



**ServiceStage**

## **API 参考**

发布日期 2023-06-25

---

# 目录

---

<b>1 使用前必读</b>	<b>1</b>
1.1 概述	1
1.2 调用说明	1
1.3 终端节点	1
1.4 基本概念	1
<b>2 API 概览</b>	<b>3</b>
<b>3 如何调用 API</b>	<b>9</b>
3.1 构造请求	9
3.2 认证鉴权	11
3.3 返回结果	12
<b>4 应用管理 API</b>	<b>13</b>
4.1 Meta	13
4.1.1 获取所有支持的应用组件运行时类型	13
4.1.2 获取所有支持的应用资源规格	15
4.2 环境	16
4.2.1 创建环境	17
4.2.2 获取所有环境	20
4.2.3 修改环境信息	23
4.2.4 根据环境 ID 删除环境	26
4.2.5 根据环境 ID 获取环境详细信息	27
4.2.6 修改环境资源	29
4.3 应用	33
4.3.1 创建应用	33
4.3.2 获取所有应用	35
4.3.3 修改应用信息	37
4.3.4 根据应用 ID 删除应用	39
4.3.5 根据应用 ID 获取应用详细信息	40
4.3.6 添加或修改应用配置信息	42
4.3.7 删除应用配置	44
4.3.8 获取应用配置	46
4.4 组件	48
4.4.1 应用中创建组件	48

4.4.2 获取应用所有组件.....	55
4.4.3 根据组件 ID 修改组件信息.....	59
4.4.4 根据应用组件 ID 删除应用组件.....	65
4.4.5 根据组件 ID 获取应用组件信息.....	66
4.5 实例.....	70
4.5.1 创建组件实例.....	70
4.5.2 获取应用组件实例.....	78
4.5.3 对组件实例的操作.....	82
4.5.4 修改应用组件实例.....	84
4.5.5 删除应用组件实例.....	88
4.5.6 根据实例 ID 获取实例详细信息.....	89
4.5.7 获取组件实例快照.....	94
4.6 Job.....	96
4.6.1 获取部署任务详细信息.....	96
<b>5 访问 Git 仓库 API.....</b>	<b>101</b>
5.1 获取仓库授权列表.....	101
5.2 获取授权重定向 URL.....	103
5.3 创建 OAuth 授权.....	104
5.4 创建私人令牌授权.....	107
5.5 创建口令授权.....	110
5.6 删除仓库授权.....	112
5.7 获取仓库 namespaces.....	113
5.8 通过 clone url 获取仓库信息.....	115
5.9 获取命名空间下所有项目.....	116
5.10 创建软件仓库项目.....	118
5.11 获取项目分支.....	119
5.12 获取项目 tags.....	121
5.13 创建项目 tags.....	122
5.14 删除项目 tags.....	124
5.15 获取项目 commits.....	125
5.16 获取项目 hooks.....	127
5.17 创建项目 hook.....	128
5.18 删除项目 hook.....	130
5.19 获取仓库文件目录.....	131
5.20 获取仓库文件内容.....	133
5.21 创建仓库文件.....	135
5.22 修改仓库文件内容.....	136
5.23 删除仓库文件.....	138
<b>6 CSE API.....</b>	<b>141</b>
6.1 调用说明.....	141
6.2 查询指定微服务静态信息.....	141
6.3 查询所有微服务静态信息.....	142

6.4 创建微服务静态信息.....	144
6.5 修改微服务静态信息.....	146
6.6 查询微服务访问契约.....	148
6.7 修改微服务访问契约.....	149
6.8 创建服务间依赖关系.....	150
6.9 查询服务 providers.....	152
6.10 查询微服务唯一标识信息.....	153
6.11 注册微服务实例.....	155
6.12 查询微服务实例 ( 根据 serviceld ) .....	156
6.13 注销微服务实例.....	158
6.14 查询微服务实例详细信息.....	159
6.15 修改微服务实例扩展信息.....	161
6.16 修改微服务实例状态.....	162
6.17 发送心跳信息.....	163
6.18 按条件查询微服务实例.....	165
6.19 查询配置.....	167
6.20 删除微服务信息.....	169
6.21 批量删除微服务静态信息.....	170
6.22 批量查询微服务实例.....	172
6.23 查询微服务的所有契约.....	176
<b>7 数据结构.....</b>	<b>179</b>
7.1 参数说明.....	179
7.1.1 HealthCheck.....	179
7.1.2 MicroServiceInstance.....	180
7.1.3 HeartbeatSetElement.....	181
7.1.4 InstanceHbRst.....	181
7.1.5 DelServicesResponse.....	181
7.1.6 MetricData.....	181
7.1.7 FunctionMetricInfo.....	182
7.1.8 InstanceMetricInfo.....	183
7.2 公共请求参数.....	183
7.2.1 MicroService.....	183
7.2.2 Properties.....	185
7.2.3 DependencyMicroService.....	185
7.2.4 Rule.....	186
7.2.5 AddOrUpdateRule.....	187
7.2.6 DataCenterInfo.....	187
7.2.7 Schema.....	188
7.2.8 environment_setting.....	188
7.2.9 phases.....	188
7.3 公共响应参数.....	189
7.3.1 WatchMicroServiceKey.....	189

---

7.3.2 SuccessdResponse.....	189
7.3.3 ServiceInfo.....	190
7.3.4 AggregateMetricInfo.....	190
7.3.5 ServiceDetail.....	191
7.3.6 TenantProject.....	192
7.3.7 Error.....	192
<b>8 权限策略说明.....</b>	<b>193</b>
8.1 策略及授权项说明.....	193
<b>9 附录.....</b>	<b>196</b>
9.1 状态码.....	196
9.2 ServiceStage 错误码.....	196
9.3 CSE 错误码.....	200
9.4 获取项目 ID.....	201
9.5 获取帐号 ID.....	202
9.6 获取服务注册发现地址.....	202

# 1 使用前必读

## 1.1 概述

应用管理与运维平台（ServiceStage）是面向企业的应用管理与运维平台，提供应用发布、部署、监控与运维等一站式解决方案。支持Java、Go、PHP、Node.js、Python、Docker、Tomcat等运行环境。支持Web应用，以及Apache ServiceComb、Spring Cloud、Dubbo、Mesher服务网格等微服务应用和通用应用，让企业应用上云更简单。

ServiceStage所提供的自研接口类型如下：

- 应用管理
- 访问Git仓库
- CSE（Cloud Service Engine）：微服务引擎

## 1.2 调用说明

ServiceStage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**系统管理员**中查询所有服务的终端节点。

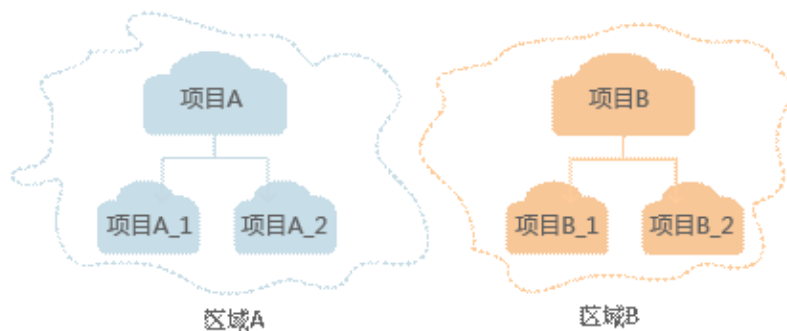
## 1.4 基本概念

- 帐号  
用户注册云平台时的帐号，帐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。
- 用户  
由帐号在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，您需要用到帐号、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  
云平台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，用户可以访问您帐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，并在子项目中创建资源，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

ServiceStage 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。

对应的接口参考请参考下表：

表 2-1 接口简介

类型	子类型	说明
应用管理接口	<a href="#">Meta接口</a>	应用模板、运行时、应用资源规格相关的接口。
	<a href="#">环境接口</a>	应用环境相关的接口。
	<a href="#">应用接口</a>	应用相关的接口。
	<a href="#">组件接口</a>	应用组件相关的接口。
	<a href="#">实例接口</a>	应用组件实例相关的接口。
	<a href="#">Job接口</a>	Job相关的接口。
访问Git仓库接口	<a href="#">访问Git仓库接口</a>	Git仓库授权、仓库命名空间、项目、分支、tags、commits、hooks、仓库文件目录、内容相关的接口。
CSE接口	<a href="#">CSE接口</a>	微服务引擎相关的接口。



## Meta 接口

表 2-2 Meta 接口介绍

API	说明
<a href="#">获取所有支持的应用组件运行时类型</a>	此API用来获取所有支持应用组件运行时类型。
<a href="#">获取所有支持的应用资源规格</a>	通过此API获取所用支持的应用资源规格。

## 环境接口

表 2-3 环境接口介绍

API	说明
<a href="#">创建环境</a>	此API用来创建环境。
<a href="#">获取所有环境</a>	此API用来获取所有已经创建环境。
<a href="#">修改环境信息</a>	此API通过环境ID修改环境信息。
<a href="#">根据环境ID删除环境</a>	此API通过环境ID删除环境。
<a href="#">根据环境ID获取环境详细信息</a>	此API通过环境ID获取环境详细信息。
<a href="#">修改环境资源</a>	此API用来修改环境资源。

## 应用接口

表 2-4 应用接口介绍

API	说明
<a href="#">创建应用</a>	此API用来创建应用。
<a href="#">获取所有应用</a>	通过此API可以获取所有已经创建的应用。
<a href="#">修改应用信息</a>	此API通过应用ID修改应用信息。
<a href="#">根据应用ID删除应用</a>	此API通过应用ID删除应用。
<a href="#">根据应用ID获取应用详细信息</a>	此API通过应用ID获取应用详细信息。
<a href="#">添加或修改应用配置信息</a>	通过此API添加或修改应用配置信息。
<a href="#">删除应用配置</a>	通过此API删除应用配置信息。
<a href="#">获取应用配置</a>	通过此API获取应用配置信息。

## 组件接口

表 2-5 组件接口介绍

API	说明
<a href="#">应用中创建组件</a>	此API用来在应用中创建组件。
<a href="#">获取应用所有组件</a>	通过此API获取应用下所有应用组件。
<a href="#">根据组件ID修改组件信息</a>	此API通过组件ID修改组件信息。
<a href="#">根据应用组件ID删除应用组件</a>	此API通过应用组件ID删除应用组件。
<a href="#">根据组件ID获取应用组件信息</a>	通过组件ID获取应用组件信息。

