查灰度版本和主版本  |
| 400 | FSS.1101 | Invalid trigger type.                                            | 无效的触发器类型     | 检查触发器类型     |
| 400 | FSS.1102 | Invalid SMN trigger parameters.                                  | 无效的SMN触发器参数  | 检查SMN触发器参数  |
| 400 | FSS.1103 | Invalid DMS trigger parameters.                                  | 无效的DMS触发器参数  | 检查DMS触发器参数  |
| 400 | FSS.1104 | Invalid DIS trigger parameters.                                  | 无效的DIS触发器参数  | 检查DIS触发器参数  |
| 400 | FSS.1106 | Invalid OBS trigger parameters.                                  | 无效的OBS触发器参数  | 检查OBS触发器参数  |
| 400 | FSS.1107 | Invalid APIG trigger parameters.                                 | 无效的APIG触发器参数 | 检查APIG触发器参数 |
| 400 | FSS.1109 | Invalid timer trigger parameters.                                | 无效的定时任务触发器参数 | 检查定时任务触发器参数 |
| 400 | FSS.1110 | Invalid LTS trigger parameters.                                  | 无效的LTS触发器参数  | 检查LTS触发器参数  |

| 状态码 | 错误码      | 错误信息                                                                           | 描述                                  | 处理措施                                |
|-----|----------|--------------------------------------------------------------------------------|-------------------------------------|-------------------------------------|
| 400 | FSS.1112 | The Kafka trigger parameters are invalid.                                      | 无效的Kafka触发器参数                       | 检查Kafka触发器参数                        |
| 400 | FSS.1113 | The username and password must be specified because Kafka SASL_SSL is enabled. | 若Kafka开启了SASL_SSL协议，所以必须设置对应的用户名和密码 | 若Kafka开启了SASL_SSL协议，检查是否设置对应的用户名和密码 |
| 400 | FSS.1114 | The subnet of the function must be the same as that of the Kafka instance.     | 函数的子网与kafka实例所在的子网不一样               | 检查函数的子网与Kafka实例所在的子网是否一致            |
| 400 | FSS.1116 | Kafka instance configuration error. Please check the username and password.    | Kafka实例配置错误，请检查对应的用户名和密码是否正确        | 检查Kafka实例配置的用户名和密码是否正确              |
| 400 | FSS.1117 | Failed to query messages from the Kafka instance.                              | 从kafka实例读取消息失败                      | 检查Kafka实例是否正常                       |
| 400 | FSS.1164 | Invalid DDS trigger parameters.                                                | 无效的DDS触发器参数                         | 检查DDS触发器参数                          |
| 400 | FSS.1172 | The database instance does not exist.                                          | 找不到指定的数据库实例                         | 检查数据库实例                             |
| 400 | FSS.1173 | DDS DB instance configuration error. Please contact DDS R&D personnel.         | DDS实例配置错误，请联系DDS研发人员                | 联系DDS开发人员                           |



| 状态码 | 错误码      | 错误信息                                                      | 描述              | 处理措施           |
|-----|----------|-----------------------------------------------------------|-----------------|----------------|
| 400 | FSS.1204 | Failed to get response from functionAccesssor.            | 无法获取函数执行的返回值    | 联系开发人员         |
| 400 | FSS.1221 | lts log has been enabled.                                 | 用户已开通LTS日志功能    | 检查LTS日志功能是否已开通 |
| 400 | FSS.1301 | The CTS trigger does not exist.                           | CTS触发器不存在       | 检查CTS触发器       |
| 400 | FSS.1303 | Access denied due to insufficient permissions.            | 权限校验失败，访问被禁止    | 检查是否授权         |
| 400 | FSS.1304 | The CTS service failed to process the request.            | 服务器未能处理请求       | 联系CTS开发人员      |
| 400 | FSS.1305 | Invalid CTS trigger parameters.                           | 无效的CTS触发器参数     | 检查CTS触发器参数     |
| 400 | FSS.1306 | The number of triggers exceeds the maximum allowed limit. | 触发器达到上限         | 检查触发器数目是否达到上限  |
| 400 | FSS.1307 | The trigger name already exists.                          | 触发器名称已存在        | 检查触发器名称是否重复    |
| 400 | FSS.1308 | The operation resource does not exist.                    | 操作资源不存在         | 检查资源是否存在       |
| 400 | FSS.1309 | Invalid function URN.                                     | functionUrn验证错误 | 检查functionUrn  |
| 400 | FSS.1310 | Unauthorized user.                                        | 获取组用户Token验证失败  | 检查权限           |
| 400 | FSS.1311 | Unauthorized access.                                      | 未开通CTS服务        | 检查CTS服务是否开通    |
| 400 | FSS.1312 | The notification name must be specified.                  | 关键通知名称为空        | 检查关键通知名称       |

