



函数 workflow

API 参考

发布日期 2021-01-19

目录

1 使用前必读	1
1.1 概述	1
1.2 调用说明	1
1.3 终端节点	1
1.4 约束限制	1
1.5 基本概念	1
2 API 概览	3
3 如何调用 API	5
3.1 构造请求	5
3.2 认证鉴权	7
3.3 返回结果	8
4 函数模型定义	10
4.1 FunctionGraph 函数模型	10
4.2 函数 Trigger Management 触发器模型	15
5 函数管理域接口	20
5.1 获取函数列表	20
5.2 获取函数的 metadata	25
5.3 获取指定函数代码	29
5.4 创建函数	32
5.5 删除函数/版本	37
5.6 修改函数代码	38
5.7 修改函数的 metadata 信息	42
5.8 发布函数版本	47
5.9 获取指定函数的版本列表	51
5.10 创建函数版本别名	55
5.11 修改函数版本别名信息	57
5.12 删除函数版本别名	59
5.13 获取函数版本的指定别名信息	60
5.14 获取指定函数所有版本别名列表	61
5.15 获取指定函数的所有触发器	63
5.16 获取指定触发器的信息	64
5.17 删除指定函数的所有触发器	66

5.18 创建触发器.....	67
5.19 删除触发器.....	69
6 函数数据域接口.....	71
6.1 同步执行函数.....	71
6.2 异步执行函数.....	73
7 权限及授权项说明.....	75
8 附录.....	77
8.1 状态码.....	77
8.2 错误码.....	78
8.3 获取项目 ID.....	88
8.4 FunctionGraph 服务监控指标说明.....	89
9 修订记录.....	91

1 使用前必读

1.1 概述

欢迎使用函数 workflow (FunctionGraph)。FunctionGraph 是一项基于事件驱动的函数托管计算服务。通过函数 workflow，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](#)。

1.2 调用说明

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](#)。

1.3 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。

1.4 约束限制

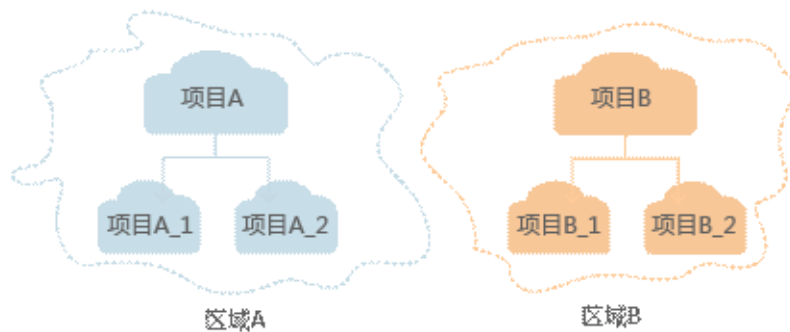
- 您能创建的函数数量与配额有关系，具体请参见 [关于配额](#)。
- 更详细的限制请参见具体 API 的说明。

1.5 基本概念

- 账号
您注册公有云时的账号，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他们进行日常管理工作。

- 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）。
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
公有云的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



2 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 2-1 接口简介

类型	说明
函数管理域接口	函数的管理域接口，包括函数获取、创建、修改、发布等。
函数数据域接口	函数的数据域接口，包括同步执行函数和异步执行函数。

函数管理域接口

表 2-2 管理域接口

API	说明
获取函数列表	获取函数列表。
获取函数的 metadata	获取指定函数的metadata。
获取指定函数代码	获取指定函数的代码。
创建函数	创建指定的函数。
删除函数/版本	删除指定的函数或者特定的版本（不允许删除latest版本）。
修改函数代码	修改指定的函数的代码。
修改函数的 metadata信息	修改指定的函数的metadata信息。
发布函数版本	发布函数版本。

API	说明
获取指定函数的版本列表	获取指定函数的版本列表。
创建函数版本别名	创建函数版本别名。
修改函数版本别名信息	修改函数版本别名信息。
删除函数版本别名	删除函数版本别名。
获取函数版本的指定别名信息	获取函数指定的版本别名信息。
获取指定函数所有版本别名列表	获取函数版本别名列表。
获取指定函数的所有触发器	获取指定函数的所有触发器设置。
获取指定触发器的信息	获取特定触发器的信息。
删除指定函数的所有触发器	删除指定函数所有触发器设置。
创建触发器	创建触发器。
删除触发器	删除触发器。

函数数据域接口

表 2-3 数据域接口

API	说明
同步执行函数	同步执行函数。
异步执行函数	异步执行函数。

3 如何调用 API

3.1 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。

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。

表 3-1 URI 中的参数说明

参数	描述
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 获取。 例如IAM服务在“ae-ad-1”区域的Endpoint为“iam.ae-ad-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。

📖 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，同一个服务的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.ae-ad-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

公有云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.ae-ad-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，***xxxxxxx*** 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.ae-ad-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": "xxxxxxxxxxxxxxxxxxxxxx"
      }
    }
  }
}
```

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。

3.2 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。

在[构造请求](#)中以调用获取用户Token接口为例说明了如何调用API。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.ae-ad-1.myhuaweicloud.com/v3/auth/projects
Content-Type: application/json
X-Auth-Token: ABCDEFJ....
```

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。

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。

3.3 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3-1](#)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。

图 3-1 获取用户 Token 响应消息头

```
connection → keep-alive
content-type → application/json
date → Tue, 12 Feb 2019 06:52:13 GMT
server → Web Server
strict-transport-security → max-age=31536000; includeSubdomains;
transfer-encoding → chunked
via → proxy A
x-content-type-options → nosniff
x-download-options → noopen
x-frame-options → SAMEORIGIN
x-iam-trace-id → 218d45ab-d674-4995-af3a-2d0255ba41b5

x-subject-token
→ MfYXQYJKoZiHvcNAQcCoIVTjCCGEOCAQExDTALBglghkgBZQMEAgEwgharBgkqhkiG9w0BwGggghacBmHsidG9rZW4iOnsiZ0hwaXUlc19hdCI6IjwMTktMDtMTNUMC
j3Ks6YgKnpVNRbW2eZ5eb785Z0kqACgkIQ1w4JIGzrpdl8LGXXSbdldq4lqHCYb8P4NaYONyEjcAgz/VefYtLWT1GSO0zxKZmiQHq82HBqHdgIZO9fuEbl5dMhdavj+33wEi
dRCE987o+k9-
+CMZSEB7bUGdSuj6eRASXl1jpPEGA270g1FruooL6jagjFkNPQuFSOU8+uSstVwRtNfsC+qTp22Rkd5MCqFGQ8LcuUxC3a+9CM8nOintWW7oeRUvhVpxk8pxX1wTEboX-
RzT6MUbvpvGw-cPNFYxjECKnoH3HRozv0vN--n5d6Nbsg==

x-ssr-protection → 1; mode=block;
```

响应消息体

响应消息体通常以结构化格式返回，与响应消息头中Content-type对应，传递除响应消息头之外的内容。

对于获取用户Token接口，返回如下消息体。为篇幅起见，这里只展示部分内容。

```
{
  "token": {
    "expires_at": "2019-02-13T06:52:13.855000Z",
    "methods": [
      "password"
    ],
    "catalog": [
      {
        "endpoints": [
          {
            "region_id": "XXXXXXXX",
            .....