## 实例接口

表 2-6 实例接口介绍

API	说明
<a href="#">创建组件实例</a>	此API用来创建应用组件实例。
<a href="#">获取应用组件实例</a>	通过此API获取组件下的所有组件实例。
<a href="#">对组件实例的操作</a>	通过此API执行对组件实例的操作。
<a href="#">修改应用组件实例</a>	通过此API修改应用组件实例。
<a href="#">删除应用组件实例</a>	通过此API删除应用组件实例。
<a href="#">根据实例ID获取实例详细信息</a>	此API通过实例ID获取实例详细信息。
<a href="#">获取组件实例快照</a>	通过此API获取应用组件实例快照信息。

## Job 接口

表 2-7 Job 接口介绍

API	说明
<a href="#">获取部署任务详细信息</a>	通过此API获取部署任务详细信息。

## 访问 Git 仓库接口

表 2-8 访问 Git 仓库接口介绍

API	说明
<a href="#">获取仓库授权列表</a>	获取Git仓库所有授权信息。
<a href="#">获取授权重定向URL</a>	获取授权重定向URL。
<a href="#">创建OAuth授权</a>	创建OAuth授权。
<a href="#">创建私人令牌授权</a>	创建私有令牌授权。
<a href="#">创建口令授权</a>	创建Git仓库口令授权。
<a href="#">删除仓库授权</a>	通过名称删除仓库授权。
<a href="#">获取仓库namespaces</a>	获取仓库namespaces。
<a href="#">通过clone url 获取仓库信息</a>	通过clone url获取仓库信息。
<a href="#">获取命名空间下所有项目</a>	获取命名空间下所有项目。
<a href="#">创建软件仓库项目</a>	创建软件仓库项目。
<a href="#">获取项目分支</a>	获取项目分支。
<a href="#">获取项目tags</a>	获取项目tags。
<a href="#">创建项目tags</a>	创建项目tags。
<a href="#">删除项目tags</a>	删除项目tags。
<a href="#">获取项目commits</a>	获取项目最近10次commits。
<a href="#">获取项目hooks</a>	获取项目hooks。
<a href="#">创建项目hook</a>	创建项目hook。
<a href="#">删除项目hook</a>	删除项目hook。
<a href="#">获取仓库文件目录</a>	获取仓库文件目录。
<a href="#">获取仓库文件内容</a>	获取仓库文件内容。
<a href="#">创建仓库文件</a>	创建仓库文件。
<a href="#">修改仓库文件内容</a>	修改仓库文件内容。
<a href="#">删除仓库文件</a>	删除仓库文件内容。

## CSE 接口

表 2-9 CSE 接口介绍

API	说明
<a href="#">查询指定微服务静态信息</a>	根据servicelId查询微服务定义信息。
<a href="#">查询所有微服务静态信息</a>	根据条件组合，查询满足所有条件的微服务定义信息。
<a href="#">创建微服务静态信息</a>	在注册微服务实例前需要创建服务静态信息，之后注册的微服务实例根据servicelId这个字段与静态信息关联，一个服务对应多个实例。 servicelId可以定制。如果定制了，系统将使用定制的服务icelId；如果没定制，系统会随机生成servicelId。
<a href="#">修改微服务静态信息</a>	创建微服务静态信息后可对服务部分字段进行更新，每次更新都需要传入完整的服务静态信息json，也就是说，即便不更新部分的字段也要作为json的属性传过去。
<a href="#">查询微服务访问契约</a>	根据指定的servicelId和schemalId查询微服务的schema信息。
<a href="#">修改微服务访问契约</a>	根据schemalId更新微服务的访问契约内容。
<a href="#">创建服务间依赖关系</a>	创建服务间的依赖关系，consumer的version必须是确定的版本，consumer的serviceName不能为*，consumer必须是已存在的服务。provider可以是还未创建的。
<a href="#">查询服务providers</a>	根据consumerId获取该服务的所有providers。
<a href="#">查询微服务唯一标识信息</a>	可通过指定条件，查询微服务servicelId或schema的唯一标识信息。
<a href="#">注册微服务实例</a>	创建微服务后就可以注册该微服务的实例了。注册微服务实例时，需提供该微服务实例相关的信息。 instanceId可定制，如果定制了，再次注册就直接将全内容覆盖。如果没定制，逻辑如下：系统自动生成id，如果endpoints内容重复，则使用原来的id。
<a href="#">查询微服务实例（根据servicelId）</a>	实例注册后可以根据service_id发现该微服务的所有实例。
<a href="#">注销微服务实例</a>	实例注册后可以根据instance_id进行实例注销。
<a href="#">查询微服务实例详细信息</a>	实例注册后可以根据serviceid和instanceId获取该实例的详细信息。
<a href="#">修改微服务实例扩展信息</a>	实例注册后可以根据instance_id进行添加/更新一个微服务实例扩展信息。
<a href="#">修改微服务实例状态</a>	实例注册后可以根据instance_id进行更新一个微服务实例状态。

API	说明
<a href="#">发送心跳信息</a>	服务提供端需要向服务中心发送心跳信息，以保证服务中心知道服务实例是否健康。
<a href="#">按条件查询微服务实例</a>	实例注册后可以根据微服务版本规则或字段条件发现该微服务的实例。
<a href="#">查询配置</a>	从配置管理中心查询配置信息。
<a href="#">删除微服务信息</a>	删除一个微服务定义及其相关信息，同时注销其所有实例信息。
<a href="#">批量删除微服务静态信息</a>	批量删除指定的微服务定义及其相关信息，同时注销其所有实例信息。
<a href="#">批量查询微服务实例</a>	实例注册后可以根据微服务版本规则或字段条件批量发现多个微服务的实例。
<a href="#">查询微服务的所有契约</a>	查询一个微服务的所有契约信息，包括schemald和summary信息。

# 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。

- **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[系统管理员](#)处获取。
- **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，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

HTTPS方法，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.{Endpoint}/v3/auth/tokens
```

## 请求消息头

附加请求头字段，如指定的URI和HTTPS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。
- **X-Project-ID**: 子项目ID，可选，在多项目场景中使用。
- **X-Domain-ID**: 帐号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.{Endpoint}/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。

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***为用户所属的帐号名称，***\*\*\*\*\****为用户登录密码，***xxxxxxxxxxxxxxxxxxxx***为project的名称，可以从[系统管理员](#)处获取。

### 📖 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的作用域为某个帐号下所有资源或帐号的某个project下的资源，详细定义请参见[获取用户Token](#)。

```
POST https://iam.{Endpoint}/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命令行、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

调用接口有如下认证方式。

- Token认证：通过Token认证调用请求。

### 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。

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.{Endpoint}/v3/auth/projects  
Content-Type: application/json  
X-Auth-Token: ABCDEFJ....
```

## 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。

### 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": "xx-xxxxxx-1",  
            .....  
          }  
        ]  
      }  
    ]  
  }  
}
```

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": "xxxxxxxxx"  
}
```

其中，error\_code表示错误码，error\_msg表示错误描述信息。

# 4 应用管理 API

## 4.1 Meta

### 4.1.1 获取所有支持的应用组件运行时类型

#### 功能介绍

此API用来获取所有支持应用组件运行时类型。

#### URI

GET /v2/{project\_id}/cas/metadata/runtimes

表 4-1 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。

#### 请求消息

请求参数

无

#### 响应消息

响应参数

表 4-2 响应参数

参数	类型	说明
runtimes	Array of objects	运行时参数，请参考表 4-3。

表 4-3 runtimes 参数说明

参数	类型	说明
type_name	String	类型名称
display_name	String	显示名称
container_default_port	Integer	容器默认端口
type_desc	String	类型描述

## 示例

### 请求示例

无

### 响应示例

```
{
  "runtimes": [
    {
      "type_name": "Java8",
      "display_name": "Java 8",
      "container_default_port": 8080,
      "type_desc": "Java 8 runtime server"
    }
  ]
}
```

## 状态码

表 4-4 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.1.2 获取所有支持的应用资源规格

### 功能介绍

通过此API获取所用支持的应用资源规格。

### URI

GET /v2/{project\_id}/cas/metadata/flavors

表 4-5 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。

### 请求消息

无

### 响应消息

表 4-6 响应参数

参数	类型	说明
flavors	Array of objects	规格参数，请参考 <a href="#">表4-7</a> 。

表 4-7 flavors 参数说明

参数	类型	说明
flavor_id	String	规格ID
storage_size	String	存储大小
num_cpu	String	CPU限制
num_cpu_init	String	CPU初始
memory_size	String	内存限制
memory_size_init	String	内存初始
label	String	展示标签

参数	类型	说明
custom	boolean	是否是自定义资源规格

## 示例

### 请求示例

无

### 响应示例

```
{
  "flavors": [
    {
      "flavor_id": "MICRO-5G:0.5C:1G",
      "storage_size": "5G",
      "num_cpu": "500m",
      "num_cpu_init": "200m",
      "memory_size": "1Gi",
      "memory_size_init": "200Mi",
      "label": "Micro: 5G Storage, 0.5 CPU, 1G Memory",
      "custom": false
    }
  ]
}
```

## 状态码

表 4-8 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.2 环境

## 4.2.1 创建环境

### 功能介绍

环境是用于应用部署和运行的计算、存储、网络等基础设施的集合。ServiceStage把相同VPC下的CCE集群加上多个ELB、DCS实例组合为一个环境，如：开发环境，测试环境，预生产环境，生产环境。环境内网络互通，可以按环境维度来管理资源、部署服务，减少具体基础设施运维管理的复杂性。

此API用来创建环境。

### URI

POST /v2/{project\_id}/cas/environments

表 4-9 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。

### 请求消息

表 4-10 请求参数说明

参数	参数类型	必选	描述
name	String	是	环境名称。 由大小写英文字母、数字、中划线(-)、下划线(_)组成，并以大小写英文字母开头，大小写英文字母或者数字结尾，长度为2~64个字符。
alias	String	否	环境别名。 最大长度为64个字符。
description	String	否	环境描述。 最大长度为128个字符。
enterprise_project_id	String	否	企业项目ID。
charge_mode	String	否	固定值：provided。
vpc_id	String	是	虚拟私有云ID。
base_resources	Array of objects	是	基础资源，请参考 <a href="#">表4-11</a> 。

参数	参数类型	必选	描述
optional_resources	Array of objects	否	可选资源，请参考 <a href="#">表4-11</a> 。

表 4-11 resource 参数

参数	参数类型	必选	描述
id	String	是	资源ID。
name	String	否	资源名称。
type	String	是	基础资源：cce、ecs。 可选资源：dcs、elb等其他类型。

## 响应消息

表 4-12 响应参数说明

参数	参数类型	描述
id	String	环境ID
name	String	环境名称
alias	String	环境别名
description	String	环境描述
project_id	String	项目ID
enterprise_project_id	String	企业项目ID
charge_mode	String	收费模式
vpc_id	String	虚拟私有云ID
base_resources	Array of objects	基础资源，请参考 <a href="#">表4-13</a>
optional_resources	Array of objects	可选资源，请参考 <a href="#">表4-13</a>
creator	String	创建人
create_time	Integer	创建时间
update_time	Integer	修改时间

表 4-13 resource

参数	参数类型	描述
id	String	资源ID
name	String	资源名称
type	String	基础资源：cce、ecs 可选资源：dcs、elb等其他类型

## 示例

### 请求示例