| 状态码 | 错误码      | 错误信息                                                                                | 描述                 | 处理措施                |
|-----|----------|-------------------------------------------------------------------------------------|--------------------|---------------------|
| 400 | FSS.1313 | The number of resources to be operated has reached the maximum allowed limit (100). | 操作资源超过100条上限       | 检查操作资源的数目是否超过上限     |
| 400 | FSS.1314 | The operation resource must be specified.                                           | 操作资源为空             | 检查是否操作资源是否为空        |
| 400 | FSS.1316 | The resource operation has already been selected.                                   | 选择重复的操作。           | 检查是否选择的相同的操作。       |
| 400 | FSS.1317 | The trigger name is too long.                                                       | 触发器名称过长            | 检查触发器名字的长度          |
| 400 | FSS.1318 | Invalid trigger operation.                                                          | 无效的触发器操作           | 检查触发器操作             |
| 400 | FSS.1404 | Invalid function initializer.                                                       | 非法的函数初始化入口         | 检查函数初始化入口           |
| 400 | FSS.1405 | Invalid initialization timeout.                                                     | 非法的函数初始化超时时间       | 检查函数初始化超时时间         |
| 400 | FSS.1406 | Invalid heartbeat handler.                                                          | 非法的函数心跳检测入口        | 检查函数心跳检测入口          |
| 400 | FSS.1407 | Invalid restore hook handler.                                                       | 无效的restore hook处理器 | 检查restore hook配置    |
| 400 | FSS.1426 | This release[{{version}}] has reserved instances configured.                        | 目标别名对应的版本已配置预留策略   | 查询目标别名对应的版本是否配置弹性策略 |
| 400 | FSS.1441 | The dependency has versions, can not update or delete.                              | 依赖包已存在版本，无法更新或删除   | 检查依赖包是否存在版本         |

| 状态码 | 错误码      | 错误信息                                                                        | 描述                   | 处理措施                     |
|-----|----------|-----------------------------------------------------------------------------|----------------------|--------------------------|
| 400 | FSS.1601 | Invalid swr image param.                                                    | 无效的SWR镜像参数           | 检查SWR镜像参数                |
| 401 | FSS.0401 | Unauthorized access.                                                        | 未授权的访问               | 检查用户的权限                  |
| 401 | FSS.1118 | Access denied.                                                              | 访问受限, 该用户不在白名单中      | 检查白名单配置                  |
| 403 | FSS.0403 | Forbidden.                                                                  | 禁止操作                 | 检查用户的操作权限                |
| 403 | FSS.1040 | The selected Xrole does not have permissions to mount the resources.        | 选择的委托没有权限挂载该资源类型     | 检查选择的委托权限                |
| 403 | FSS.1041 | The number of functions exceeds the maximum allowed limit (400).            | 租户的函数数量超过限制 (400)    | 检查租户的函数数量是否超限            |
| 403 | FSS.1042 | The total code size of functions exceeds the maximum allowed limit (20 GB). | 租户的函数的总大小超过限制 (20G)  | 检查函数的总大小是否超过限制 (20G)     |
| 403 | FSS.1043 | The number of aliases exceeds the maximum allowed limit.                    | 别名数量超过了允许的最大限制       | 检查别名数量是否超过了允许的最大限制       |
| 403 | FSS.1044 | The number of apps exceeds the maximum allowed limit (400).                 | 应用数量超过了允许的最多限制 (400) | 检查应用数量是否超过了允许的最多限制 (400) |

| 状态码 | 错误码      | 错误信息                                                                                      | 描述                      | 处理措施                        |
|-----|----------|-------------------------------------------------------------------------------------------|-------------------------|-----------------------------|
| 403 | FSS.1045 | The number of dependencies exceeds the maximum allowed limit.                             | 依赖包数量超过了最大限制            | 检查依赖包数量是否超过了最大限制            |
| 403 | FSS.1046 | The dependency is inaccessible.                                                           | 该依赖包不可用                 | 检查依赖包是否可用                   |
| 403 | FSS.1047 | The number of bound VPCs exceeds the maximum limit allowed for a tenant.                  | 租户的绑定的VPC数量超过了最大限制      | 检查租户的绑定的VPC数量是否超过了最大限制      |
| 403 | FSS.1048 | The number of bound VPCs exceeds the maximum limit allowed for a project.                 | 单个项目的绑定的VPC数量超过了最大限制    | 检查单个项目的绑定的VPC数量是否超过了最大限制    |
| 403 | FSS.1049 | The number of file systems mounted to the function exceeds the maximum allowed limit (5). | 单个函数挂载文件系统数量超过了最大限制 (5) | 检查单个函数挂载文件系统数量是否超过了最大限制 (5) |
| 403 | FSS.1068 | The number of events configured for the function exceeds the maximum allowed limit.       | 单个函数的事件数超过了最大限制         | 检查单个函数的事件数是否超过了最大限制         |
| 403 | FSS.1069 | The size of EventData exceeds 4 KB.                                                       | 事件大小超过了允许的最大限制 (4KB)    | 检查代码大小是否超过了允许的最大限制 (4KB)    |
| 403 | FSS.1073 | The function event already exists.                                                        | 事件在函数中已经存在              | 检查事件在函数中是否已经存在              |