```

当接口调用出错时，会返回错误码及错误信息说明，错误响应的Body体格式如下所示。

```
{
  "error_code": "FGS.0111",
  "error_msg": "xxxxxxxx"
}
```

其中，error_code表示错误码，error_msg表示错误描述信息。

4 函数模型定义

4.1 FunctionGraph 函数模型

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4-1所示。

表 4-1 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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
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。最小值为128, 最大值为4096。
cpu	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), 函数占用的CPU为基础CPU: 200 millicores, 再加上内存按比例占用的CPU, 计算方法: 内存/128 * 100 + 200。

名称	说明
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4-2 。
depend_list	依赖包列表。
strategy_config	函数策略配置，请参考 表4-3 。
extend_config	函数扩展配置。
dependencies	依赖代码包列表，请参考 表4-5 。

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4-4。
mount_config	文件系统配置，请参考表4-6。

表 4-2 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 4-3 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 4-4 func_vpc 参数说明

参数	类型	是否必选	说明
vpc_name	String	-	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	-	子网名称
subnet_id	String	当func_vpc非空时必选	子网编号
cidr	String	-	子网掩码
gateway	String	-	网关

表 4-5 dependency 参数说明

参数	类型	是否必选	说明
owner	String	-	依赖包属主的domainId
link	String	-	依赖包在OBS上的链接
runtime	String	-	依赖包语言类型（仅作为分类条件）
etag	String	-	依赖包md5值
size	Int	-	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）

表 4-6 mount_config 参数说明

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

表 4-7 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的非0整数）
user_group_id	Int	mount_user非空时必选	用户组ID（-1 ~ +65534的非0整数）

表 4-8 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(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格式如下。

```
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>。

4.2 函数 Trigger Management 触发器模型

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4-9](#)所示。

表 4-9 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、OBS、TIMER、CTS、kafka。
display_name	触发器类型显示值。
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。
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4-10所示。

表 4-10 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、OBS、TIMER、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4-11所示。

表 4-11 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。

- OBS触发器数据如下。

```
{
  "bucket": "yourBucketName",
  "events": ["s3:ObjectCreated:Put"],
  "prefix": "yourPrefix",
  "suffix": "yourSuffix"
}
```

表 4-12 OBS 触发器数据说明

字段名称	字段说明
bucket	OBS桶名，必填。
events	OBS触发事件集合，取值有如下情况： ["s3:ObjectCreated:*","s3:ObjectCreated:Put","s3:ObjectCreated:Post","s3:ObjectCreated:Copy","s3:ObjectCreated:CompleteMultipartUpload","s3:ObjectRemoved:*","s3:ObjectRemoved:DeleteMarkerCreated","s3:ObjectRemoved:Delete"]，必填。 其中“s3:objectcreated:*”包含所有其他以“s3:objectcreated”开头的所有事件，“s3:objectremoved:*”包含所有以“s3:objectremoved”开头的 所有事件。
prefix	OBS对象的前缀，选填。
suffix	OBS对象的后缀，选填。

- 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4-13所示。

表 4-13 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。

- 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4-14所示。

表 4-14 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 4-15 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",
  "db_name": "string",
  "collection_name": "string",
  "db_user": "string",
  "db_password": string,
  "batch_size": int,
}
```

表 4-16 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
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

5 函数管理域接口

5.1 获取函数列表

功能介绍

获取函数列表。

URI

GET /v2/{project_id}/fgs/functions?marker={marker}&maxitems={maxitems}

URI参数说明如[表5-1](#)所示。

表 5-1 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	租户Project ID

请求消息

请求参数说明如[表5-2](#)所示。

表 5-2 请求参数说明

参数	类型	必选	说明
marker	Int	可选	上一次查询到的最后的记录位置。
maxitems	Int	可选	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。

响应消息

响应参数说明如表5-3所示。

表 5-3 响应参数说明

名称	参数类型	说明
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。
runtime	String	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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
timeout	Int	函数执行超时时间, 超时函数将被强行停止, 范围3~900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。
memory_size	Int	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	Int	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), 函数占用的CPU为基础CPU: 200 millicores, 再加上内存按比例占用的CPU, 计算方法: 内存/128 * 100 + 200。
code_type	String	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。

名称	参数类型	说明
code_url	String	<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	String	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	Int64	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。
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_code	String	函数代码，请参考 表5-4 。
depend_list	[]String	依赖包列表。
strategy_config	String	函数策略配置，请参考 表5-5 。
extend_config	String	函数扩展配置。
dependencies	[]*String	依赖代码包。

名称	参数类型	说明
initializer_handler	String	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	Int	初始化超时时间，超时函数将被强行停止，范围1 ~ 300秒。
func_vpc	*String	VPC配置，请参考表5-6。
mount_config	*String	磁盘挂载配置，请参考表5-7

表 5-4 func_code 参数说明

参数	类型	说明
file	String	函数代码内容（已废弃）。
link	String	函数代码链接（可通过OBS SDK下载）。

表 5-5 strategy_config 参数说明

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

表 5-6 func_vpc 参数说明

参数	类型	是否必选	说明
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。

表 5-7 mount_config 参数说明

参数	类型	是否必选	说明
mount_user	*String	必选	磁盘挂载用户配置，请参考表5-8。
func_mounts	[]*String	必选	磁盘挂载函数配置，请参考表5-9。

表 5-8 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的非0整数)
user_group_id	Int	mount_user非空时必选	用户组ID (-1 ~ +65534的非0整数)

表 5-9 func_mounts 参数说明

参数	类型	是否必选	说明
id	String	-	唯一编号
mount_type	String	func_mounts非空时必选	挂载类型(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”，必选	远端挂载路径 (例如192.168.0.12:/data)
local_mount_path	String	func_mounts非空时必选	函数访问路径
status	String	-	状态标识取值ACTIVE/DISABLED

示例

请求示例

```
GET /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions?marker=0&maxitems=400
HTTP/1.1
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200 OK
{
```

```
"functions": [  
  {  
    "func_urn": "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",  
    "func_name": "test",  
    "user_domain": "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":  
    "decfce6939297b0b5ec6d1a23bf9c725870f5e69fc338a89a6a4029264688dc26338f56d08b6535d  
e47f15ad538e22ca66613b9a46f807d50b687bb53fded1c6",  
    "version": "latest",  
    "image_name": "latest-5qe8e",  
    "xrole": "cff",  
    "description": "111",  
    "last_modified": "2018-03-28T11:30:32+08:00",  
    "func_code": {},  
    "strategy_config": {  
      "concurrency": -1,  
      "initializer_handler": "index.initializer",  
      "initializer_timeout": 3  
    }  
  },  
  "next_marker": 45  
]
```

失败时的返回格式:

```
HTTP/1.1 403 Forbidden  
{  
  "error_code": "FSS.0403",  
  "error_msg": "namespace and token mismatch"  
}
```

状态码

状态码请参见[状态码](#)。

5.2 获取函数的 metadata

功能介绍

获取指定函数的metadata。

URI

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

URI参数说明如[表5-10](#)所示。

表 5-10 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，详细解释见 FunctionGraph函数模型 的描述。

请求消息

无

响应消息

响应参数说明如[表5-11](#)所示。

表 5-11 响应参数说明

名称	参数类型	说明
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。
runtime	String	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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
timeout	Int	函数执行超时时间，超时函数将被强行停止，范围3 ~ 900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。
memory_size	Int	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。

名称	参数类型	说明
cpu	Int	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），函数占用的CPU为基础CPU：200 millicores，再加上内存按比例占用的CPU，计算方法：内存/128 *100 + 200。
code_type	String	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	String	<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	String	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	Int64	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。
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。

名称	参数类型	说明
last_modified	String	函数最后一次更新时间。
depend_list	[]String	依赖包列表。
strategy_config	String	函数策略配置，请参考 表5-12 。
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“.”。 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	Int	初始化超时时间，超时函数将被强行停止，范围1~300秒。
func_vpc	*String	VPC配置，请参考 表5-6 。
mount_config	*String	磁盘挂载配置，请参考 表5-7 。

表 5-12 strategy_config 参数说明

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

示例

请求示例

```
GET /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/config HTTP/1.1
```

响应示例

成功时的返回格式：

```
HTTP/1.1 200 OK
{
  "code_filename": "index.js",
  "code_size": 272,
  "code_type": "inline",
  "cpu": 300,
  "description": "111",
  "digest":
    "decbce6939297b0b5ec6d1a23bf9c725870f5e69fc338a89a6a4029264688dc26338f56d08b653de47f15ad538e22ca66613b9a46f807d50b687bb53fdded1c6",
  "func_name": "test",
  "func_urn": "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest",
  "handler": "test.handler",
```

```
"image_name": "latest-5qe8e",
"last_modified": "2018-03-28T11:30:32+08:00",
"memory_size": 128,
"namespace": "7aad83af3e8d42e99ac194e8419e2c9b",
"package": "default",
"project_name": "xxxxxxxx",
"runtime": "Node.js6.10",
"timeout": 3,
"user_domain": "cff01_hk",
"version": "latest",
"xrole": "cff",
"strategy_config": {
  "concurrency": -1
},
}
```