```
{
  "name": "development-env",
  "description": "",
  "charge_mode": "provided",
  "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
  "base_resources": [
    {
      "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "type": "cce",
      "name": "cce-test"
    }
  ],
  "optional_resources": [
    {
      "id": "9963306a-791e-468c-9306-cf80a9d00298",
      "type": "elb",
      "name": "elb-2dd2"
    }
  ]
}
```

### 响应示例

```
{
  "id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
  "name": "development-env",
  "alias": null,
  "description": "",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "charge_mode": "provided",
  "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
  "base_resources": [
    {
      "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "type": "cce",
      "name": "cce-test"
    }
  ],
  "optional_resources": [
    {
      "id": "9963306a-791e-468c-9306-cf80a9d00298",
      "type": "elb",
      "name": "elb-2dd2"
    }
  ],
  "creator": "ss-test",
  "create_time": 1610418873730,
  "update_time": 1610418873730
}
```



## 状态码

表 4-14 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

### 4.2.2 获取所有环境

#### 功能介绍

此API用来获取所有已经创建的环境。

#### URI

GET /v2/{project\_id}/cas/environments

表 4-15 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。

表 4-16 查询参数

参数	参数类型	必选	描述
limit	Integer	否	指定个数，取值[0, 100]或者1000，指定的值不在该范围内的话相当于赋值为10。默认1000，用于不分页的场景。分页的场景取值请用[0, 100]。
offset	Integer	否	指定偏移量，表示查询第n条数据后的结果。

参数	参数类型	必选	描述
order_by	String	否	排序字段，默认按创建时间排序。 排序字段支持枚举值：create_time、name、update_time。传的值不在支持的枚举值范围内的话按默认排序字段排序。
order	String	否	desc/asc，默认desc。

## 请求消息

无

## 响应消息

表 4-17 响应参数

参数	参数类型	描述
count	Integer	环境总数。
environments	Array of objects	环境参数，请参考 <a href="#">表4-18</a> 。

表 4-18 environments 参数

参数	参数类型	描述
id	String	环境ID
name	String	环境名称
alias	String	环境别名
description	String	环境描述
project_id	String	项目ID
enterprise_project_id	String	企业项目ID
charge_mode	String	收费模式
vpc_id	String	虚拟私有云ID
base_resources	Array of objects	基础资源，请参考 <a href="#">表4-19</a>
optional_resources	Array of objects	可选资源，请参考 <a href="#">表4-19</a>

参数	参数类型	描述
creator	String	创建人
create_time	Integer	创建时间
update_time	Integer	修改时间

表 4-19 resource 参数

参数	参数类型	描述
id	String	资源ID
name	String	资源名称
type	String	基础资源：cce、ecs 可选资源：dcs、elb等其他类型

## 示例

### 请求示例

无

### 响应示例

```
{
  "environments": [
    {
      "id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
      "name": "development-env",
      "alias": null,
      "description": "",
      "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
      "enterprise_project_id": "0",
      "charge_mode": "provided",
      "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
      "base_resources": [
        {
          "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
          "type": "cce",
          "name": "cce-test"
        }
      ],
      "optional_resources": [
        {
          "id": "9963306a-791e-468c-9306-cf80a9d00298",
          "type": "elb",
          "name": "elb-2dd2"
        }
      ],
      "creator": "ss-test",
      "create_time": 1610418873730,
      "update_time": 1610418873730
    }
  ],
  "count": 1
}
```

## 状态码

表 4-20 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

### 4.2.3 修改环境信息

#### 功能介绍

此API通过环境ID修改环境信息。

#### URI

PUT /v2/{project\_id}/cas/environments/{environment\_id}

表 4-21 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。
environment_id	String	是	环境ID。

## 请求消息

表 4-22 请求参数

参数	参数类型	是否必选	描述
name	String	否	环境名称。 由大小写英文字母、数字、中划线 (-)、下划线 (_) 组成，并以大小写英文字母开头，大小写英文字母或者数字结尾，长度为2~64个字符。
alias	String	否	环境别名。 最大长度为64个字符。
description	String	否	环境描述。 最大长度为128个字符。

## 响应消息

表 4-23 响应参数

参数	参数类型	描述
id	String	环境ID
name	String	环境名称
alias	String	环境别名
description	String	环境描述
project_id	String	项目ID
enterprise_project_id	String	企业项目ID
charge_mode	String	固定值：provided。
vpc_id	String	虚拟私有云ID
base_resources	Array of objects	基础资源，请参考 <a href="#">表4-24</a>
optional_resources	Array of objects	可选资源，请参考 <a href="#">表4-24</a>
creator	String	创建人
create_time	Integer	创建时间
update_time	Integer	修改时间

表 4-24 resources 参数

参数	参数类型	描述
id	String	资源ID
name	String	资源名称
type	String	基础资源：cce、ecs 可选资源：dcs、elb等其他类型

## 示例

### 请求示例

```
{
  "name": "development-env2",
  "description": "here is description"
}
```

### 响应示例

```
{
  "id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
  "name": "development-env2",
  "alias": null,
  "description": "here is description",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "charge_mode": "provided",
  "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
  "base_resources": [
    {
      "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "type": "cce",
      "name": "cce-test"
    }
  ],
  "optional_resources": [
    {
      "id": "9963306a-791e-468c-9306-cf80a9d00298",
      "type": "elb",
      "name": "elb-2dd2"
    }
  ],
  "creator": "ss-test",
  "create_time": 1610418873730,
  "update_time": 1610420992462
}
```

## 状态码

表 4-25 状态码说明

状态码	描述
200	操作成功。

状态码	描述
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.2.4 根据环境 ID 删除环境

### 功能介绍

此API通过环境ID删除环境。

### URI

DELETE /v2/{project\_id}/cas/environments/{environment\_id}

表 4-26 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。
environment_id	String	是	环境ID。

### 请求消息

无

### 响应消息

无

### 示例

请求示例

无

响应示例

无

## 状态码

表 4-27 状态码说明

状态码	描述
204	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.2.5 根据环境 ID 获取环境详细信息

### 功能介绍

此API通过环境ID获取环境详细信息。

### URI

GET /v2/{project\_id}/cas/environments/{environment\_id}

表 4-28 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。
environment_id	String	是	环境ID。

### 请求消息

无

### 响应消息

表 4-29 响应参数

参数	参数类型	描述
id	String	环境ID



参数	参数类型	描述
name	String	环境名称
alias	String	环境别名
description	String	环境描述
project_id	String	项目ID
enterprise_project_id	String	企业项目ID
charge_mode	String	固定值：provided。
vpc_id	String	虚拟私有云ID
base_resources	Array of objects	基础资源，请参考 <a href="#">表4-30</a>
optional_resources	Array of objects	可选资源，请参考 <a href="#">表4-30</a>
creator	String	创建人
create_time	Integer	创建时间
update_time	Integer	修改时间

表 4-30 resources 参数

参数	参数类型	描述
id	String	资源ID
name	String	资源名称
type	String	基础资源：cce、ecs 可选资源：dcs、elb等其他类型

## 示例

### 请求示例

无

### 响应示例

```
{
  "id": "ea011e01-2eb5-453f-87bf-874e4a855abe",
  "name": "dev-env",
  "alias": null,
  "description": "develop environment",
  "project_id": "bf8523d898b64e4eb956e3be3555ca16",
  "enterprise_project_id": "0",
  "charge_mode": "provided",
  "vpc_id": "234241234124xxvasf2342xxxxxxxxxx",
```

```
"base_resources": [  
  {  
    "id": "211112333sd332w23322332",  
    "type": "cce"  
  }  
],  
"optional_resources": null,  
"creator": "string",  
"create_time": 1578984198394,  
"update_time": 1578984198394  
}
```

## 状态码

表 4-31 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.2.6 修改环境资源

### 功能介绍

此API用来修改环境资源。

### URI

PATCH /v2/{project\_id}/cas/environments/{environment\_id}/resources

表 4-32 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户ID。
environment_id	String	是	环境ID。

## 请求消息

表 4-33 请求参数

参数	参数类型	必选	描述
add_base_resources	Array of objects	否	添加基础资源，请参考表4-34。
add_optional_resources	Array of objects	否	添加可选资源，请参考表4-35。
remove_resources	Array of objects	否	移除资源，请参考表4-36。

表 4-34 add\_base\_resources 参数

参数	参数类型	必选	描述
id	String	是	资源ID
name	String	否	资源名称
type	String	是	基础资源：cce、ecs

表 4-35 add\_optional\_resources 参数

参数	参数类型	必选	描述
id	String	是	资源ID
name	String	否	资源名称
type	String	是	可选资源：dcs、elb等其他类型

表 4-36 remove\_resources 参数

参数	参数类型	必选	描述
id	String	是	资源ID
type	String	是	基础资源：cce、ecs 可选资源：dcs、elb等其他类型

## 响应消息

**表 4-37** 参数说明

参数	参数类型	描述
id	String	环境ID
name	String	环境名称
alias	String	环境别名
description	String	环境描述
project_id	String	项目ID
enterprise_project_id	String	企业项目ID
charge_mode	String	固定值：provided。
vpc_id	String	虚拟私有云ID
base_resources	Array of objects	基础资源。请参考 <a href="#">表4-38</a>
optional_resources	Array of objects	可选资源。请参考 <a href="#">表4-38</a>
creator	String	创建人
create_time	Integer	创建时间
update_time	Integer	修改时间

**表 4-38** schemas 参数

参数	参数类型	描述
id	String	资源ID
name	String	资源名称
type	String	基础资源：cce、ecs 可选资源：dcs、elb等其他类型

## 示例

### 请求示例

```
{
  "add_base_resources": [
    {
      "id": "ed2f3420-7031-4d93-b92b-e360cd4acf9e",
      "type": "ecs",
      "name": "ecs-9876"
    }
  ]
}
```