| 状态码 | 错误码      | 错误信息                                                                                            | 描述                           | 处理措施                           |
|-----|----------|-------------------------------------------------------------------------------------------------|------------------------------|--------------------------------|
| 403 | FSS.1094 | The version cannot be deleted when it is used as the additional version of an alias.            | 不能删除正在被灰度版本使用的版本号            | 检查灰度版本号是否正在被使用                 |
| 403 | FSS.1096 | The file sharing protocol of the mounted resource is not NFS.                                   | 挂载资源的文件共享协议不是NFS             | 检查挂载资源的文件共享协议是否为NFS            |
| 403 | FSS.1108 | The bucket configuration of the current trigger conflicts with that of an existing OBS trigger. | 当前触发器的存储桶配置与现有OBS触发器的存储桶配置冲突 | 检查触发器的存储桶配置与现有OBS触发器的存储桶配置是否冲突 |
| 403 | FSS.1121 | Forbidden                                                                                       | 访问受限，请检查是否有配置对应的DMS委托        | 检查是否配置对应的DMS委托                 |
| 403 | FSS.1122 | Forbidden                                                                                       | 访问受限，请检查是否有配置对应的DIS委托        | 检查是否配置对应的DIS委托                 |
| 403 | FSS.1123 | The number of pull triggers exceeds the maximum allowed limit (10).                             | 拉模式的触发器数量达到最大值10             | 检查拉模式的触发器数量是否达到最大值10           |
| 403 | FSS.1124 | The number of APIs exceeds the maximum allowed limit.                                           | API数量超过了最大限制                 | 检查API数量是否超过了最大限制               |
| 403 | FSS.1125 | Forbidden                                                                                       | 无权限访问，请检查是否配置了对应的LTS委托       | 检查是否配置了对应的LTS委托                |

| 状态码 | 错误码      | 错误信息                                                         | 描述                   | 处理措施                |
|-----|----------|--------------------------------------------------------------|----------------------|---------------------|
| 403 | FSS.1126 | You do not have permissions to call the API.                 | 没有权限访问API            | 检查是否有权限访问API        |
| 403 | FSS.1127 | The EPS user does not have permissions to call the API.      | EPS用户没有权限访问该API      | 检查权限                |
| 403 | FSS.1128 | list enterprise failed.                                      | 企业项目用户获取有权限企业列表失败    | 检查企业项目细粒度权限         |
| 403 | FSS.1154 | Aliases of a function bound with triggers cannot be deleted. | 该函数别名下面存在触发器，不能删除该别名 | 删除函数别名之前，先删除其下的触发器  |
| 404 | FSS.0404 | The specified resource cannot be found.                      | 没有找到指定的资源            | 检查访问的资源是否存在         |
| 404 | FSS.1050 | The mounted resource cannot be found.                        | 找不到挂载的资源类型           | 检查挂载的资源类型是否存在       |
| 404 | FSS.1051 | The function does not exist.                                 | 找不到指定的函数             | 检查函数是否存在            |
| 404 | FSS.1052 | The version does not exist.                                  | 找不到指定的函数版本           | 检查函数版本是否存在          |
| 404 | FSS.1053 | The alias does not exist.                                    | 找不到指定的函数版本别名         | 检查函数的指定版本别名是否存在     |
| 404 | FSS.1054 | The function code package does not exist in OBS.             | 从OBS找不到指定的函数代码包      | 检查指定的函数代码包在OBS中是否存在 |
| 404 | FSS.1055 | The app does not exist in obs.                               | 从OBS找不到指定的函数应用       | 检查指定的函数应用在OBS中是否存在  |

| 状态码 | 错误码      | 错误信息                                                        | 描述              | 处理措施             |
|-----|----------|-------------------------------------------------------------|-----------------|------------------|
| 404 | FSS.1056 | The dependency does not exist.                              | 找不到指定的依赖包       | 检查依赖包是否存在        |
| 404 | FSS.1057 | The function name does not exist in the YAML file.          | 在YAML文件中找不到该函数名 | 检查YAML文件中是否有该函数名 |
| 404 | FSS.1059 | The function template does not exist.                       | 函数模板不存在         | 检查函数模板是否存在       |
| 404 | FSS.1060 | The event template cannot be found.                         | 事件模板不存在         | 检查事件模板是否存在       |
| 404 | FSS.1070 | The event cannot be found.                                  | 事件不存在           | 检查事件是否存在         |
| 404 | FSS.1086 | The function async invocation configuration does not exist. | 函数异步配置不存在       | 请检查函数异步配置        |
| 404 | FSS.1111 | The Kafka resource cannot be found.                         | 指定的kafka不存在     | 检查指定的Kafka是否存在   |
| 404 | FSS.1130 | The DIS stream does not exist.                              | 没有找到指定的DIS通道    | 检查指定的DIS通道是否存在   |
| 404 | FSS.1131 | The trigger does not exist.                                 | 没有找到指定的触发器      | 检查指定的触发器是否存在     |
| 404 | FSS.1132 | The SMN trigger does not exist. View the SMN console.       | 没有找到指定的SMN触发器   | 检查指定的SMN触发器是否存在  |
| 404 | FSS.1133 | The DMS trigger does not exist.                             | 没有找到指定的DMS触发器   | 检查DMS触发器是否存在     |