失败时的返回格式（示例）：

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1051",
  "error_msg": "Not found the function"
}
```

状态码

状态码请参见[状态码](#)。

5.3 获取指定函数代码

功能介绍

获取指定函数的代码。

URI

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

URI参数说明如[表5-13](#)所示。

表 5-13 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，详细解释见 FunctionGraph函数模型 的描述。

请求消息

无

响应消息

响应参数说明如表5-14所示。

表 5-14 响应参数说明

名称	参数类型	说明
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。
runtime	String	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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
concurrency	Int	<ul style="list-style-type: none"> 0: 函数被禁用。 -1: 函数被启用。
code_type	String	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	String	<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	String	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	String	函数大小, 单位: 字节。
func_code	String	函数代码, 请参考表5-15。
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。
depend_list	String	依赖包列表。
strategy_config	String	函数策略配置, 请参考表5-16。

名称	参数类型	说明
func_vpc	func_vpc	VPC配置, 请参考表5-6。

表 5-15 func_code 参数说明

参数	类型	说明
file	String	函数代码内容 (已废弃)。
link	String	函数代码链接 (可通过OBS SDK下载)。

表 5-16 strategy_config 参数说明

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

示例

请求示例

```
GET /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/code HTTP/1.1
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200 OK
{
  "code_filename": "index.js",
  "code_size": 272,
  "code_type": "inline",
  "digest":
"decbce6939297b0b5ec6d1a23bf9c725870f5e69fc338a89a6a4029264688dc26338f56d08b6535de47f15ad538e22ca66613b9a46f807d50b687bb53fded1c6",
  "func_code": {
    "file": "",
    "link": "https://functionstorage-hk06.obs.xx-xxx.xxxxxxxxcloud.com/xxx/d2b0xxxf6e65/default/test143/latest/index.zip"
  },
  "func_name": "test",
  "func_urn": "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest",
  "last_modified": "2018-03-28T11:30:32+08:00",
  "runtime": "Node.js6.10",
  "strategy_config": {
    "concurrency": -1
  },
}
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1052",
  "error_msg": "Not found the function version"
}
```

状态码

状态码请参见[状态码](#)。

5.4 创建函数

功能介绍

创建指定的函数。

URI

POST /v2/{project_id}/fgs/functions

URI参数说明如[表5-17](#)所示。

表 5-17 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID

请求消息

请求参数说明如[表5-18](#)所示。

表 5-18 请求参数说明

参数	类型	必选	说明
func_name	String	必选	函数名称。
package	String	必选	函数所属的分组，用户自定义名称，可默认填写：default。
code_type	String	必选	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

参数	类型	必选	说明
code_url	String	可选	<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。
description	String	可选	函数的描述。
code_filename	String	可选	代码的文件名。 <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。
handler	String	必选	函数执行入口。
memory_size	Int	必选	函数消耗的内存，单位M。 取值范围：128、256、512、768、1024、1280、1536、1792、2048、2560、3072、3584、4096。 最小值为128，最大值为4096。
runtime	String	必选	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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
timeout	Int	必选	函数的超时时间，范围3~900秒。
user_data	String	可选	用户自定义的name/value信息。
role	String	可选	如函数需要访问其他云服务，则必填。
func_code_file	String	必选	函数代码内容。 <ul style="list-style-type: none"> 当“code_type”为“inline”、“zip”或“jar”时，必选，且代码必须要进行base64编码。 当“code_type”为“obs”时，可选。

响应消息

响应参数说明如表5-19所示。

表 5-19 响应参数说明

名称	参数类型	说明
func_urn	String	函数的URN (Uniform Resource Name)，唯一标识函数。

名称	参数类型	说明
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。
runtime	String	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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
timeout	Int	函数执行超时时间，超时函数将被强行停止，范围3 ~ 900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。
memory_size	Int	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	Int	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），函数占用的CPU为基础CPU：200 millicores，再加上内存按比例占用的CPU，计算方法：内存/128 * 100 + 200。
code_type	String	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	String	<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。

名称	参数类型	说明
code_filename	String	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	Int64	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。
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	*String	VPC配置，请参考 表5-6 。
func_code	String	函数代码，请参考 表5-4 。
depend_list	[]String	依赖包列表。
strategy_config	String	函数策略配置，请参考 表5-20 。
extend_config	String	函数扩展配置。
dependencies	[]*String	依赖代码包。
initializer_handler	String	函数初始化入口。
initializer_timeout	Int	函数的初始化时间，范围1 ~ 300秒。

表 5-20 strategy_config 参数说明

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

示例

请求示例

```
POST /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions HTTP/1.1
{
  "func_name": "test",
  "package": "default",
  "description": "",
  "handler": "test.handler",
  "memory_size": 128,
  "timeout": 3,
  "runtime": "Python2.7",
  "user_data": "",
  "code_type": "inline",
  "func_code": {
    "file":
"aW1wb3J0IGpzb24KZGVmlGhhbmRsZXIlgKGV2ZW50LCBjb250ZXh0KToKICAgIG91dHB1dCA9ICdl
ZWxsbyBtZXNzYWdlOiAnIAnIcsganNvbi5kdW1wcyhldmVudCkKICAgIHJldHVybiBvdXRwdXQ="
  }
}
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200 OK
{
  "func_urn":
"urn:fss:xxxxxxx:c3b2459a6d5e4b548e6777e57852692d:function:default:TestCreateFunctionInP
ythonSdk:latest",
  "func_name": "TestCreateFunctionInPythonSdk",
  "user_domain": "FGS_hwx559619",
  "namespace": "c3b2459a6d5e4b548e6777e57852692d",
  "project_name": "xxxxxxx",
  "package": "default",
  "runtime": "Python3.6",
  "timeout": 30,
  "handler": "index.handler",
  "memory_size": 128,
  "cpu": 300,
  "code_type": "inline",
  "code_filename": "index.py",
  "code_size": 110,
  "digest":
"1c8610d1870731a818a037f1d2adf3223e8ac351aeb293fb1f8eabd2e9820069a61ed8b5d38182e7
60adc33a307d0e957afc357f415cd8c9c3ff6f0426fd85cd",
  "version": "latest",
  "image_name": "latest-0zf5g",
  "last_modified": "2019-03-07T18:37:19+08:00",
  "concurrency": 0,
  "strategy_config": {
```

```
"concurrency": -1
},
}
```

失败时的返回格式:

```
HTTP/1.1 409 Conflict
{
  "error_code": "FSS.1061",
  "error_msg": "The function has existed"
}
```

状态码

状态码请参见[状态码](#)。

5.5 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，包含所有版本或者别名以及触发器。

URI

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

URI参数说明如[表5-21](#)所示。

表 5-21 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参考 FunctionGraph函数模型 。 说明 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:tes t。

请求消息

无

响应消息

无

示例

请求示例

```
DELETE /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:v20170830-181539 HTTP/1.1
```

响应示例

成功时的返回格式:

```
HTTP/1.1 204
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1051",
  "error_msg": "Not found the function"
}
```

状态码

状态码请参见[状态码](#)。

5.6 修改函数代码

功能介绍

修改指定的函数的代码。

URI

```
PUT /v2/{project_id}/fgs/functions/{function_urn}/code
```

URI参数说明如[表5-22](#)所示。

表 5-22 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参考 表4-1 。

请求消息

请求参数说明如[表5-23](#)所示。

表 5-23 请求参数说明

参数	类型	必选	说明
code_type	String	必选	函数代码类型，请参考表4-1。
code_url	String	可选	函数代码包在OBS上的地址。 当“code_type”为“obs”时，该字段必选。
func_code.file	String	可选	函数代码内容。 <ul style="list-style-type: none"> 当“code_type”为“inline”、“zip”或“jar”时，必选，且代码必须要进行base64编码。 当“code_type”为“obs”时，可选。
depend_list	[]*String	可选	函数代码依赖包。
code_filename	String	可选	函数文件名，前面为文件名，后面为文件类型

响应消息

响应参数说明如表5-24所示。

表 5-24 响应参数说明

名称	参数类型	说明
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。
runtime	String	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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
concurrency	Int	<ul style="list-style-type: none"> 0：函数被禁用。 -1：函数被启用。
code_type	String	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

名称	参数类型	说明
code_url	String	<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	String	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	String	函数大小，单位：字节。
func_code	String	函数代码，请参考 表5-25 。
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。
depend_list	[]String	依赖包列表。
strategy_config	String	函数策略配置，请参考 表5-26 。
dependencies	[]dependency	依赖代码包。
func_vpc	func_vpc	VPC配置，请参考 表5-6 。

表 5-25 func_code 参数说明

参数	类型	说明
file	String	函数代码内容（已废弃）。
link	String	函数代码链接（可通过OBS SDK下载）。

表 5-26 strategy_config 参数说明

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

示例

请求示例