```

    }
  ],
  "add_optional_resources": [
    {
      "id": "default",
      "type": "cse",
      "name": "Cloud Service Engine"
    }
  ],
  "remove_resources": [
    {
      "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "type": "cce"
    }
  ]
}

```

### 响应示例

```

{
  "id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
  "name": "development-env2",
  "alias": null,
  "description": "here is description",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "charge_mode": "provided",
  "vpc_id": "29d55020-ae0e-4a18-871c-93e6976ee7bd",
  "base_resources": [
    {
      "id": "ed2f3420-7031-4d93-b92b-e360cd4acf9e",
      "type": "ecs",
      "name": "ecs-9876"
    }
  ],
  "optional_resources": [
    {
      "id": "9963306a-791e-468c-9306-cf80a9d00298",
      "type": "elb",
      "name": "elb-2dd2"
    },
    {
      "id": "default",
      "type": "cse",
      "name": "Cloud Service Engine"
    }
  ],
  "creator": "ss-test",
  "create_time": 1610418873730,
  "update_time": 1610420992462
}

```

## 状态码

表 4-39 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.3 应用

### 4.3.1 创建应用

#### 功能介绍

应用是一个功能相对完备的业务系统，由一个或多个特性相关的组件组成。  
此API用来创建应用。

#### URI

POST /v2/{project\_id}/cas/applications

表 4-40 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。

#### 请求消息

表 4-41 请求参数

参数	参数类型	必选	描述
name	String	是	应用名称。 由大小写英文字母、数字、中划线(-)、下划线(_)组成，并以大小写英文字母开头，大小写英文字母或者数字结尾，长度为2~64个字符。
description	String	否	应用描述。 最大长度为128个字符。
enterprise_project_id	String	否	企业项目ID。默认值是“0”。

## 响应消息

表 4-42 响应参数

参数	参数类型	描述
id	String	应用ID
name	String	应用名称
description	String	应用描述
creator	String	创建人
project_id	String	项目ID
enterprise_project_id	String	企业项目ID
create_time	Integer	创建时间
update_time	Integer	修改时间
unified_model	String	是否统一模型

## 示例

### 请求示例

```
{
  "name": "app-xpmtii",
  "description": "test"
}
```

### 响应示例

```
{
  "id": "e5213b62-0e3c-476f-9960-3e4108787350",
  "name": "app-xpmtii",
  "description": "test",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "creator": "ss-test",
  "create_time": 1610432385245,
  "update_time": 1610432385245,
  "unified_model": null
}
```

## 状态码

表 4-43 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

状态码	描述
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.3.2 获取所有应用

### 功能介绍

通过此API可以获取所有已经创建的应用。

### URI

GET /v2/{project\_id}/cas/applications

表 4-44 路径参数

参数	参数类型	必选	描述
project_id	String	是	项目ID。

表 4-45 查询参数

参数	参数类型	必选	描述
limit	String	否	指定个数，取值[0, 100]或者1000，指定的值不在该范围内的话相当于赋值为10。默认1000，用于不分页的场景。分页的场景取值请用[0, 100]。
offset	String	否	指定偏移量，表示查询第n条数据后的结果。
order_by	String	否	排序字段，默认按创建时间排序。 排序字段支持枚举值：create_time, name, update_time。传的值不在支持的枚举值范围内的话按默认排序字段排序。
order	String	否	desc/asc，默认desc。



## 请求消息

无

## 响应消息

表 4-46 响应参数

参数	参数类型	描述
count	Integer	应用总数。
applications	Array of object	应用信息，请参考 <a href="#">表4-47</a> 。

表 4-47 applications 参数

参数	参数类型	描述
id	String	应用ID
name	String	应用名称
description	String	应用描述
creator	String	创建人
project_id	String	项目ID
enterprise_project_id	String	企业项目ID
create_time	Integer	创建时间
update_time	Integer	修改时间
unified_model	String	是否开启统一模型
component_count	Integer	组件个数

## 示例

### 请求示例

无

### 响应示例

```
{
  "count": 1,
  "applications": [
    {
      "id": "e5213b62-0e3c-476f-9960-3e4108787350",
      "name": "app-xpmtii",
      "description": "test",
```

```
"project_id": "e7d2e9c589e5445e808a8ff0d1235aca",  
"enterprise_project_id": "0",  
"creator": "ss-test",  
"create_time": 1610432385245,  
"update_time": 1610432385245,  
"unified_model": null,  
"component_count": 0  
}  
]  
}
```

## 状态码

表 4-48 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

### 4.3.3 修改应用信息

#### 功能介绍

此API通过应用ID修改应用信息。

#### URI

PUT /v2/{project\_id}/cas/applications/{application\_id}

表 4-49 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。
application_id	String	是	应用ID。

## 请求消息

表 4-50 请求参数

参数	参数类型	必选	描述
name	String	否	应用名称。 由大小写英文字母、数字、中划线(-)、下划线(_)组成,并以大小写英文字母开头,大小写英文字母或者数字结尾,长度为2~64个字符。
description	String	否	应用描述。 最大长度为128个字符。

## 响应消息

表 4-51 响应参数

参数	参数类型	描述
id	String	应用ID。
name	String	应用名称
description	String	应用描述
creator	String	创建人
project_id	String	项目ID
enterprise_project_id	String	企业项目ID
create_time	Integer	创建时间
update_time	Integer	修改时间
unified_model	String	是否开启统一模型

## 示例

### 请求示例

```
{  
  "name": "app-test",  
  "description": "test"  
}
```

### 响应示例

```
{  
  "id": "e5213b62-0e3c-476f-9960-3e4108787350",  
}
```

```
"name": "app-test",
"description": "test",
"project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
"enterprise_project_id": "0",
"creator": "ss-test",
"create_time": 1610432385245,
"update_time": 1610433070875,
"unified_model": null
}
```

## 状态码

表 4-52 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

### 4.3.4 根据应用 ID 删除应用

#### 功能介绍

此API通过应用ID删除应用。

#### URI

DELETE /v2/{project\_id}/cas/applications/{application\_id}

表 4-53 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。
application_id	String	是	应用ID。

#### 请求消息

无

## 响应消息

无

## 示例

请求示例

无

响应示例

无

## 状态码

表 4-54 状态码说明

状态码	描述
204	操作成功。
400	错误的请求。
404	请求对象不存在。
409	请求与当前对象状态冲突。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.3.5 根据应用 ID 获取应用详细信息

### 功能介绍

此API通过应用ID获取应用详细信息。

### URI

GET /v2/{project\_id}/cas/applications/{application\_id}

表 4-55 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。
application_id	String	是	应用ID。

## 请求消息

无

## 响应消息

表 4-56 响应参数

参数	参数类型	描述
id	String	应用ID
name	String	应用名称
description	String	应用描述
creator	String	创建人
project_id	String	项目ID
enterprise_project_id	String	企业项目ID
create_time	Integer	创建时间
update_time	Integer	修改时间
unified_model	String	是否开启企业项目
component_count	Integer	包含组件个数

## 示例

### 请求示例

无

### 响应示例

```
{
  "id": "e5213b62-0e3c-476f-9960-3e4108787350",
  "name": "app-test",
  "description": "test",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "enterprise_project_id": "0",
  "creator": "ss-test",
  "create_time": 1610432385245,
  "update_time": 1610433070875,
  "unified_model": null,
  "component_count": 0
}
```

## 状态码

表 4-57 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

### 4.3.6 添加或修改应用配置信息

#### 功能介绍

通过此API添加或修改应用配置信息。

应用配置主要是指应用的公共环境变量，该应用下部署的组件可以继承。

#### URI

PUT /v2/{project\_id}/cas/applications/{application\_id}/configuration

表 4-58 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID

#### 请求消息

表 4-59 请求参数

参数	参数类型	必选	描述
environment_id	String	是	环境ID。
configuration	Object	是	应用配置信息，如公共环境变量，请参考 <a href="#">表4-60</a> 。

表 4-60 configuration 参数

参数	参数类型	必选	描述
env	Array of objects	是	应用环境变量，请参考 <a href="#">表4-61</a> 。 如果多个环境变量的名称重复，则只有最后一个会生效。

表 4-61 env 参数

参数	参数类型	必选	描述
name	String	是	环境变量名称。 由字母、数字、下划线、中划线或者点号组成，不能以数字开头，且长度为[1-64]个字符。
value	String	是	环境变量取值。

## 响应消息

表 4-62 响应参数

参数	参数类型	描述
application_id	String	应用ID。
environment_id	String	环境ID。
configuration	Object	应用配置信息。请参考 <a href="#">表4-63</a> 。

表 4-63 configuration 参数

参数	参数类型	描述
env	Array of objects	应用环境变量。请参考 <a href="#">表4-64</a> 。

表 4-64 env 参数

参数	参数类型	描述
name	String	环境变量名称
value	String	环境变量取值



## 示例

### 请求示例

```
{
  "environment_id": "0fa75dfe-0d32-4bc1-848b-8008cf3f2567",
  "configuration": {
    "env": [
      {
        "name": "app-env",
        "value": "env"
      }
    ]
  }
}
```

### 响应示例

```
{
  "application_id": "06cf2fda-af45-44b4-9e22-6294c4804515",
  "environment_id": "0fa75dfe-0d32-4bc1-848b-8008cf3f2567",
  "configuration": {
    "env": [
      {
        "name": "app-env",
        "value": "env"
      }
    ]
  }
}
```

## 状态码

表 4-65 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

### 4.3.7 删除应用配置

#### 功能介绍

通过此API删除应用配置信息。

#### URI

DELETE /v2/{project\_id}/cas/applications/{application\_id}/configuration

表 4-66 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。
application_id	String	是	应用ID。

表 4-67 查询参数

参数	参数类型	必选	描述
environment_id	String	是	环境ID。

## 请求消息

无

## 响应消息

无

## 示例

请求示例

无

响应示例

无

## 状态码

表 4-68 状态码说明

状态码	描述
204	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.3.8 获取应用配置

### 功能介绍

通过此API获取应用配置信息。

### URI

GET /v2/{project\_id}/cas/applications/{application\_id}/configuration

表 4-69 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID。
application_id	String	是	应用ID。

表 4-70 查询参数

参数	参数类型	必选	描述
environment_id	String	否	环境ID，如果未提供，查询所有环境。

### 请求消息

无

### 响应消息

表 4-71 响应参数

参数	参数类型	描述
configuration	Array of objects	配置信息，请参考 <a href="#">表4-72</a> 。

表 4-72 应用配置信息

参数	参数类型	描述
application_id	String	应用ID。
environment_id	String	环境ID。
configuration	Object	应用配置信息，请参考 <a href="#">表4-73</a> 。

表 4-73 configuration 参数

参数	参数类型	描述
env	Array of objects	应用组件环境变量，请参考 <a href="#">表4-74</a> 。

表 4-74 env 参数

参数	参数类型	描述
name	String	环境变量名称
value	String	环境变量取值

## 示例

### 请求示例

无

### 响应示例