| 状态码 | 错误码      | 错误信息                                                        | 描述             | 处理措施               |
|-----|----------|-------------------------------------------------------------|----------------|--------------------|
| 404 | FSS.1134 | The DIS trigger does not exist.                             | 没有找到指定的DIS触发器  | 检查DIS触发器是否存在       |
| 404 | FSS.1135 | The DMS queue does not exist.                               | 没有找到指定的DMS队列   | 检查指定的DMS队列是否存在     |
| 404 | FSS.1136 | The OBS trigger does not exist.                             | 没有找到指定的OBS触发器  | 检查OBS触发器是否存在       |
| 404 | FSS.1137 | Invalid trigger type.                                       | 无效的触发器类型       | 检查触发器类型是否合法        |
| 404 | FSS.1138 | The APIG trigger does not exist.                            | 没有找到指定的APIG触发器 | 检查APIG触发器是否存在      |
| 404 | FSS.1139 | The DMS consumer group does not exist.                      | 没有找到指定的DMS消费组  | 检查DMS消费组是否存在       |
| 404 | FSS.1140 | The timer trigger does not exist.                           | 没有找到指定的定时任务触发器 | 检查定时任务触发器是否存在      |
| 404 | FSS.1160 | The LTS trigger does not exist.                             | 没有找到指定的LTS触发器  | 检查LTS触发器是否存在       |
| 404 | FSS.1161 | The LTS topic does not exist.                               | 没有找到指定的LTS日志主题 | 检查指定的LTS日志主题是否存在   |
| 404 | FSS.1163 | The DDS trigger cannot be found.                            | 找不到指定的DDS触发器   | 检查指定的DDS触发器是否存在    |
| 404 | FSS.1171 | The SMN topic does not exist. Create it on the SMN console. | 该SMN的topic不存在  | 检查指定的SMN的topic是否存在 |
| 404 | FSS.1174 | The Kafka trigger does not exist.                           | 找不到指定的Kafka触发器 | 检查定的Kafka触发器是否存在   |
| 406 | FSS.0406 | Not acceptable.                                             | 请求格式无效         | 对着api检查请求body体     |



| 状态码 | 错误码      | 错误信息                                            | 描述               | 处理措施           |
|-----|----------|-------------------------------------------------|------------------|----------------|
| 406 | FSS.1151 | The OBS bucket is in a different region.        | OBS桶所在区域与当前区域不匹配 | 更换OBS桶         |
| 408 | FSS.0408 | Request timeout.                                | 请求超时             | 检查网络           |
| 409 | FSS.0409 | The specified resource already exists.          | 指定的资源已经存在        | 检查是否重复创建资源     |
| 409 | FSS.1061 | The function already exists.                    | 函数已经存在           | 检查是否重复创建函数     |
| 409 | FSS.1062 | The version already exists.                     | 函数版本已经存在         | 检查是否重复创建函数版本   |
| 409 | FSS.1063 | The alias already exists.                       | 函数版本别名已经存在       | 检查是否重复创建函数版本别名 |
| 409 | FSS.1064 | The app already exists.                         | 函数应用已经存在         | 检查是否重复创建函数应用   |
| 409 | FSS.1065 | The dependency already exists.                  | 依赖包已经存在          | 检查依赖包是否重复创建    |
| 409 | FSS.1066 | The version is already in use by another alias. | 该版本号已经在其他别名中使用了  | 检查版本号是否已被占用    |
| 409 | FSS.1067 | The function template already exists.           | 函数模板已经存在         | 检查函数模板是否重复创建   |
| 409 | FSS.1141 | The SMN trigger already exists.                 | 指定的SMN触发器已经存在    | 检查SMN触发器是否重复创建 |
| 409 | FSS.1142 | The DMS trigger already exists.                 | 指定的DMS触发器已经存在    | 检查DMS触发器是否重复创建 |
| 409 | FSS.1143 | The DIS trigger already exists.                 | 指定的DIS触发器已经存在    | 检查DIS触发器是否重复创建 |
| 409 | FSS.1145 | The OBS trigger already exists.                 | 指定的OBS触发器已经存在    | 检查OBS触发器是否重复创建 |