```
PUT /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxx:
7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/code HTTP/1.1
{
  "code_type": "inline",
  "func_code": {
    "file":
"aW1wb3J0IGpzb24KZGVmIGhhbmRsZXIoZXZlbnQsIGNvbnRleHQpOgogICAgb3V0cHV0ID0gJ0hl
bGxvIE1zZyBmb3JgbW9kaWZ5OiAnIChsganNvbi5kdW1wcyhldmVudCkKICAgIGFrID0gY29udGV4d
C5nZXRBY2Nlc3NLZXkoKQogICAgc2sgPSBjb250ZXh0LmdldFNlY3JldEtleSgpCiAgICB0b2t1biA9IGN
vbnRleHQwZ2V0VG9rZW4oKQogICAgcHJpbnQgJ2FrOicgKyBhawogICAgcHJpbnQgJ3NrOicgKyBza
wogICAgcHJpbnQgJ3Rva2VuOicgKyB0b2t1bgogICAgcmV0dXJlIG91dHB1dAo=",
  },
  "strategy_config": {
    "concurrency": -1
  },
}
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200 OK
{
  "code_filename": "index.js",
  "code_size": 273,
  "code_type": "inline",
  "code_url": "",
  "digest":
"af40294713c964d24f52fd567022cb7e03373b8acfafc2526bacde08a864e21dd214ad4fe567cd8a6
541822ee76171ca802da6e7d135c07689a6072930e09824",
  "func_code": {
    "file": "",
    "link": "https://functionstorage-hk06.obs.xx-xxx.xxxxxxxxcloud.com/xxx/d2b0xxxf6e65/default/
test143/latest/index.zip"
  },
  "func_name": "test",
  "func_urn": "urn:fss:xxxxxxx:
7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest",
  "last_modified": "2018-02-26T11:55:41+08:00",
  "runtime": "Node.js6.10",
  "concurrency": 0,
  "depend_list": [],
  "strategy_config": {
    "concurrency": -1
  },
  "dependencies": [],
  "func_vpc": null
}
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1052",
  "error_msg": "Not found the function version"
}
```

状态码

状态码请参见[状态码](#)。

5.7 修改函数的 metadata 信息

功能介绍

修改指定的函数的metadata信息。

URI

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

URI参数说明如[表5-27](#)所示。

表 5-27 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参考 FunctionGraph函数模型 。

请求消息

请求参数说明如[表5-28](#)所示。

表 5-28 请求参数说明

参数	类型	必选	说明
runtime	String	可选	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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
code_type	String	必选	函数代码类型，请参考 表4-1 。
code_url	String	可选	函数代码包在OBS上的地址，当“code_type”为“obs”时必选。
description	String	可选	函数的描述。
handler	String	可选	函数执行入口，请参考 表4-1 。
memory_size	Int	可选	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。
timeout	Int	可选	函数的超时时间。

参数	类型	必选	说明
user_data	String	可选	用户自定义的name/value参数。
xrole	String	可选	如函数需要访问其他云服务，则必填。
app_xrole	*String	可选	如函数需要访问其他云服务app，则必填。
initializer_handler	String	可选	函数初始化入口。
initializer_timeout	Int	可选	函数的初始化时间，范围1 ~ 300秒。
func_vpc_subnet_id	String	可选	虚拟私有云子网编号。
func_vpc_vpc_id	String	可选	虚拟私有云唯一标识。
mount_config	mount_config	可选	文件系统配置，请参考 表4-6 。

响应消息

响应参数说明如[表5-29](#)所示。

表 5-29 响应参数说明

名称	参数类型	说明
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。
runtime	String	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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
timeout	Int	函数执行超时时间，超时函数将被强行停止，范围3 ~ 900秒。

名称	参数类型	说明
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。
memory_size	Int	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	Int	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），函数占用的CPU为基础CPU：200 millicores，再加上内存按比例占用的CPU，计算方法：内存/128 * 100 + 200。
code_type	String	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	String	<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	String	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	Int64	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。
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。

名称	参数类型	说明
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_code	String	函数代码，请参考表5-4。
depend_list	[]String	依赖包列表。
strategy_config	String	函数策略配置，请参考表5-30。
extend_config	String	函数扩展配置。
dependencies	[]*String	依赖代码包。
initializer_handler	String	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	Int	初始化超时时间，超时函数将被强行停止，范围1 ~ 300秒。
func_vpc	func_vpc	VPC配置，请参考表5-6。
mount_config	mount_config	文件系统配置，请参考表4-6

表 5-30 strategy_config 参数说明

参数	类型	必选	说明
concurrency	Int	必选	0: 函数被禁用。 -1: 函数被启用。

示例

请求示例

```
PUT
/v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxx:
```



```
7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/config HTTP/1.1
{
  "description": "",
  "handler": "test.handler",
  "memory_size": 128,
  "timeout": 3,
  "runtime": "Python",
  "user_data": "",
  "code_type": "inline",
  "func_code": {
    "file":
"aW1wb3J0IGpzY24KZGVmIGhhbmRsZXIgaGVhZGVzZW50LCBjb250ZXh0KToKICAgIG91dHB1dCA9ICdl
ZWxsbyBtZXNzYWdlOiAnIiCsganNvbi5kdW1wcyhldmVudCkKICAgIHJldHVybiBvdXRwdXQ="
  },
  "xrole": "cffservice"
}
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200
{
  "func_urn": "urn:fs:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",
  "func_name": "test",
  "user_domain": "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":
"decfce6939297b0b5ec6d1a23bf9c725870f5e69fc338a89a6a4029264688dc26338f56d08b6535d
e47f15ad538e22ca66613b9a46f807d50b687bb53fded1c6",
  "version": "latest",
  "image_name": "latest-5qe8e",
  "xrole": "cff",
  "last_modified": "2018-03-28T11:30:32+08:00",
  "strategy_config": {
    "concurrency": -1
  },
  "initializer_handler": "index.initializer",
  "initializer_timeout": 3
}
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1051",
  "error_msg": "Not found the function"
}
```

状态码

状态码请参见[状态码](#)。

5.8 发布函数版本

功能介绍

发布函数版本。

以最新latest代码发布版本，版本号系统自动生成，一个函数最多只能有10个版本。

URI

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

URI参数说明如[表5-31](#)所示。

表 5-31 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参考 FunctionGraph函数模型 。

请求消息

请求参数说明如[表5-32](#)所示。

表 5-32 请求参数说明

参数	类型	必选	说明
digest	String	可选	待发布函数的代码的digest。 如果不提供，则使用latest版本的代码进行发布。
version	String	可选	待发布的版本号。 如果未指定，则会使用当前时间格式化为“yyyymmdd-HHMMSS”的字符串作为发布版本号。
description	String	可选	待发布的版本描述。

响应消息

响应参数说明如[表5-33](#)所示。

表 5-33 响应参数说明

名称	参数类型	说明
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。
runtime	String	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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
timeout	Int	函数执行超时时间, 超时函数将被强行停止, 范围3 ~ 900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。
memory_size	Int	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	Int	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), 函数占用的CPU为基础CPU: 200 millicores, 再加上内存按比例占用的CPU, 计算方法: 内存/128 * 100 + 200。
code_type	String	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	String	<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。

名称	参数类型	说明
code_filename	String	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	Int64	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。
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_code	String	函数代码，请参考 表5-4 。
depend_list	[]String	依赖包列表。
strategy_config	String	函数策略配置，请参考 表5-34 。
extend_config	String	函数扩展配置。
dependencies	[]*String	依赖代码包。请参考 表4-5
initializer_handler	String	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	Int	初始化超时时间，超时函数将被强行停止，范围1 ~ 300秒。

名称	参数类型	说明
func_vpc	*String	VPC配置, 请参考表5-6。
mount_config	*String	磁盘挂载配置, 请参考表5-7。

表 5-34 strategy_config 参数说明

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

示例

请求示例