```
{
  "configuration": [
    {
      "application_id": "e5213b62-0e3c-476f-9960-3e4108787350",
      "environment_id": "00078e9d-a61c-476e-ac63-a10c9cb2638e",
      "configuration": {
        "env": [
          {
            "name": "env",
            "value": "value1"
          }
        ]
      }
    }
  ]
}
```

## 状态码

表 4-75 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.4 组件

### 4.4.1 应用中创建组件

#### 功能介绍

应用组件是组成应用的某个业务特性实现，以代码或者软件包为载体，可独立部署运行。

此API用来在应用中创建组件。

#### URI

POST /v2/{project\_id}/cas/applications/{application\_id}/components

表 4-76 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID

#### 请求消息

表 4-77 请求参数

参数	参数类型	必选	描述
name	String	是	应用组件名称。 由大小写英文字母、数字、中划线 (-)、下划线 (_) 组成，并以大小写英文字母开头，大小写英文字母或者数字结尾，长度为2~64个字符。
runtime	String	是	运行时，可查询 <a href="#">获取所有支持的应用组件运行时类型</a> 接口返回的type_name获取。
category	String	是	应用组件类型如：Webapp、MicroService、Common。

参数	参数类型	必选	描述
sub_category	String	否	应用组件子类型。 Webapp的子类型有Web。 MicroService的子类型有Java Chassis、Go Chassis、Mesher、Spring Cloud、Dubbo。 Common的子类型可以为空。
description	String	否	描述。 最大长度为128个字符。
source	Object	否	代码/软件包来源，参考表4-78。
build	Object	否	组件构建，参考表4-82。

表 4-78 source 参数

参数	参数类型	必选	描述
kind	String	否	类型，支持源码code和artifact软件包。
spec	Object	否	对应code请参考表4-79；对应artifact请参考表4-80。

表 4-79 code spec 参数

参数	参数类型	必选	描述
repo_type	String	是	代码仓类型，支持GitHub、GitLab、Gitee、Bitbucket。
repo_url	String	是	代码仓url，如：https://github.com/example/demo.git。
repo_ref	String	否	代码分支或者Tag，默认是master。
repo_auth	String	是	授权名称，在授权列表获取。

表 4-80 artifact spec 参数

参数	参数类型	必选	描述
storage	String	是	存储方式，支持对象存储obs。
type	String	是	类别，支持package。

参数	参数类型	必选	描述
url	String	是	软件包地址。
auth	String	否	认证方式，支持iam，none，默认是iam。
properties	Object	否	软件包的其他属性，只有在选择对象存储obs的时候才需要添加。参考 <a href="#">表4-81</a> 。

表 4-81 artifact spec properties 参数

参数	参数类型	必选	描述
endpoint	String	否	obs的终端地址，如： <a href="#">https://obs.region_id.external_domain_name.com</a> 。
bucket	String	否	软件包在obs的桶名。
key	String	否	obs桶中的对象，一般是软件包名，有文件夹的话要加上文件夹的路径。比如test.jar或者demo/test.jar。

表 4-82 build 参数

参数	参数类型	必选	描述
parameters	Map<String, Object>	否	请参考 <a href="#">表4-83</a> 。

表 4-83 parameters 参数

参数	参数类型	必选	描述
build_command	String	否	编译命令。默认： 1. 根目录存在build.sh： <code>./build.sh</code> 2. 根据运行系统，示例如下： Java和Tomcat： <code>mvn clean package</code> Nodejs： <code>npm build</code>
dockerfile_path	String	否	dockerfile地址。默认是根目录./。

参数	参数类型	必选	描述
artifact_namespace	String	否	构建归档组织，默认cas_{project_id}。
cluster_id	String	否	指定构建集群的id。
node_label_selector	Map<String, String>	否	key是标签的键，value是标签的值。

## 响应消息

表 4-84 响应参数

参数	参数类型	描述
id	String	应用组件ID。
name	String	应用组件名称。
status	Integer	取值0或1。 0：表示正常状态。 1：表示正在删除。
runtime	String	运行时。
category	String	应用组件类型如：Webapp、MicroService、Common。
sub_category	String	应用组件子类型。 Webapp的子类型有Web。 MicroService的子类型有Java Chassis、Go Chassis、Mesher、Spring Cloud、Dubbo。 Common的子类型可以为空。
description	String	描述。
project_id	String	项目ID。
application_id	String	应用ID。
source	Object	代码/软件包来源，请参考 <a href="#">表4-85</a> 。
build	Object	构建工程，请参考 <a href="#">表4-89</a> 。
pipeline_ids	id list	流水线Id列表，最多10个。
create_time	Integer	创建时间。
update_time	Integer	修改时间。



参数	参数类型	描述
creator	String	创建者。

表 4-85 source 参数

参数	参数类型	描述
kind	String	类型，支持源码code和artifact软件包。
spec	Object	对应code请参考 <a href="#">表4-86</a> ；对应artifact请参考 <a href="#">表4-87</a> 。

表 4-86 code spec 参数

参数	参数类型	描述
repo_type	String	代码仓类型，支持GitHub、GitLab、Gitee、Bitbucket。
repo_url	String	代码仓url，如：https://github.com/example/demo.git。
repo_ref	String	代码分支或者Tag，默认是master。
repo_auth	String	授权名称，在授权列表获取。

表 4-87 artifact spec 参数

参数	参数类型	描述
storage	String	存储方式。
type	String	类别。
url	String	软件包/源码地址。
auth	String	认证方式。
properties	Object	软件包的其他属性，只有在选择对象存储obs的时候才需要添加。参考 <a href="#">表4-88</a> 。

表 4-88 artifact spec properties 参数

参数	参数类型	必选	描述
endpoint	String	否	obs的终端地址，如：https://obs.region_id.external_domain_name.com。
bucket	String	否	软件包在obs的桶名。
key	String	否	obs桶中的对象，一般是软件包名，有文件夹的话要加上文件夹的路径。比如test.jar或者demo/test.jar。

表 4-89 build 参数

参数	参数类型	描述
ID	String	类型。
parameters	Map<String, Object>	请参考表4-90。

表 4-90 parameters 参数

参数	参数类型	描述
build_cmd	String	编译命令。
dockerfile_path	String	dockerfile地址。
artifact_namespace	String	构建归档组织。
cluster_id	String	指定构建集群的id。
node_selector	Map<String, String>	key是标签的键，value是标签的值。

## 示例

### 请求示例

```
{
  "name": "mycomponent",
  "runtime": "Java8",
  "category": "MicroService",
  "sub_category": "Java Chassis",
  "description": "",
  "build": {
    "parameters": {
      "artifact_namespace": "ns"
    }
  }
},
```

```
"source": {
  "kind": "artifact",
  "spec": {
    "storage": "obs",
    "type": "package",
    "url": "obs://myapp/demo.jar",
    "properties": {
      "bucket": "myapp",
      "key": "demo.jar",
      "endpoint": "https://obs.region_id.external_domain_name.com"
    }
  }
}
```

### 响应示例

```
{
  "id": "384eb8d4-c193-4d84-9558-6fda2366b536",
  "name": "mycomponent",
  "runtime": "Java8",
  "category": "MicroService",
  "sub_category": "Java Chassis",
  "description": "",
  "project_id": "384eb8d4-c193-4d84-9558-6fda23698536",
  "application_id": "a8f7eed5-0aa0-4251-9723-c9119a6bf56d",
  "source": {
    "kind": "artifact",
    "spec": {
      "storage": "obs",
      "type": "package",
      "url": "obs://myapp/demo.jar",
      "properties": {
        "bucket": "myapp",
        "key": "demo.jar",
        "endpoint": "https://obs.region_id.external_domain_name.com"
      }
    }
  },
  "build": {
    "id": "w3dpv7p0t1vpxvey5hjb22iuwxway1vupwx0nae1",
    "parameters": {
      "artifact_namespace": "ns",
      "use_public_cluster": true
    }
  },
  "pipeline_ids": null,
  "status": 0,
  "creator": "xxx",
  "create_time": 1610333934288,
  "update_time": 1610333934288
}
```

## 状态码

表 4-91 状态码说明

状态码	描述
200	操作成功
400	错误的请求
404	请求对象不存在
500	内部错误

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.4.2 获取应用所有组件

### 功能介绍

通过此API获取应用下所有应用组件。

### URI

GET /v2/{project\_id}/cas/applications/{application\_id}/components

表 4-92 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID

表 4-93 查询参数

参数	参数类型	必选	描述
limit	String	否	指定个数，取值[0, 100]或者1000，指定的值不在该范围内的话相当于赋值为10。默认1000，用于不分页的场景。分页的场景取值请用[0, 100]。
offset	String	否	指定偏移量。
order_by	String	否	排序字段，默认按创建时间排序。排序字段支持枚举值：create_time、name、update_time。
order	String	否	desc/asc，默认desc。

### 请求消息

无

## 响应消息

表 4-94 响应参数

参数	参数类型	描述
count	Integer	组件总数。
components	Array of objects	组件参数，请参考 <a href="#">表4-95</a> 。

表 4-95 components 参数

参数	参数类型	描述
id	String	应用组件ID。
name	String	应用组件名称。
status	integer	取值0或1。 0：表示正常状态。 1：表示正在删除。
runtime	String	运行时。
category	String	应用组件类型如：Webapp、MicroService、Common。
sub_category	String	应用组件子类型。 Webapp的子类型有Web。 MicroService的子类型有Java Chassis、Go Chassis、Mesher、Spring Cloud、Dubbo。 Common的子类型可以为空。
application_id	String	应用ID。
source	Object	代码/软件包来源，请参考 <a href="#">表4-96</a> 。
description	String	描述。
creator	String	创建人。
create_time	Integer	创建时间。
update_time	Integer	修改时间。

表 4-96 source 参数

参数	参数类型	描述
kind	String	类型，支持源码code和artifact软件包。

参数	参数类型	描述
spec	Object	对应code请参考 <a href="#">表4-97</a> ；对应artifact请参考 <a href="#">表4-98</a> 。

表 4-97 code spec 参数

参数	参数类型	描述
repo_type	String	代码仓类型，支持GitHub、GitLab、Gitee、Bitbucket
repo_url	String	代码仓url，如：https://github.com/example/demo.git
repo_ref	String	代码分支或者Tag，默认是master
repo_auth	String	授权名称，在授权列表获取

表 4-98 artifact specc 参数

参数	参数类型	描述
storage	String	存储方式
type	String	类别
url	String	软件包地址
auth	String	认证方式
properties	Object	软件包的其他属性，只有在选择对象存储obs的时候才需要添加。参考 <a href="#">表4-99</a>

表 4-99 artifact spec properties 参数

参数	参数类型	描述
endpoint	String	obs的终端地址，比如：https://obs.region_name.external_domain_name.com
bucket	String	软件包在obs的桶名
key	String	obs桶中的对象，一般是软件包名，有文件夹的话要加上文件夹的路径。比如test.jar或者demo/test.jar

## 示例

### 请求示例

无

### 响应示例