| 状态码 | 错误码      | 错误信息                                         | 描述              | 处理措施                |
|-----|----------|----------------------------------------------|-----------------|---------------------|
| 409 | FSS.1146 | The APIG trigger already exists.             | 指定的APIG触发器已经存在  | 检查APIG触发器是否重复创建     |
| 409 | FSS.1147 | The request path already exists.             | APIG触发路径已存在     | 检查APIG触发路径是否已经存在    |
| 409 | FSS.1148 | The timer trigger already exists.            | 指定的定时任务触发器已经存在  | 检查指定的定时任务触发器是否已经存在  |
| 409 | FSS.1149 | The LTS trigger already exists.              | LTS触发器已存在       | 检查LTS触发器是否已存在       |
| 409 | FSS.1150 | The Kafka trigger already exists.            | 指定的kafka触发器已经存在 | 检查指定的Kafka触发器是否已经存在 |
| 409 | FSS.1166 | The DDS trigger already exists.              | 指定的DDS触发器已经存在   | 检查指定的DDS触发器是否已经存在   |
| 410 | FSS.0410 | The specified resource does not exist.       | 指定的资源不存在        | 检查指定的资源是否存在         |
| 412 | FSS.1033 | The dependency is currently in use.          | 依赖包正在使用，不能被删除   | 删除依赖包之前，请检查是否被使用    |
| 412 | FSS.1090 | The subnet is not in the ACTIVE state.       | 子网不是ACTIVE状态    | 检查子网状态是否为可用         |
| 412 | FSS.1095 | The mounted resource is not ready.           | 挂载资源还没有准备好      | 准备好挂载资源             |
| 412 | FSS.1153 | The triggering conditions have not been met. | 触发器前置条件不满足      | 检查触发器前置条件           |

| 状态码 | 错误码      | 错误信息                                                                                                      | 描述                   | 处理措施                     |
|-----|----------|-----------------------------------------------------------------------------------------------------------|----------------------|--------------------------|
| 412 | FSS.1440 | Specified dependency version is used by X functions, please delete the function or the relation at first. | 当前依赖包版本正在被X个函数使用     | 检查依赖包版本是否被使用             |
| 413 | FSS.0413 | The request body is too large.                                                                            | 请求体大小超过限制            | 检查请求体                    |
| 413 | FSS.1071 | The size of the code package to be uploaded exceeds the maximum allowed limit (50 MB).                    | 上传代码包大小超过限制 ( 50M )  | 检查代码包大小                  |
| 413 | FSS.1072 | The size of the inline code exceeds the maximum allowed limit (10 KB).                                    | 在线编辑代码大小超过限制 ( 10K ) | 检查在线编辑代码大小是否超过限制 ( 10K ) |
| 413 | FSS.1201 | The request body is too large.                                                                            | 请求体大小超过限制            | 检查请求体大小                  |
| 424 | FSS.0424 | Invalid dependency.                                                                                       | 无效的依赖                | 检查依赖是否有效                 |
| 426 | FSS.0426 | An upgrade is required.                                                                                   | 不支持该操作, 需要升级         | 联系开发人员                   |
| 426 | FSS.1152 | The selected bucket cannot be used to create a trigger.                                                   | 所选的OBS桶不支持创建触发器      | 联系开发人员                   |
| 428 | FSS.0428 | The preconditions have not been met.                                                                      | 先决条件不满足              | 检查先决条件                   |

| 状态码 | 错误码      | 错误信息                                                                                                                              | 描述                               | 处理措施                      |
|-----|----------|-----------------------------------------------------------------------------------------------------------------------------------|----------------------------------|---------------------------|
| 429 | FSS.0429 | Too many requests.                                                                                                                | 太多的请求                            | 请求数过多，稍后重试                |
| 429 | FSS.1429 | Function concurrent invocation requests exceeds max value, please increase the max instances num or max concurrency per instance. | 函数并发调用请求数超过最大值（函数最大实例数*函数单实例并发数） | 建议将函数最大实例数或者单实例并发数配置调大后重试 |
| 500 | FSS.0500 | Internal service error.                                                                                                           | 服务临时性异常，服务内部调用出错，稍后重试            | 服务临时性异常，服务内部调用出错，稍后重试     |
| 500 | FSS.1162 | The operation cannot take effect immediately due to service exception.                                                            | 由于服务异常，该操作无法立即生效                 | 联系开发人员                    |
| 500 | FSS.1202 | The response body or callback body is invalid because they do not contain any status code.                                        | 函数返回的 body 无效                    | 联系开发人员                    |
| 500 | FSS.1206 | Failed to convert result for apig.                                                                                                | 无法转换 apig 的结果                    | 检查 apig 返回或联系开发人员         |
| 500 | FSS.1302 | Failed to save the data.                                                                                                          | 触发器存储失败                          | 稍后重试或联系开发人员               |
| 500 | FSS.1315 | The CTS service is unavailable.                                                                                                   | 服务器异常                            | 联系 CTS 开发人员               |
| 500 | FSS.4001 | function entry is not found.                                                                                                      | 找不到函数入口。                         | 请检查函数的执行入口和上传代码的函数入口是否一致。 |