```
POST /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/versions HTTP/1.1
{
  "digest": "",
  "version": "1.0.0",
  "description": "test publish version"
}
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200
{
  "func_urn": "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",
  "func_name": "test",
  "user_domain": "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":
  "decfce6939297b0b5ec6d1a23bf9c725870f5e69fc338a89a6a4029264688dc26338f56d08b6535de47f15ad538e22ca66613b9a46f807d50b687bb53fded1c6",
  "version": "latest",
  "image_name": "latest-5qe8e",
  "xrole": "cff",
  "description": "111",
  "last_modified": "2018-03-28T11:30:32+08:00",
  "func_code": {
  },
  "strategy_config": {
```

```
"concurrency": -1
},
"initializer_handler": "index.initializer",
"initializer_timeout": 3 }
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1051",
  "error_msg": "Not found the function"
}
```

状态码

状态码请参见[状态码](#)。

5.9 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。

URI

GET /v2/{project_id}/fgs/functions/{function_urn}/versions?
marker={marker}&maxitems={maxitems} HTTP/1.1

URI参数说明如[表5-35](#)所示。

表 5-35 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参考 FunctionGraph函数模型 。
marker	String	必选	上一次查询到的最后的记录位置。
maxitems	String	必选	每次查询获取的最大函数记录数量。

请求消息

无

响应消息

响应参数说明如[表5-36](#)所示。

表 5-36 响应参数说明

名称	参数类型	说明
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。
runtime	String	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、Python 2.7、Python 3.6、Java 8、Go 1.8、C#.NET Core 2.0、C#.NET Core 2.1、C#.NET Core 3.1和PHP 7.3。
timeout	Int	函数执行超时时间, 超时函数将被强行停止, 范围3 ~ 900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。
memory_size	Int	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	Int	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), 函数占用的CPU为基础CPU: 200 millicores, 再加上内存按比例占用的CPU, 计算方法: 内存/128 * 100 + 200。
code_type	String	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	String	<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。

名称	参数类型	说明
code_filename	String	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	Int64	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。
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_code	String	函数代码，请参考 表5-4 。
depend_list	[]String	依赖包列表。
strategy_config	String	函数策略配置，请参考 表5-37 。
extend_config	String	函数扩展配置。
dependencies	[]*String	依赖代码包。请参考 表4-5
initializer_handler	String	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	Int	初始化超时时间，超时函数将被强行停止，范围1 ~ 300秒。

名称	参数类型	说明
func_vpc	*String	VPC配置, 请参考表5-6。
mount_config	*String	磁盘挂载配置, 请参考表5-7。

表 5-37 strategy_config 参数说明

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

示例

请求示例

```
GET /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/versions?marker=1&maxtems=10 HTTP/1.1
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200
{
  "versions": [
    {
      "func_urn": "urn:fss:xxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test",
      "func_name": "test",
      "user_domain": "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_filename": "index.js",
      "code_size": 272,
      "digest": "decbce6939297b0b5ec6d1a23bf9c725870f5e69fc338a89a6a4029264688dc26338f56d08b6535de47f15ad538e22ca66613b9a46f807d50b687bb53fded1c6",
      "version": "latest",
      "image_name": "latest-5qe8e",
      "xrole": "cff",
      "description": "111",
      "last_modified": "2018-03-28T11:30:32+08:00",
      "func_code": {
      },
      "strategy_config": {
        "concurrency": -1
      }
    }
  ]
}
```

```

    },
    "initializer_handler": "index.initializer",
    "initializer_timeout": 3
  }
],
"next_marker": 1
}

```

失败时的返回格式:

```

HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1051",
  "error_msg": "Not found the function"
}

```

状态码

状态码请参见[状态码](#)。

5.10 创建函数版本别名

功能介绍

创建函数灰度版本别名。

URI

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

URI参数说明如[表5-38](#)所示。

表 5-38 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。

请求消息

请求参数说明如[表5-39](#)所示。

表 5-39 请求参数说明

参数	类型	必选	说明
name	String	必选	函数别名名称。
version	String	必选	别名指向的具体版本号。

参数	类型	必选	说明
description	String	可选	别名描述信息
additional_version_weights	String	可选	json, 灰度版本号为key, 权重为value。

响应消息

响应参数说明如表5-40所示。

表 5-40 响应参数说明

名称	参数类型	说明
name	String	函数别名名称。
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	String	json, 灰度版本号为key, 权重为value。

示例

请求示例

```
POST /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/aliases HTTP/1.1
{
  "name": "dev",
  "version": "latest",
  "additional_version_weights": {"1.0": 10 }
}
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200
{
  "name": "dev",
  "version": "latest",
  "description": "",
  "last_modified": "2017-06-26 03:21:10",
  "additional_version_weights": {"1.0": 10 },
  "alias_urn": "urn:fss:xxxxxxxx: 7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:dev"
}
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1051",
  "error_msg": "Not found the function"
}
```

状态码

状态码请参见[状态码](#)。

5.11 修改函数版本别名信息

功能介绍

修改函数版本别名信息。

URI

PUT /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}

URI参数说明如[表5-41](#)所示。

表 5-41 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。
alias_name	String	必选	函数别名。

请求消息

请求参数说明如[表5-42](#)所示。

表 5-42 请求参数说明

参数	类型	必选	说明
version	String	必选	别名指向的新的版本号。
description	String	可选	函数别名描述。
additional_version_weights	String	可选	json，灰度版本号为key，权重为value。

响应消息

响应参数说明如[表5-43](#)所示。

表 5-43 响应参数说明

名称	参数类型	说明
name	String	待修改的别名名称。
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	String	json，灰度版本号为key，权重为value。

示例

请求示例

```
PUT /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxxxx:
7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/aliases/dev HTTP/1.1
{
  "version": "v20170725-152305",
  "description": "this is my version alias",
  "additional_version_weights": {"1.0": 10 }
}
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200
{
  "name": "dev",
  "version": "latest",
  "description": "",
  "last_modified": "2017-06-26 03:21:10",
  "additional_version_weights": {"1.0": 10 },
  "alias_urn": "urn:fss:xxxxxxxx: 7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:
dev"
}
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1051",
  "error_msg": "Not found the function"
}
```

状态码

状态码请参见[状态码](#)。

5.12 删除函数版本别名

功能介绍

删除函数版本别名。

说明

版本号被灰度版本使用时是不能删除的，如需删除灰度版本请先删除别名。

URI

DELETE /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}

URI参数说明如表5-44所示。

表 5-44 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。
alias_name	String	必选	待删除的别名名称。

请求消息

无

响应消息

无

示例

请求示例

```
DELETE /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fs:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/aliases/dev HTTP/1.1
```

响应示例

成功时的返回格式:

```
HTTP/1.1 204
```

失败时的返回格式:

```
HTTP / 1.1 404 Not Found  
{
```

```
"error_code": "FSS.1053",  
"error_msg": "Not found the function alias"  
}
```

状态码

状态码请参见[状态码](#)。

5.13 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。

URI

GET /v2/{project_id}/fgs/functions/{function_urn}/aliases/{alias_name}

URI参数说明如[表5-45](#)所示。

表 5-45 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。
alias_name	String	必选	别名名称。

请求消息

无

响应消息

响应参数说明如[表5-46](#)所示。

表 5-46 响应参数说明

名称	参数类型	说明
name	String	待获取的别名名称。
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。

名称	参数类型	说明
additional_version_weights	String	json, 灰度版本号为key, 权重为value。
alias_urn	String	版本别名唯一标识。

示例

请求示例

```
GET /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest / aliases/dev HTTP/1.1
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200
{
  "name":"dev",
  "version":"latest",
  "description":"my dev version",
  "last_modified":"2017-06-26 03:21:10",
  "additional_version_weights":{"1.0":10 },
  "alias_urn":"urn:fss:xxxxxxx: 7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:!dev"
}
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1053",
  "error_msg": "Not found the function alias"
}
```

状态码

状态码请参见[状态码](#)。

5.14 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。

URI

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

URI参数说明如[表5-47](#)所示。

表 5-47 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。

请求消息

无

响应消息

响应参数说明如[表5-48](#)所示。

表 5-48 响应参数说明

名称	参数类型	说明
name	String	待获取的别名名称。
version	String	别名对应的版本名称。
description	String	别名描述信息。
additional_version_weights	String	json，灰度版本号为key，权重为value。
last_modified	String	别名最后修改时间。
alias_urn	String	版本别名唯一标识。

示例

请求示例