```
{
  "components": [
    {
      "id": "384eb8d4-c193-4d84-9558-6fda2366b536",
      "name": "mycomponent",
      "runtime": "Java8",
      "category": "MicroService",
      "sub_category": "Java Chassis",
      "description": "",
      "project_id": "384eb8d4-c193-4d84-9558-6fda23698536",
      "application_id": "a8f7eed5-0aa0-4251-9723-c9119a6bf56d",
      "source": {
        "kind": "artifact",
        "spec": {
          "storage": "obs",
          "type": "package",
          "url": "obs://myapp/demo.jar",
          "properties": {
            "bucket": "myapp",
            "key": "demo.jar",
            "endpoint": "https://obs.region_id.external_domain_name.com"
          }
        }
      }
    }
  ],
  "build": {
    "id": "w3dpv7p0t1vpxvey5hjb22iuwxway1vupwx0nae1",
    "parameters": {
      "artifact_namespace": "ns",
      "use_public_cluster": true
    }
  },
  "pipeline_ids": null,
  "status": 0,
  "creator": "xxx",
  "create_time": 1610333934288,
  "update_time": 1610333934288
}
},
"count": 1
}
```

## 状态码

表 4-100 状态码说明

状态码	描述
200	操作成功
400	错误的请求
404	请求对象不存在
500	内部错误

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

### 4.4.3 根据组件 ID 修改组件信息

#### 功能介绍

此API通过组件ID修改组件信息。

#### URI

PUT /v2/{project\_id}/cas/applications/{application\_id}/components/  
{component\_id}

表 4-101 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID
component_id	String	是	组件ID

#### 请求消息

表 4-102 请求参数

参数	参数类型	必选	描述
name	String	否	应用组件名称。 由大小写英文字母、数字、中划线(-)、下划线(_)组成，并以大小写英文字母开头，大小写英文字母或者数字结尾，长度为2~64个字符。
description	String	否	描述。 最大长度为128个字符。
source	Object	否	代码/软件包来源，请参考 <a href="#">表4-103</a> 。
build	Object	否	构建工程，请参考 <a href="#">表4-107</a> 。

表 4-103 source 参数

参数	参数类型	必选	描述
kind	String	是	类型，支持源码code和artifact软件包。
spec	Object	是	对应code请参考 <a href="#">表4-104</a> ；对应artifact请参考 <a href="#">表4-105</a> 。



表 4-104 code spec 参数

参数	参数类型	必选	描述
repo_type	String	是	代码仓类型，支持GitHub、GitLab、Gitee、Bitbucket
repo_url	String	是	代码仓url，如：https://github.com/example/demo.git
repo_ref	String	否	代码分支或者Tag，默认是master
repo_auth	String	是	授权名称，在授权列表获取

表 4-105 artifact spec 参数

参数	参数类型	必选	描述
storage	String	是	存储方式，支持对象存储obs
type	String	是	类别，支持package
url	String	是	软件包/源码地址
auth	String	否	认证方式，支持iam，none，默认是iam
properties	Object	否	软件包的其他属性，只有在选择对象存储obs的时候才需要添加。 <a href="#">表4-106</a>

表 4-106 artifact spec properties 参数

参数	参数类型	必选	描述
endpoint	String	否	obs的终端地址，比如：https://obs.region_id.external_domain_name.com
bucket	String	否	软件包在obs的桶名
key	String	否	obs桶中的对象，一般是软件包名，有文件夹的话要加上文件夹的路径。比如test.jar或者demo/test.jar

表 4-107 build 参数

参数	参数类型	必选	描述
parameters	Map<String, Object>	否	请参考表4-108。

表 4-108 parameters 参数

参数	参数类型	必选	描述
build_cmd	String	否	编译命令。默认： 1. 根目录存在build.sh: ./build.sh 2. 根据运行系统，示例如下： <ul style="list-style-type: none"><li>• Java和Tomcat: mvn clean package</li><li>• Nodejs: npm build</li></ul>
dockerfile_path	String	否	dockerfile地址。默认是根目录./
artifact_name_space	String	否	构建归档组织。默认cas_{project_id}
cluster_id	String	否	指定构建集群的id。
node_label_selector	Map<String, String>	否	key是标签的键，value是标签的值。

## 响应消息

表 4-109 响应参数

参数	参数类型	描述
id	String	应用组件ID。
name	String	应用组件名称
status	integer	取值0或1。 0: 表示正常状态。 1: 表示正在删除。
runtime	String	运行时。

参数	参数类型	描述
category	String	应用组件类型如：Webapp、MicroService、Common。
sub_category	String	应用组件子类型。 Webapp的子类型有Web。 MicroService的子类型有Java Chassis、Go Chassis、Mesher、Spring Cloud、Dubbo。 Common的子类型可以为空。
description	String	描述。
pipeline_ids	Array<String>	流水线Id列表，最多10个。
project_id	String	项目ID。
application_id	String	应用ID。
source	Object	代码/软件包来源，请参考 <a href="#">表4-110</a> 。
build	Object	构建工程，请参考 <a href="#">表4-113</a> 。
create_time	Integer	创建时间。
update_time	Integer	修改时间。
creator	String	创建者。

表 4-110 source 参数

参数	参数类型	描述
kind	String	类型。
spec	Object	请参考 <a href="#">表4-111</a> 。

表 4-111 spec 参数

参数	参数类型	描述
storage	String	存储方式
type	String	类别
url	String	软件包/源码地址
auth	String	认证方式
properties	Object	软件包的其他属性，只有在选择对象存储obs的时候才需要添加，请参考 <a href="#">表4-112</a>

表 4-112 artifact spec properties 参数

参数	参数类型	描述
endpoint	String	obs的终端地址, 比如: https://obs.region_id.external_domain_name.com
bucket	String	软件包在obs的桶名
key	String	obs桶中的对象, 一般是软件包名, 有文件夹的话要加上文件夹的路径。比如test.jar或者demo/test.jar

表 4-113 build 参数

参数	参数类型	描述
ID	String	类型
parameters	Map<String, Object>	请参考表4-114

表 4-114 parameters 参数

参数	参数类型	描述
build_cmd	String	编译命令。
dockerfile_path	String	dockerfile地址。
artifact_namepace	String	构建归档组织。
cluster_id	String	指定构建集群的id。
node_label_selector	Map<String, String>	key是标签的键, value是标签的值。

## 示例

### 请求示例

```
{
  "name": "test-micro",
  "description": "",
  "source": {
    "kind": "artifact",
    "spec": {
      "storage": "obs",
      "type": "package",
      "url": "obs://test-soft/fusionweather-1.0.0.jar",
      "auth": "iam",
      "webUrl": "",
      "properties": {
```

```
        "bucket": "test-soft",
        "endpoint": "https://obs.region_id.development.com:443",
        "key": "fusionweather-1.0.0.jar"
      }
    },
    "build": {
      "parameters": {
        "artifact_namespace": "ns",
        "use_public_cluster": false,
        "cluster_id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
        "cluster_name": "cce-test",
        "cluster_type": "VirtualMachine"
      }
    },
    "pipeline_ids": null
  }
}
```

### 响应示例

```
{
  "id": "1cfdda6f-84cd-4ead-8e09-628fabf662e2",
  "name": "test-micro",
  "runtime": "Java8",
  "category": "MicroService",
  "sub_category": "Java Chassis",
  "description": "",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "application_id": "4d92833a-fa05-4fc0-a761-e67620022310",
  "source": {
    "kind": "artifact",
    "spec": {
      "storage": "obs",
      "type": "package",
      "url": "obs://test-soft/fusionweather-1.0.0.jar",
      "auth": "iam",
      "webUrl": "",
      "properties": {
        "bucket": "test-soft",
        "endpoint": "https://obs.region_id.development.com:443",
        "key": "fusionweather-1.0.0.jar"
      }
    }
  },
  "build": {
    "id": "zpdkj42b3rnko8bmyojzt1ng828ubychw13j8flv",
    "parameters": {
      "artifact_namespace": "ns",
      "use_public_cluster": false,
      "cluster_id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "cluster_name": "cce-test",
      "cluster_type": "VirtualMachine"
    }
  },
  "pipeline_ids": null,
  "status": 0,
  "creator": "ss-test",
  "create_time": 1610331760105,
  "update_time": 1610519881943
}
```

## 状态码

表 4-115 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

### 4.4.4 根据应用组件 ID 删除应用组件

#### 功能介绍

此API通过应用组件ID删除应用组件。

#### URI

DELETE /v2/{project\_id}/cas/applications/{application\_id}/components/  
{component\_id}

表 4-116 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID
component_id	String	是	组件ID

表 4-117 查询参数

参数	参数类型	必选	描述
force	boolean	否	是否强制删除

#### 请求消息

无

## 响应消息

无

## 示例

请求示例

无

响应示例

无

## 状态码

表 4-118 状态码说明

状态码	描述
204	操作成功
400	错误的请求
404	请求对象不存在
409	请求与当前对象状态冲突
500	内部错误

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.4.5 根据组件 ID 获取应用组件信息

### 功能介绍

通过组件ID获取应用组件信息。

### URI

GET /v2/{project\_id}/cas/applications/{application\_id}/components/{component\_id}

表 4-119 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID

参数	参数类型	必选	描述
component_id	String	是	组件ID

## 请求消息

无

## 响应消息

表 4-120 响应参数

参数	参数类型	描述
id	String	应用组件ID。
name	String	应用组件名称。
status	integer	取值0或1。 0: 表示正常状态。 1: 表示正在删除。
runtime	String	运行时。
category	String	应用组件类型如: Webapp、MicroService、Common。
sub_category	String	应用组件子类型。 Webapp的子类型有Web。 MicroService的子类型有Java Chassis、Go Chassis、Mesher、Spring Cloud、Dubbo。 Common的子类型可以为空。
description	String	描述。
pipeline_ids	Array<String>	流水线Id列表, 最多10个。
project_id	String	项目ID。
application_id	String	应用ID。
source	Object	代码/软件包来源, 请参考 <a href="#">表4-121</a> 。
build	Object	构建工程, 请参考 <a href="#">表4-124</a> 。
create_time	Integer	创建时间。
update_time	Integer	修改时间。
creator	String	创建者。



表 4-121 source 参数

参数	参数类型	描述
kind	String	类型。
spec	Object	请参考 <a href="#">表4-122</a> 。

表 4-122 spec 参数

参数	参数类型	描述
storage	String	存储方式
type	String	类别
url	String	软件包/源码地址
auth	String	认证方式
properties	Object	软件包的其他属性，只有在选择对象存储obs的时候才需要添加。 <a href="#">表4-123</a>

表 4-123 artifact spec properties 参数

参数	参数类型	描述
endpoint	String	obs的终端地址，比如： <a href="https://obs.region_id.external_domain_name.com">https://obs.region_id.external_domain_name.com</a>
bucket	String	软件包在obs的桶名
key	String	obs桶中的对象，一般是软件包名，有文件夹的话要加上文件夹的路径。比如test.jar或者demo/test.jar

表 4-124 bulid 参数

参数	参数类型	描述
ID	String	类型
parameters	Map<String,String>	请参考 <a href="#">表4-125</a>

表 4-125 parameters 参数

参数	参数类型	描述
build_cmd	String	编译命令。
dockerfile_path	String	dockerfile地址。
artifact_names pace	String	构建归档组织。
cluster_id	String	指定构建集群的id。
node_label_sel ector	Map<String, String>	key是标签的键，value是标签的值。

## 示例

### 请求示例

无

### 响应示例