| 状态码 | 错误码      | 错误信息                                                                | 描述               | 处理措施                       |
|-----|----------|---------------------------------------------------------------------|------------------|----------------------------|
| 500 | FSS.4002 | The user function failed to run.                                    | 用户函数运行报错         | 请根据错误提示信息排查自身代码问题。         |
| 500 | FSS.4003 | state content is too large.                                         | 状态信息太大。          | 检查状态信息。                    |
| 500 | FSS.4004 | the return value of the function exceeds the maximum allowed limit. | 函数返回值内容超平台限制。    | 请检查函数返回值内容的大小是否超过平台限制（6M）。 |
| 500 | FSS.4005 | state is undefined.                                                 | 状态未定义。           | 检查状态信息。                    |
| 500 | FSS.4006 | function name has no service ID.                                    | 函数不是有状态函数。       | 检查函数是否配置了serviceID。        |
| 500 | FSS.4007 | function result is invalid.                                         | 函数调用结果非法。        | 根据错误提示检查函数代码。              |
| 500 | FSS.4008 | response body size %d exceeds the limit of %d.                      | 函数请求返回值body超过限制。 | 检查函数请求返回值。                 |
| 500 | FSS.4009 | function initialization exception.                                  | init 函数异常。       | 请检查 init 函数代码逻辑。           |
| 500 | FSS.4010 | invoke timed out after %s.                                          | 函数调用超时。          | 请检查函数超时的原因和配置的超时时间         |
| 500 | FSS.4011 | Initialization function execution times out.                        | 初始化函数执行超时        | 请检查函数初始化超时的原因              |
| 500 | FSS.4012 | load function timed out after %s.                                   | 函数加载超时。          | 请检查obs下载代码包是否成功，请联系相关运维人员。 |
| 500 | FSS.4013 | failed to check auth of the request.                                | 有状态请求鉴权失败。       | 请检查函数鉴权信息。                 |

| 状态码 | 错误码      | 错误信息                                                                | 描述                                            | 处理措施                                                       |
|-----|----------|---------------------------------------------------------------------|-----------------------------------------------|------------------------------------------------------------|
| 500 | FSS.4014 | reading the storeObject timeout.Use larger memory                   | 加载依赖包超时                                       | 请增加函数的内存规格                                                 |
| 500 | FSS.4020 | runtime dir xxx is forbidden.                                       | 用户chmod /home/snuser 和 /opt/function/runtime。 | 检查用户代码是否 chmod /home/snuser 和 /opt/function/runtime 这两个目录。 |
| 500 | FSS.4021 | trigger request of this tenant is limited at this frontend instance | 函数的并发数超过限制                                    | 请联系技术支持工程师协助解决                                             |
| 500 | FSS.4024 | function metadata is not found.                                     | 函数元信息不存在。                                     | 请检测是否存在此函数。                                                |
| 500 | FSS.4025 | can not create a state using a stateless function                   | 使用无状态函数创建状态实例                                 | 请检查函数是否为有状态函数                                              |
| 500 | FSS.4026 | instanceName is not exist                                           | 删除或查询有状态实例ID时，指定了未创建的实例名                      | 请检查有状态实例名是否已经成功创建                                          |
| 500 | FSS.4027 | cannot be created repeatedly instanceID                             | 使用了同一个实例名给同一个函数创建状态实例                         | 请检查函数是否使用同一个有状态实例名重复创建状态实例                                 |
| 500 | FSS.4140 | the size of request body is beyond maximum body size.               | 函数的请求体超过平台限制(默认6M)                            | 请检查函数的请求体大小是否超过限制，如果仍无法解决，请联系技术支持工程师协助解决                   |
| 500 | FSS.4201 | function initialization failed                                      | 用户函数初始化失败                                     | 请检查初始化函数代码                                                 |
| 500 | FSS.4202 | runtime bootstrap not found                                         | 函数找不到 bootstrap 文件                            | 请检查 bootstrap 脚本文件路径配置是否正确                                 |