```
GET /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/aliases HTTP/1.1
```

响应示例

成功时的返回格式：

```
HTTP/1.1 200 OK
[
  {
    "alias_urn": "urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:!testqq",
    "additional_version_weights":{"1.0":10 },
    "last_modified": "2018-03-21T10:06:30+08:00",
    "name": "testqq",
    "version": "latest"
  }
]
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1051",
  "error_msg": "Not found the function"
}
```

状态码

状态码请参见[状态码](#)。

5.15 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。

URI

GET /v2/{project_id}/fgs/triggers/{function_urn}

请求消息

请求参数说明如[表5-49](#)所示。

表 5-49 请求参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。

响应消息

响应参数说明如[表5-50](#)所示。

表 5-50 响应参数说明

名称	参数类型	说明
trigger_id	String	触发器ID。
trigger_status	String	触发器状态。
trigger_type_code	String	触发器类型代码。

名称	参数类型	说明
event_data	String	触发器相关定义数据，以JSON结构形式出现。 说明 所有触发器的数据说明，请参考 触发器实例数据 。
last_updated_time	String	最后更新时间。
created_time	String	创建时间。

示例

请求示例

```
GET /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/triggers/urn:fss:xxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest
HTTP/1.1
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200 OK
[
  {
    "trigger_id": "0586f1e2-8db2-4d2a-97bd-989f67d9fd8b",
    "trigger_type_code": "TIMER",
    "trigger_status": "ACTIVE",
    "event_data": {
      "name": "Timer-tg0q",
      "schedule": "3m",
      "schedule_type": "Rate"
    }
  }
  "last_updated_time": "2020-04-23T15:02:17+08:00",
  "created_time": "2020-04-23T15:02:17+08:00"
}
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{
  "error_code": "FSS.1051",
  "error_msg": "Error getting associated function"
}
```

状态码

状态码请参见[状态码](#)。

5.16 获取指定触发器的信息

功能介绍

获取特定触发器的信息。

URI

GET /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}

URI 参数说明如表 5-51 所示。

表 5-51 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	租户 Project ID。
function_urn	String	必选	函数的 URN，请参见 FunctionGraph 函数模型 。
trigger_type_code	String	必选	触发器类型代码，取值：SMN、APIG、OBS、TIMER、CTS、kafka。
trigger_id	String	必选	触发器编码。

请求消息

无

响应消息

响应参数说明如表 5-52 所示。

表 5-52 响应参数说明

名称	参数类型	说明
trigger_id	String	触发器 ID。
trigger_type_code	String	触发器类型代码。
trigger_status	String	触发器状态。
event_data	String	触发器相关定义数据，以 JSON 结构形式出现。 说明 所有触发器的数据说明请参考 触发器实例数据 。
last_updated_time	String	最后更新时间。
created_time	String	创建时间。

示例

请求示例

```
GET https://{functiongraph_endpoint}/v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/triggers/
urn:fs:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/TIMER/
9a14fae1-78cf-4185-ac7a-429eb6dc41fb HTTP/1.1
```

响应示例

成功时的返回格式:

```
{
  "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": "2019-10-29T17:15:53+08:00",
  "created_time": "2019-10-29T17:15:53+08:00"
}
```

失败时的返回格式:

```
{
  "error_code": "FS.0019",
  "error_msg": "Not found the function"
}
```

状态码

状态码请参见[状态码](#)。

5.17 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）。

URI

DELETE /v2/{project_id}/fgs/triggers/{function_urn}

URI参数说明如[表5-53](#)所示。

表 5-53 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	租户Project ID。
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。

请求消息

无

响应消息

无

示例

请求示例

```
DELETE /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/triggers/urn:fss:xxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest HTTP/1.1
```

响应示例

成功时的返回格式:

```
HTTP/1.1 204
```

失败时的返回格式:

```
HTTP/1.1 404  
{"error_code": "FS.0019", "error_msg": "Not found the function"}
```

状态码

状态码请参见[状态码](#)。

5.18 创建触发器

功能介绍

创建触发器。

URI

```
POST /v2/{project_id}/fgs/triggers/{function_urn}
```

URI参数说明如[表5-54](#)所示。

表 5-54 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	Project ID。
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。

请求消息

请求参数说明如[表5-55](#)所示。

表 5-55 请求参数说明

参数	类型	必选	说明
trigger_type_code	String	必选	触发器类型。
event_type_code	String	必选	事件类型。
trigger_status	String	必选	触发器状态，取值为ACTIVE或者DISABLED。
event_data	String	必选	事件消息。

响应消息

响应参数说明如[表5-56](#)所示。

表 5-56 响应参数说明

名称	参数类型	说明
trigger_id	String	触发器ID。
trigger_type_code	String	触发器类型代码。
event_type_code	String	事件类型代码。
trigger_status	String	触发器状态：ACTIVE / DISABLED。
event_data	String	触发器相关定义数据，以JSON结构形式出现。 说明 所有触发器的数据说明，请参考 触发器实例数据 。
last_updated_time	String	最后更新时间。
created_time	String	创建时间。

示例

请求示例

```
POST /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/triggers/urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest
{
  "trigger_type_code": "TIMER",
  "event_type_code": "MessageCreated",
  "trigger_status": "ACTIVE",
  "event_data": {
```

```
"name": "Timer-tps7",  
"schedule_type": "Rate",  
"schedule": "3m",  
"user_event": ""  
}  
}
```

响应示例

成功时的返回格式:

```
HTTP/1.1 201 Created  
{  
  "trigger_id": "be1cb36a-5efd-40ed-8376-7525bfcbe848",  
  "trigger_type_code": "TIMER",  
  "trigger_status": "ACTIVE",  
  "event_data": {  
    "name": "Timer-tps7",  
    "schedule": "3m",  
    "schedule_type": "Rate"  
  },  
  "last_updated_time": "2020-04-23T15:07:51+08:00",  
  "created_time": "2020-04-23T15:07:51+08:00"  
}
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found  
{  
  "error_code": "FSS.1051",  
  "error_msg": "Error getting associated function"  
}
```

状态码

状态码请参见[状态码](#)。

5.19 删除触发器

功能介绍

删除触发器。

URI

DELETE /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/
{trigger_id}

URI参数说明如[表5-57](#)所示。

表 5-57 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	租户Project ID。

参数	类型	必选	说明
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。
trigger_type_code	String	必选	触发器类型代码，取值：SMN、APIG、OBS、TIMER、CTS、kafka。
trigger_id	String	必选	触发器编码。

请求消息

无

响应消息

无

示例

请求示例

```
DELETE /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/triggers/urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/CTS/f4748d95-7ce7-4f9e-9434-67316a828d94 HTTP/1.1
```

响应示例

成功时的返回格式：

```
HTTP/1.1 204 No Content
```

失败时的返回格式：

```
HTTP/1.1 400 Bad Request
{
  "error_code": "FSS.0400",
  "error_msg": "Invalid trigger data"
}
```

状态码

状态码请参见[状态码](#)。

6 函数数据域接口

6.1 同步执行函数

功能介绍

同步执行函数。

说明

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。操作示例可以参考用户指南中的[测试管理](#)部分文档。

URI

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

URI参数说明如[表6-1](#)所示。

表 6-1 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	租户Project ID。
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。

Header: X-Cff-Log-Type, 取值为: tail (返回函数执行后的4K日志), 或者为空 (不返回日志)。

请求消息

函数的event, json格式。

参数	类型	必选	说明
{自定义key}	Map<String,String>	否	执行函数请求体，为json格式

响应消息

响应参数说明如表6-2所示。

表 6-2 响应参数说明

参数	类型	说明
X-Cff-Function-Log	String	base64编码后的函数执行日志。
X-Cff-Invoke-Summary	JSON	执行摘要。
Body	JSON	函数执行结果。

X-Cff-Invoke-Summary示例如下：

```
{"duration":1.913,"billingDuration":100,"memorySize":128,"memoryUsed":41.51171875}
```

表 6-3 X-Cff-Invoke-Summary 参数说明

参数	说明
duration	函数执行时长，单位毫秒(ms)。
billingDuration	收费时长，单位毫秒(ms)。
memorySize	配置内存，单位兆(MB)。
memoryUsed	实际使用内存，单位兆(MB)。

示例

请求示例