```
{
  "id": "1cfdda6f-84cd-4ead-8e09-628fabf662e2",
  "name": "test-micro",
  "runtime": "Java8",
  "category": "MicroService",
  "sub_category": "Java Chassis",
  "description": "",
  "project_id": "e7d2e9c589e5445e808a8ff0d1235aca",
  "application_id": "4d92833a-fa05-4fc0-a761-e67620022310",
  "source": {
    "kind": "artifact",
    "spec": {
      "storage": "obs",
      "type": "package",
      "url": "obs://test-soft/fusionweather-1.0.0.jar",
      "auth": "iam",
      "webUrl": "",
      "properties": {
        "bucket": "test-soft",
        "endpoint": "https://obs.region_id.development.com:443",
        "key": "fusionweather-1.0.0.jar"
      }
    }
  },
  "build": {
    "id": "zpdkj42b3rnko8bmyojzt1ng828ubychw13j8flv",
    "parameters": {
      "artifact_namespace": "ns",
      "use_public_cluster": false,
      "cluster_id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "cluster_name": "cce-test",
      "cluster_type": "VirtualMachine"
    }
  },
  "pipeline_ids": null,
  "status": 0,
  "creator": "ss-test",
  "create_time": 1610331760105,
  "update_time": 1610519881943
}
```

## 状态码

表 4-126 状态码说明

状态码	描述
200	操作成功
400	错误的请求
404	请求对象不存在
500	内部错误

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.5 实例

### 4.5.1 创建组件实例

#### 功能介绍

此API用来创建应用组件实例。

#### URI

POST /v2/{project\_id}/cas/applications/{application\_id}/components/{component\_id}/instances

表 4-127 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID
component_id	String	是	组件ID

## 请求消息

表 4-128 请求参数

参数	参数类型	必选	描述
name	String	是	应用组件实例名称。 由小写英文字母、数字、中划线(-)组成，并以小写英文字母开头，小写英文字母或者数字结尾，长度为2~63个字符。
environment_id	String	是	环境ID。
replica	Integer	是	实例副本数。
flavor_id	String	是	资源规格，可查询 <a href="#">获取所有支持的应用资源规格</a> 接口获取系统预定义好的资源规格。 如需自定义资源规格，格式为：CUSTOM-xxG:xxC-xxC:xxGi-xxGi。其中： <ul style="list-style-type: none"><li>xxG表示给组件实例分配的存储的大小，为保留字段，当前未生效，可以指定为一个固定的数字。</li><li>xxC-xxC用于指定给组件实例分配的cpu最大核数和最小核数。</li><li>xxGi-xxGi用于指定给组件实例分配的最大内存和最小内存。</li></ul> 例如，CUSTOM-10G:0.5C-0.25C:1.6Gi-0.8Gi，表示给组件实例分配的CPU核数最大为0.5核，最小为0.25核，分配的最大内存为1.6Gi，最小为0.8Gi。
artifacts	Map<String, Object>	是	组件部署件。key为组件名，对于Docker多容器场景，key为容器名称。请参考 <a href="#">表4-145</a> 。
version	String	是	应用组件版本号，满足版本语义，如1.0.0。
configuration	Object	否	配置参数，如设置环境变量，部署配置，运维监控等，默认空。请参考 <a href="#">表4-129</a> 。
description	String	否	描述。 最大长度为128个字符。
external_accesses	Array of objects	否	外网访问，参考 <a href="#">表4-146</a> 。
refer_resources	Array of objects	是	部署资源，参考 <a href="#">表4-147</a> 。

表 4-129 configuration 参数

参数	参数类型	必选	描述
env	Array of objects	否	环境变量，参考 <a href="#">表4-130</a>
storage	Array of objects	否	数据存储配置，参考 <a href="#">表4-131</a>
strategy	Object	否	升级策略，参考 <a href="#">表4-132</a>
lifecycle	Object	否	生命周期，参考 <a href="#">表4-133</a>
scheduler	Object	否	调度策略，参考 <a href="#">表4-134</a>
probes	Object	否	健康检查，参考 <a href="#">表4-135</a>

表 4-130 environment 参数

参数	参数类型	必选	描述
name	String	是	环境变量名。 由字母、数字、下划线、中划线或者点号组成，不能以数字开头，且长度为1~64个字符。
value	String	是	环境变量值。

表 4-131 storage 参数

参数	参数类型	必选	描述
type	String	是	存储类型，取值为： <ul style="list-style-type: none"><li>HostPath：主机路径挂载</li><li>EmptyDir：临时路径挂载</li><li>ConfigMap：配置项挂载</li><li>Secret：密钥挂载</li><li>PersistentVolumeClaim：云存储挂载</li></ul>
parameters	Object	是	存储的参数，参考 <a href="#">表4-144</a>
mounts	Array of objects	是	挂载到容器的目录，参考 <a href="#">表4-143</a>

表 4-132 strategy 参数

参数	参数类型	必选	描述
upgrade	String	否	升级策略，默认为RollingUpdate。取值为Recreate和RollingUpdate，分别为替换升级和滚动升级。

表 4-133 lifecycle 参数

参数	参数类型	必选	描述
entrypoint	Object	否	启动命令，参考 <a href="#">表4-141</a>
post-start	Object	否	启动后处理，参考 <a href="#">表4-141</a>
pre-stop	Object	否	停止前处理，参考 <a href="#">表4-141</a>

表 4-134 scheduler 参数

参数	参数类型	必选	描述
affinity	Object	否	亲和性，参考 <a href="#">表4-140</a>
anti-affinity	Object	否	反亲和性，参考 <a href="#">表4-140</a>

表 4-135 probes 参数

参数	参数类型	必选	描述
livenessProbe	Object	否	组件存活探针，参考 <a href="#">表4-136</a>
readinessProbe	Object	否	组件业务探针，参考 <a href="#">表4-136</a>

表 4-136 probes\_object 参数

参数	参数类型	必选	描述
type	String	是	取值可以为http, tcp和command。对应的检查方式分别为HTTP请求检查，TCP端口检查和执行命令检查。

参数	参数类型	必选	描述
parameters	Object	是	检查参数。 <ul style="list-style-type: none"><li>• type为http, 参考<a href="#">表4-137</a>。</li><li>• type为tcp, 参考<a href="#">表4-139</a>。</li><li>• type为command, 参考<a href="#">表4-138</a>。</li></ul>
delay	integer	否	表示启动后多久开始探测。
timeout	integer	否	表示探测超时时间。

表 4-137 probes\_param\_http 参数

参数	参数类型	必选	描述
scheme	String	是	HTTP或者HTTPS。
host	String	否	默认为POD的IP。可以指定自定义的IP。
port	integer	是	端口。
path	String	是	请求路径。

表 4-138 probes\_param\_command 参数

参数	参数类型	必选	描述
command	Array of String	是	命令列表

表 4-139 probes\_param\_tcp 参数

参数	参数类型	必选	描述
port	integer	是	端口

表 4-140 scheduler\_affinity 参数

参数	参数类型	必选	描述
az	Array of String	否	az名称列表
node	Array of String	否	节点私有IP列表
application	Array of String	否	组件实例名列表

表 4-141 lifecycle\_entrypoint 参数

参数	参数类型	必选	描述
command	Array of String	是	可执行的命令。
args	Array of String	是	运行参数。

表 4-142 lifecycle\_process 参数

参数	参数类型	必选	描述
type	String	是	取值为command或者http。command为执行命令行，http为发送http请求。
parameters	Object	是	启动后处理或者停止前处理参数。参考 <a href="#">表 4-143</a> 。

表 4-143 lifecycle\_process\_parameters 参数

参数	参数类型	必选	描述
command	Array of String	是	命令参数，比如["sleep", "1"]。适用于command类型。
host	String	否	默认为POD实例的IP地址。也可以自己指定。适用于http类型。
port	integer	是	端口号。适用于http类型。
path	String	是	请求url。适用于http类型。



表 4-144 storage\_parameters 参数

参数	参数类型	必选	描述
path	String	否	主机路径，适用于HostPath的存储类型
name	String	否	配置项或者密钥的名字，适用于ConfigMap和Secret的存储类型
claimName	String	否	PVC的名字，适用于PersistentVolumeClaim的存储类型

表 4-145 artifact 参数

参数	参数类型	必选	描述
storage	String	是	存储方式，支持swr、obs。
type	String	是	类别，虚拟机部署为package，容器部署为image。
url	String	是	软件包/镜像地址。对于基于虚拟机部署的组件，值为软件包的地址。对于基于容器部署的组件，值为镜像的地址或者“组件名:v\${index}”，后者表示使用组件来源中的源码或者软件包自动构建生成的镜像。
auth	String	是	认证方式，支持iam和none，默认为iam。
version	String	否	版本号。
properties	Map<String, Object>	否	属性。

表 4-146 external\_accesses 参数

参数	参数类型	必选	描述
protocol	String	否	协议，支持http、https。
address	String	否	访问地址。
forward_port	integer	否	端口号。

表 4-147 refer\_resources 参数

参数	参数类型	必选	描述
id	String	是	资源ID。 注意：如果type为ecs的时候，该字段的值固定为“Default”。
type	String	是	基础资源：cce、ecs。 可选资源：dcs、elb等其他类型。
refer_alias	String	否	应用别名，dcs时才提供，支持“distributed_session”、“distributed_cache”、“distributed_session, distributed_cache”，默认值是“distributed_session, distributed_cache”。
parameters	Map<String, Object>	否	引用资源参数。 <b>须知</b> <ul style="list-style-type: none"><li>当type是cce的时候，该参数必填，需要指定部署组件所在的集群的命名空间，比如{"namespace": "default"}</li><li>当type为ecs的时候，该参数必填，需要指定组件部署在哪些主机上，比如{"hosts": ["04d9f887-9860-4029-91d1-7d3102903a69", "04d9f887-9860-4029-91d1-7d3102903a70"]}</li></ul>

## 响应消息

表 4-148 响应参数说明

参数	参数类型	描述
job_id	String	Job ID，用于查询创建任务信息
instance_id	String	应用组件实例ID

## 示例

### 请求示例