| 状态码 | 错误码      | 错误信息                                                | 描述                    | 处理措施                                                 |
|-----|----------|-----------------------------------------------------|-----------------------|------------------------------------------------------|
| 500 | FSS.4203 | runtime process is exited                           | 用户 bootstrap 脚本未执行完退出 | 请检查 bootstrap 脚本是否存在异常退出的情况                          |
| 500 | FSS.4204 | The user bootstrap script syntax is incorrect.      | 用户 bootstrap 脚本语法错误   | 请根据错误提示信息检查 bootstrap 脚本代码。                          |
| 500 | FSS.4205 | runtime memory limit exceeded                       | 函数使用的内存总量超过限制         | 请检查函数代码占用的内存大小和配置的内存大小限制                             |
| 500 | FSS.4206 | failed to mount volumes for function                | 用户磁盘挂载失败              | 请检查代码中使用的磁盘信息（如：路径等）是否与配置信息一致。如果仍无法解决，请联系技术支持工程师协助解决 |
| 500 | FSS.4207 | disk usage exceed limit                             | 磁盘使用超出限额              | 请检查函数代码中是否超额使用磁盘存储。如果仍无法解决，请联系技术支持工程师协助解决            |
| 500 | FSS.4208 | function invocation canceled                        | 请求被取消                 | 用户调用了请求取消接口                                          |
| 500 | FSS.4218 | the function fails to be started for multiple times | 函数频繁拉起失败              | 检查函数代码                                               |
| 502 | FSS.0502 | Bad gateway.                                        | 网关错误                  | 联系开发人员                                               |
| 502 | FSS.1319 | Invalid trigger name.                               | 无效触发器名称               | 更换触发器名称                                              |
| 503 | FSS.0503 | Service unavailable.                                | 服务不可用                 | 联系开发人员                                               |
| 503 | FSS.1115 | The network is unreachable.                         | 网络不可达                 | 检查网络                                                 |
| 503 | FSS.1169 | The network is unreachable.                         | 网络不可达                 | 检查网络                                                 |

| 状态码 | 错误码      | 错误信息                                            | 描述              | 处理措施      |
|-----|----------|-------------------------------------------------|-----------------|-----------|
| 503 | FSS.1170 | Authentication error.                           | 鉴权失败, 请检查DDS的配置 | 检查DDS配置   |
| 503 | FSS.1401 | Failed to obtain the image information.         | 获取镜像信息失败        | 检查获取镜像的配置 |
| 503 | FSS.1402 | Failed to pull the image to create a container. | 加载镜像创建容器失败      | 检查镜像地址    |
| 503 | FSS.1403 | Failed to pull the image to delete a container. | 加载镜像删除容器失败      | 检查镜像地址    |
| 504 | FSS.0504 | Gateway timeout.                                | 网关超时            | 检查网络      |

## 9.3 获取账号、IAM 用户、项目、用户组、区域、委托的名称和 ID

### 获取账号、IAM 用户的名称和 ID

- 从控制台获取账号名、账号ID、用户名、用户ID、项目名称、项目ID  
在调用接口时，部分URI中需要填入用户名、用户ID、账号名、账号ID、项目名称、项目ID，在“我的凭证”页面可以获取这些信息。
  - 登录华为云管理控制台。
  - 单击用户名，在下拉列表中单击“我的凭证”。
  - 在“API凭证”页面的项目列表中查看账号名、账号ID、用户名、用户ID、项目名称、项目ID。

图 9-1 查看帐号名、帐号 ID、用户名、用户 ID、项目名称、项目 ID



- 调用API获取用户ID、项目ID
  - 获取用户ID请参考：[管理员查询IAM用户列表](#)。



- 获取项目ID请参考：[查询指定条件下的项目列表](#)。

### 获取用户组名称和 ID

1. 登录IAM控制台，在左侧导航栏选择“用户组”。
2. 单击需要查询的用户组前的“ $\vee$ ”，即可查询用户组名称、用户组ID。

### 获取区域 ID

1. 登录IAM控制台，在左侧导航栏选择“项目”。
2. “项目”列的内容即为所属区域对应的ID。

### 获取委托名称和 ID

1. 登录IAM控制台，选择“委托”页签。
2. 鼠标移动到需要查询名称和ID的委托上，气泡框中出现的第一行为委托名称，第二行为委托ID。

图 9-2 查看委托 ID



## 9.4 FunctionGraph 服务监控指标说明

### 功能说明

本节定义了FunctionGraph服务中“函数”上报云监控服务的监控指标。

指标的命名空间和维度，您可以通过云监控服务提供的API接口来检索FunctionGraph服务中“函数”产生的监控指标和告警信息。

### 命名空间

SYS.FunctionGraph

### 监控指标

表 9-2 监控指标

| 指标名称  | 显示名  | 描述            | 单位 | 上限值 | 下限值 | 建议阈值 | 值类型 | 指标意义          | 所属维度                 |
|-------|------|---------------|----|-----|-----|------|-----|---------------|----------------------|
| count | 调用次数 | 该指标用于统计函数调用次数 | 次数 | -   | 0   | -    | Int | 该指标用于统计函数调用次数 | package-functionname |