```
POST /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/invocations HTTP/1.1
{
  "message":"Hello World"
}
```

响应示例

成功时的返回格式:

```
HTTP/1.1 200 OK
{"message": "hello world from FunctionStage"}
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found
{"error_code": "FSS.0404", "error_msg": "Not found the specified resource"}
```

状态码

状态码请参见[状态码](#)。

6.2 异步执行函数

功能介绍

异步执行函数。

URI

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

URI参数说明如[表6-4](#)所示。

表 6-4 URI 参数说明

参数	类型	必选	说明
project_id	String	必选	租户Project ID。
function_urn	String	必选	函数的URN，请参见 FunctionGraph函数模型 。

请求消息

函数的event，json格式。

参数	类型	必选	说明
{自定义key}	Map<String,String>	否	执行函数请求体，为json格式

响应消息

响应参数说明如[表6-5](#)所示。

表 6-5 响应参数说明

参数	类型	说明
Body	JSON	本次执行的请求ID。

示例

请求示例

```
POST /v2/7aad83af3e8d42e99ac194e8419e2c9b/fgs/functions/urn:fss:xxxxxxxxx:7aad83af3e8d42e99ac194e8419e2c9b:function:default:test:latest/invocations-async HTTP/1.1
```

响应示例

成功时的返回格式:

```
HTTP/1.1 202 Accepted  
{ "request_id": "e834cb5b-1b2b-4c6b-b41c-8bd10fd41826" }
```

失败时的返回格式:

```
HTTP/1.1 404 Not Found  
{ "error_code": "FSS.0404", "error_msg": "function 'test' not exist" }
```

函数被禁用时返回格式:

```
HTTP / 1.1 429 Disabled  
{  
  "error_code": "FSS.0429",  
  "error_msg": "Function Disabled"  
}
```

状态码

状态码请参见[状态码](#)。

7 权限及授权项说明

表 7-1 FunctionGraph 授权项明细

权限	对应API接口	授权项
获取函数列表	GET /v2/{project_id}/fgs/functions	FunctionGraph:function:li st
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:g etCode
创建函数	POST /v2/{project_id}/fgs/ functions	FunctionGraph:function:c reate
删除函数/版本	DELETE /v2/{project_id}/fgs/ functions/{function_urn}	FunctionGraph:function:d elete
修改函数代码	PUT /v2/{project_id}/fgs/ functions/{function_urn}/code	FunctionGraph:function:u pdateCode
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:c reateVersion
获取指定函数的 版本列表	GET /v2/{project_id}/fgs/ functions/{function_urn}/versions	FunctionGraph:function:li stVersion
创建函数版本别 名	POST /v2/{project_id}/fgs/ functions/{function_urn}/aliases	FunctionGraph:function:c reateAlias
修改函数版本别 名信息	PUT /v2/{project_id}/fgs/ functions/{function_urn}/aliases/ {alias_name}	FunctionGraph:function:u pdateAlias
删除函数版本别 名	DELETE /v2/{project_id}/fgs/ functions/{function_urn}/aliases/ {alias_name}	FunctionGraph:function:d eleteAlias

权限	对应API接口	授权项
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
创建触发器	POST /v2/{project_id}/fgs/triggers/{function_urn}	FunctionGraph:trigger:create
删除触发器	DELETE /v2/{project_id}/fgs/triggers/{function_urn}/{trigger_type_code}/{trigger_id}	FunctionGraph:trigger:delete
同步执行函数	POST /v2/{project_id}/fgs/functions/{function_urn}/invocations	FunctionGraph:function:invoke
异步执行函数	POST /v2/{project_id}/fgs/functions/{function_urn}/invocations-async	FunctionGraph:function:invokeAsync

8 附录

8.1 状态码

状态码如表8-1所示

表 8-1 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。

8.2 错误码

表 8-2 FunctionGraph 函数错误码

状态码	错误码	错误信息	错误说明
400	FSS.0400	Invalid parameter.	无效参数
401	FSS.0401	Unauthorized access.	未授权的访问
403	FSS.0403	Forbidden	禁止操作
404	FSS.0404	The specified resource was not found.	没有找到指定的资源
406	FSS.0406	Not acceptable.	请求格式无效，例如：请求body的JSON格式与要求的不一样
408	FSS.0408	Request timeout.	请求超时
409	FSS.0409	The specified resource already exists.	指定的资源已经存在
410	FSS.0410	The specified resource does not exist.	指定的资源不存在
413	FSS.0413	The request body is too large.	请求体大小超过限制
424	FSS.0424	Invalid dependency.	无效的依赖
426	FSS.0426	An upgrade is required.	不支持该操作，需要升级
428	FSS.0428	The preconditions have not been met.	先决条件不满足，例如删除某个资源必须先删除其他相关资源
429	FSS.0429	Too many requests.	太多的请求
500	FSS.0500	Internal server error.	服务临时性异常，服务内部调用出错，稍后重试

状态码	错误码	错误信息	错误说明
502	FSS. 0502	Bad gateway.	网关错误
503	FSS. 0503	Service unavailable.	服务不可用
504	FSS. 0504	Gateway timeout.	网关超时
400	FSS. 1001	Invalid query parameter.	无效的查询参数
400	FSS. 1002	Invalid function name.	无效的函数名称
400	FSS. 1003	Invalid function handler.	无效的函数Handler
400	FSS. 1004	Invalid Package (function app).	无效的函数包
400	FSS. 1005	Invalid runtime.	无效的Runtime
400	FSS. 1006	Invalid function code entry mode.	无效的代码类型
400	FSS. 1007	Invalid function memory.	无效的内存
400	FSS. 1008	Invalid function timeout.	无效的超时时间
400	FSS. 1009	Invalid function UserData.	无效的环境变量
400	FSS. 1010	Invalid URL.	无效的URL
400	FSS. 1011	Invalid function code.	无效的函数代码
400	FSS. 1012	The function code must be configured.	函数代码不能为空
400	FSS. 1013	Invalid file type.	无效的文件类型
400	FSS. 1014	Invalid function alias.	无效的函数版本别名
400	FSS. 1015	Invalid function version.	别名对应的版本无效

状态码	错误码	错误信息	错误说明
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，当前函数代码已经被发布过版本
400	FSS. 1017	The number of items in the UserData field exceeds the maximum allowed limit (20).	环境变量个数超过限制（20）
400	FSS. 1018	The total size of the UserData field exceeds the maximum allowed limit (2 KB).	环境变量总大小超过限制（2K）
400	FSS. 1019	The description exceeds the maximum allowed limit.	描述信息长度超过限制（512）
400	FSS. 1022	Only one YAML file is allowed.	只容许有一个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
400	FSS. 1024	Invalid dependency.	无效的依赖包
400	FSS. 1025	Invalid YAML file.	无效的YAML文件
400	FSS. 1026	Invalid Concurrency.	无效的并发策略
400	FSS. 1027	Invalid packageName (app name).	无效的包名或者应用名称
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
400	FSS. 1029	The default app cannot be deleted.	默认的应用不能删除
400	FSS. 1031	The dependency already exists.	依赖包已经存在
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
412	FSS. 1033	The dependency is currently in use.	依赖包正在使用，不能被删除
400	FSS. 1034	Invalid image URL.	无效的镜像地址

状态码	错误码	错误信息	错误说明
400	FSS.1035	The image does not exist.	该镜像不存在
400	FSS.1036	The VPC does not exist.	该VPC不存在
400	FSS.1037	No subnet matches the specified ID.	没有匹配的子网
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
400	FSS.1039	The mounting path is invalid.	无效的挂载路径
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
403	FSS.1041	The number of functions exceeds the maximum allowed limit.	租户的函数数量超过限制 (400)
403	FSS.1042	The total code size of functions exceeds the maximum allowed limit.	租户的函数的总大小超过限制 (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
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)
403	FSS.1045	The number of dependencies exceeds the maximum allowed limit.	依赖包数量超过了最大限制
403	FSS.1046	The dependency is inaccessible.	该依赖包不可用
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
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

状态码	错误码	错误信息	错误说明
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）
404	FSS.1050	The mounted resource cannot be found.	找不到挂载的资源类型
404	FSS.1051	The function does not exist.	找不到指定的函数
404	FSS.1052	The version does not exist.	找不到指定的函数版本
404	FSS.1053	The alias does not exist.	找不到指定的函数版本别名
404	FSS.1054	The function app does not exist in OBS.	从OBS找不到指定的函数代码包
404	FSS.1055	The app does not exist.	从OBS找不到指定的函数应用
404	FSS.1056	The dependency does not exist.	找不到指定的依赖包
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
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
404	FSS.1059	The function template does not exist.	函数模板不存在
404	FSS.1060	The event template cannot be found.	事件模板不存在
409	FSS.1061	The function already exists.	函数已经存在
409	FSS.1062	The version already exists.	函数版本已经存在
409	FSS.1063	The alias already exists.	函数版本别名已经存在
409	FSS.1064	The app already exists.	函数应用已经存在

状态码	错误码	错误信息	错误说明
409	FSS.1065	The dependency already exists.	依赖包已经存在
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
409	FSS.1067	The function template already exists.	函数模板已经存在
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
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)
404	FSS.1070	The event cannot be found.	事件不存在
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)
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)
403	FSS.1073	The function event already exists.	事件在函数中已经存在
400	FSS.1074	The event field is invalid.	无效的事件参数
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之间的非0整数
412	FSS.1090	The subnet is not in the \"ACTIVE\" state.	子网不是ACTIVE状态
400	FSS.1091	The additional version is invalid.	无效的灰度版本
400	FSS.1092	The weight of the additional version is invalid.	无效的灰度版本权重值
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

状态码	错误码	错误信息	错误说明
403	FSS.1094	The version cannot be deleted because it has been 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
412	FSS.1095	The mounted resource is not ready.	挂载资源还没有准备好
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
400	FSS.1101	Invalid trigger type.	无效的trigger类型
400	FSS.1102	Invalid SMN trigger parameter.	无效的SMN触发器参数
400	FSS.1106	Invalid OBS trigger parameters.	无效的OBS触发器参数
400	FSS.1107	Invalid APIG trigger parameters.	无效的APIG触发器参数
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
400	FSS.1109	Invalid timer trigger parameters.	无效的定时任务触发器参数
404	FSS.1111	The Kafka resource cannot be found.	指定的kafka不存在
400	FSS.1112	The Kafka trigger parameters are invalid.	无效的kafka触发器参数
400	FSS.1113	The username and password must be specified because Kafka SASL_SSL is enabled.	kakfa开启了SASL_SSL协议，所以必须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kakfa实例所在的子网不一样
503	FSS.1115	The network is unreachable.	网络不可达

状态码	错误码	错误信息	错误说明
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
400	FSS.1117	Failed to query messages from the Kafka instance.	从kafka实例读取消息失败
401	FSS.1118	Access denied.	访问受限，该用户不在白名单中
403	FSS.1121	Forbidden	访问受限，请检查是否有配置对应委托
403	FSS.1122	Forbidden	访问受限，请检查是否有配置对应委托
403	FSS.1123	The number of pull triggers exceeds the maximum allowed limit.	拉模式的触发器数量达到最大值10
403	FSS.1124	The number of APIs exceeds the maximum allowed limit.	API数量超过了最大限制
403	FSS.1125	Forbidden	无权限访问
403	FSS.1126	You do not have permissions to call the API.	没有权限访问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
404	FSS.1131	The trigger does not exist.	没有找到指定的触发器
404	FSS.1132	The SMN trigger does not exist. View the SMN console.	没有找到指定的SMN触发器
404	FSS.1136	The OBS trigger does not exist.	没有找到指定的OBS触发器
404	FSS.1137	Invalid trigger type.	没有找到触发器类型
404	FSS.1138	The APIG trigger does not exist.	没有找到指定的APIG触发器
404	FSS.1140	The timer trigger does not exist.	没有找到指定的定时任务触发器

状态码	错误码	错误信息	错误说明
409	FSS.1141	The SMN trigger already exists.	指定的SMN触发器已经存在
409	FSS.1145	The OBS trigger already exists.	指定的OBS触发器已经存在
409	FSS.1146	The APIG trigger already exists.	指定的APIG触发器已经存在
409	FSS.1147	The request path already exists.	APIG触发器路径已经存在
409	FSS.1148	The timer trigger already exists.	指定的定时任务触发器已经存在
409	FSS.1150	The Kafka trigger already exists.	指定的kafka触发器已经存在
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
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
412	FSS.1153	The triggering conditions have not been met.	触发器前置条件不满足
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
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
503	FSS.1169	The network is unreachable.	网络不可达
404	FSS.1171	The SMN topic does not exist. Create it on the SMN console.	该SMN的topic不存在
400	FSS.1172	The database or collection does not exist.	找不到指定的数据库实例
404	FSS.1174	The Kafka trigger does not exist.	找不到指定的kafka触发器
413	FSS.1201	The request body is too large.	请求体大小超过限制

状态码	错误码	错误信息	错误说明
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
500	FSS.1302	Failed to save the data.	触发器存储失败
400	FSS.1303	Access denied due to insufficient permissions.	权限校验失败，访问被禁止
400	FSS.1306	The number of triggers exceeds the maximum allowed limit.	触发器达到上限
400	FSS.1307	The trigger name already exists.	触发器名称已存在
400	FSS.1308	The operation resource does not exist.	操作资源不存在
400	FSS.1309	Invalid function URN.	functionUrn验证错误
400	FSS.1310	Unauthorized user.	获取组用户Token验证失败
400	FSS.1312	The notification name must be specified.	关键通知名称为空
400	FSS.1313	The number of operation resources has reached the maximum allowed limit.	操作资源超过100条上限
400	FSS.1314	The operation resource must be specified.	操作资源为空
400	FSS.1316	The resource operation has already been selected.	相同操作资源
400	FSS.1317	The trigger name is too long.	触发器名字太长了
400	FSS.1318	Invalid trigger operation.	无效的触发器操作
502	FSS.1319	Invalid trigger name.	无效触发器名称
503	FSS.1401	Failed to obtain the image information.	获取镜像信息失败

状态码	错误码	错误信息	错误说明
503	FSS. 1402	Failed to pull the image to create a container.	加载镜像创建容器失败
503	FSS. 1403	Failed to pull the image to delete a container.	加载镜像删除容器失败
400	FSS. 1404	Invalid function initializer.	非法的函数初始化入口
400	FSS. 1405	Invalid initialization timeout.	非法的函数初始化超时时间

8.3 获取项目 ID

从控制台获取项目 ID

在调用接口的时候，部分URL中需要填入项目ID，获取项目ID的步骤如下：

1. 注册并登录管理控制台。
2. 单击用户名，在下拉列表中单击“我的凭证”。
在“我的凭证”页面，查看项目ID。

调用 API 获取项目 ID

项目ID还用通过调用[查询指定条件下的项目信息](#)API获取。

获取项目ID的接口为“GET https://{Endpoint}/v3/projects”，其中{Endpoint}为IAM的终端节点，可以从[地区和终端节点](#)获取。接口的认证鉴权请参见[认证鉴权](#)。

响应示例如下，其中projects下的“id”即为项目ID。

```
{
  "projects": [
    {
      "domain_id": "65382450e8f64ac0870cd180d14e684b",
      "is_domain": false,
      "parent_id": "65382450e8f64ac0870cd180d14e684b",
      "name": "xxx",
      "description": "",
      "links": {
        "next": null,
        "previous": null,
        "self": "https://www.example.com/v3/projects/a4a5d4098fb4474fa22cd05f897d6b99"
      },
      "id": "a4a5d4098fb4474fa22cd05f897d6b99",
      "enabled": true
    }
  ],
  "links": {
    "next": null,
    "previous": null,
    "self": "https://www.example.com/v3/projects"
  }
}
```

8.4 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 8-3 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
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

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
max Duration	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
min Duration	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 8-4 维度

Key	Value
package-functionname	应用名称-函数名称

9 修订记录

表 9-1 修订记录

日期	修订记录
2021-01-19	修改以下内容： <ul style="list-style-type: none">• 刷新 FunctionGraph 函数模型 章节。• 刷新 获取函数列表 章节。
2020-10-30	第一次发布。