| 指标名称        | 显示名    | 描述                | 单位 | 上限值 | 下限值 | 建议阈值 | 值类型 | 指标意义              | 所属维度                 |
|-------------|--------|-------------------|----|-----|-----|------|-----|-------------------|----------------------|
| failcount   | 错误次数   | 该指标用于统计函数调用错误次数   | 次数 | -   | 0   | -    | Int | 该指标用于统计函数调用错误次数   | package-functionname |
| rejectcount | 被拒绝次数  | 该指标用于统计函数调用被拒绝次数  | 次数 | -   | 0   | -    | Int | 该指标用于统计函数调用被拒绝次数  | package-functionname |
| duration    | 平均运行时间 | 该指标用于统计函数调用平均运行时间 | 毫秒 | -   | 0   | -    | Int | 该指标用于统计函数调用平均运行时间 | package-functionname |
| maxDuration | 最大运行时间 | 该指标用于统计函数调用最大运行时间 | 毫秒 | -   | 0   | -    | Int | 该指标用于统计函数调用最大运行时间 | package-functionname |
| minDuration | 最小运行时间 | 该指标用于统计函数调用最小运行时间 | 毫秒 | -   | 0   | -    | Int | 该指标用于统计函数调用最小运行时间 | package-functionname |

## 维度

表 9-3 维度

| Key                  | Value     |
|----------------------|-----------|
| package-functionname | 应用名称-函数名称 |

# 10 修订记录

表 10-1 修订记录

| 发布日期       | 修订记录                                                                                                                                          |
|------------|-----------------------------------------------------------------------------------------------------------------------------------------------|
| 2022-11-21 | 新增接口内容如下：<br>分页查询指定函数流记录                                                                                                                      |
| 2022-11-14 | 新增接口内容如下：<br>创建依赖包版本<br>获取依赖包版本列表<br>获取依赖包版本详情<br>删除依赖包版本<br>获取按指定指标排序的函数列表<br>获取函数预留实例配置列表<br>获取函数预留实例数量<br>修改接口内容如下：<br>下线“函数异步执行并返回预留实例ID” |
| 2022-09-07 | 新增错误码：<br>FSS.4208<br>新增接口内容如下：<br>修改函数预留实例数量                                                                                                 |
| 2022-07-29 | 新增错误码：<br>FSS.1206<br>新增接口内容如下：<br>函数异步执行并返回预留实例ID<br>更新函数最大实例数<br>设置函数异步配置信息<br>停止函数异步调用请求                                                   |

| 发布日期       | 修订记录                                                                                                                                                                                                                                                                                                                                                                                                             |
|------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 2022-04-29 | 新增接口内容如下：<br>同步执行函数流                                                                                                                                                                                                                                                                                                                                                                                             |
| 2022-04-02 | 新增接口内容如下：<br>异步执行工作流<br>删除函数流<br>查询函数流<br>创建函数流<br>获取指定函数流执行实例列表<br>获取指定函数流执行实例<br>获取指定函数流实例的元数据<br>修改指定函数流实例的元数据<br>获取函数流指标<br>获取指定函数流指标<br>重试函数流<br>停止函数流                                                                                                                                                                                                                                                      |
| 2022-01-21 | 接口修改内容如下：<br>函数管理域 <ul style="list-style-type: none"> <li>● 新增获取函数异步调用请求列表接口。</li> <li>● 新增修改函数调用链配置接口。</li> <li>● 新增获取函数调用链配置接口。</li> </ul>                                                                                                                                                                                                                                                                     |
| 2020-08-29 | 修改内容如下： <ul style="list-style-type: none"> <li>● 新增获取依赖包列表接口。</li> <li>● 新增创建依赖包接口。</li> <li>● 新增删除指定的依赖包接口。</li> <li>● 新增获取指定依赖包接口。</li> <li>● 新增更新指定依赖包接口。</li> <li>● 新增获取指定函数的测试事件列表接口。</li> <li>● 新增创建测试事件接口。</li> <li>● 新增删除指定测试事件接口。</li> <li>● 新增获取测试事件详细信息接口。</li> <li>● 新增更新测试事件详细信息接口。</li> <li>● 新增查询租户配额接口。</li> <li>● 新增获取指定时间段的函数运行指标接口。</li> <li>● 新增租户函数统计信息接口。</li> <li>● 新增更新触发器接口。</li> </ul> |
| 2020-06-16 | 新增API Explorer跳转链接。                                                                                                                                                                                                                                                                                                                                                                                              |

| 发布日期       | 修订记录                                                         |
|------------|--------------------------------------------------------------|
| 2020-03-30 | 删除工作流的API接口。                                                 |
| 2019-05-27 | 按照V3.0版本模板调整API参考，不涉及接口变化。                                   |
| 2018-11-16 | 函数实现与云监控服务的对接，新增FunctionGraph监控指标说明。                         |
| 2018-8-29  | 函数新增禁用功能，相关内容修改。                                             |
| 2018-8-20  | 1. 支持的运行时新增Node.js 8.10，增加相关内容。<br>2. 触发器类型新增LTS、CTS，增加相关内容。 |
| 2018-6-30  | 增加FunctionGraph函数相关的API接口。                                   |
| 2018-5-15  | 修改文档结构。                                                      |
| 2018-02-28 | 第一次正式发布。                                                     |