```
{
  "name": "component-instance-name",
  "environment_id": "6e763000-9128-4a9d-adea-34c42cc5344d",
  "flavor_id": "MICRO-5G:0.5C:1G",
  "replica": 1,
  "artifacts": {
    "container-name": {
      "storage": "swr",
      "type": "image",
      "url": "{IP}:20202/xxxxxxx/junit-platformstest:1.0",
      "auth": "iam"
    }
  }
}
```

```
    },
    "version": 1,
    "description": "instance desc",
    "configuration": {
      "env": [
        {
          "name": "log-level",
          "value": "warn"
        }
      ]
    },
    "refer_resources": [
      {
        "id": "b6862a62-d916-11e9-bdf1-0255ac101fd9",
        "type": "cce",
        "parameters": {
          "namespace": "default"
        }
      },
      {
        "id": "8c0a45cc-626f-4d65-8257-507ee059aa9a",
        "type": "elb"
      }
    ]
  }
}
```

### 响应示例

```
{
  "instance_id": "89f5baf5-efe4-4f12-9c0d-734d2af5a184",
  "job_id": "JOB66761060-f209-407c-a093-4df6f531b9dc"
}
```

## 状态码

表 4-149 状态码说明

状态码	描述
200	操作成功
400	错误的请求
404	请求对象不存在
500	内部错误

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.5.2 获取应用组件实例

### 功能介绍

通过此API获取组件下的所有组件实例。

## URI

GET /v2/{project\_id}/cas/applications/{application\_id}/components/{component\_id}/instances

表 4-150 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID
component_id	String	是	组件ID

表 4-151 查询参数

参数	参数类型	必选	描述
limit	Integer	否	指定个数，取值[0, 100]或者1000，指定的值不在该范围内的话相当于赋值为10。默认1000，用于不分页的场景。分页的场景取值请用[0, 100]。
offset	Integer	否	指定偏移量。
order_by	String	否	排序字段，默认按创建时间排序。排序字段支持枚举值：create_time、name、version、update_time。
order	String	否	desc/asc，默认desc。

## 请求消息

无

## 响应消息

表 4-152 响应参数

参数	参数类型	描述
count	Integer	实例总数
instances	Array of objects	实例参数，请参考 <a href="#">表4-153</a> 。

表 4-153 instance 参数

参数	参数类型	描述
id	String	应用组件实例ID。
application_id	String	应用ID。
component_id	String	组件ID。
name	String	应用组件实例名称。
environment_id	String	应用组件环境ID。
platform_type	String	运行平台类型。 应用可以在不同的平台上运行，可选用的平台的类型有以下几种：cce、vmapp。
artifacts	Map<String, Object>	组件部署件。key为组件component_name，对于Docker多容器场景，key为容器名称,请参考表4-154。
version	String	应用组件版本号。
create_time	Integer	创建时间。
update_time	Integer	修改时间。
external_accesses	Array of objects	访问方式，请参考表4-155。
status_detail	Object	状态详情，请参考表4-156。

表 4-154 artifact 参数

参数	参数类型	描述
storage	String	存储方式，支持swr、obs
type	String	类别，虚拟机部署为package，容器部署为image
url	String	软件包/镜像地址
auth	String	认证方式，支持iam和none，默认为iam
version	String	版本号
properties	Map<String, String>	属性

表 4-155 external\_accesses 参数

参数	参数类型	描述
protocol	String	协议
address	String	访问地址
forward_port	Integer	应用组件进程监听端口
type	String	类型
status	String	状态
create_time	Integer	创建时间
update_time	Integer	修改时间

表 4-156 status\_detail 参数

参数	参数类型	描述
enterprise_project_id	String	企业项目ID
status	String	实例状态
available_replica	Integer	正常实例副本数
replica	Integer	实例副本数
fail_detail	String	失败描述
last_job_id	String	最近Job ID
last_job_statuses	String	最近Job的状态

## 示例

### 请求示例

无

### 响应示例

```
{
  "instances": [
    {
      "id": "11eddb33-140b-4e51-b1e2-6ec265373ca3",
      "application_id": "4d92833a-fa05-4fc0-a761-e67620022310",
      "component_id": "1cfdda6f-84cd-4ead-8e09-628fabf662e2",
      "name": "test-micro-test-env-7iafjk",
      "environment_id": "61b81021-21d5-42f3-b80e-0b6bd10dbf7d",
      "platform_type": "cce",
      "version": "1.0.0",
      "artifacts": {
```

```
    "test-micro": {
      "storage": "swr",
      "type": "image",
      "url": "swr.region_id.development.com/ns/examples:v1",
      "auth": "iam",
      "version": "1.0.0",
      "properties": {}
    }
  },
  "create_time": 1610331819996,
  "update_time": 1610331830398,
  "status_detail": {
    "status": "RUNNING",
    "replica": 1,
    "available_replica": 1,
    "fail_detail": null,
    "last_job_id": "JOB7bc0366c-dc43-41f9-b65f-8d6078038488",
    "last_job_status": "SUCCEEDED",
    "enterprise_project_id": "0"
  }
},
"count": 1
}
```

## 状态码

表 4-157 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

### 4.5.3 对组件实例的操作

#### 功能介绍

通过此API操作组件实例。

#### URI

POST /v2/{project\_id}/cas/applications/{application\_id}/components/{component\_id}/instances/{instance\_id}/action

表 4-158 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID
component_id	String	是	组件ID
instance_id	String	是	组件实例ID

## 请求消息

表 4-159 请求参数

参数	参数类型	必选	描述
action	String	是	操作，支持start、stop、restart、scale、rollback。
parameters	Object	否	操作参数，action为scale和rollback时必须，请参考 <a href="#">表4-160</a> 。

表 4-160 parameters 参数说明

参数	参数类型	必选	描述
replica	Integer	否	实例数，在scale操作时必须。
hosts	Array	否	ECS id列表，指定虚拟机伸缩的时候部署的ECS主机，在scale操作时必须。
version	String	否	版本号，在rollback操作时必须，通过 <a href="#">获取组件实例快照</a> 获取。

## 响应消息

表 4-161 响应参数

参数	参数类型	描述
job_id	String	Job ID。

## 示例

### 请求示例

```
{  
  "action": "scale",  
}
```



```
"parameters": {  
  "replica": 4  
}
```

### 响应示例

```
{  
  "job_id": "JOB66761060-f209-407c-a093-4df6f531b9dc"  
}
```

## 状态码

表 4-162 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

### 4.5.4 修改应用组件实例

#### 功能介绍

通过此API修改应用组件实例。

#### URI

PUT /v2/{project\_id}/cas/applications/{application\_id}/components/  
{component\_id}/instances/{instance\_id}

表 4-163 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID
component_id	String	是	组件ID
instance_id	String	是	组件实例ID

## 请求消息

表 4-164 请求参数

参数	参数类型	必选	描述
version	String	是	应用组件版本号，满足版本语义，如1.0.1。
flavor_id	String	否	<p>资源规格，可查询<a href="#">获取所有支持的应用资源规格</a>接口获取系统预定义好的资源规格。</p> <p>如果需要自定义资源规格的话，自定义资源规格的格式为：CUSTOM-xxG:xxC:xxGi-xxGi。其中：</p> <ul style="list-style-type: none"> <li>xxG表示给组件实例分配的存储的大小，为保留字段，当前未生效，可以指定为一个固定的数字。</li> <li>xxC-xxC用于指定给组件实例分配的cpu最大核数和最小核数。</li> <li>xxGi-xxGi用于指定给组件实例分配的最大内存和最小内存。</li> </ul> <p>例如，CUSTOM-10G:0.5C-0.25C:1.6Gi-0.8Gi，表示给组件实例分配的cpu核数最大为0.5核，最小为0.25核，分配的最大内存为1.6Gi，最小为0.8Gi。</p>
artifacts	Map<String, Object>	否	组件部署件。key为组件名，对于Docker多容器场景，key为容器名称,请参考 <a href="#">表4-165</a> 。
configuration	Map<String, Object>	否	配置参数，如设置环境变量，部署配置, 运维监控等，默认空。请参考 <a href="#">创建组件实例的表4-129</a> 。
description	String	否	描述。 最大长度为128个字符。
external_accesses	Array of objects	否	访问方式，请参考 <a href="#">表4-166</a> 。
refer_resources	Array of objects	否	部署资源，请参考 <a href="#">表4-167</a> 。

表 4-165 artifact 参数

参数	参数类型	必选	描述
storage	String	是	存储方式，支持swr和obs。
type	String	是	类别，虚拟机部署为package，容器部署为image。

参数	参数类型	必选	描述
url	String	是	软件包/镜像地址。对于基于虚拟机部署的组件，值为软件包的地址。对于基于容器部署的组件，值为镜像的地址或者“组件名:v\${index}”，后者表示使用组件来源中的源码或者软件包自动构建生成的镜像。
auth	String	是	认证方式，支持iam和none，默认为iam。
version	String	否	版本号。
properties	Map<String, String>	否	属性。

表 4-166 external\_accesses 参数

参数	参数类型	必选	描述
protocol	String	否	协议，支持http和https
address	String	否	访问地址，如www.example.com
forward_port	Integer	否	应用组件进程监听端口

表 4-167 refer\_resources 参数

参数	参数类型	必选	描述
id	String	是	资源ID。
type	String	是	资源类型，含dcs等类型。
refer_alias	String	否	应用别名，dcs时才提供，支持“distributed_session”、“distributed_cache”、“distributed_session, distributed_cache”，默认值是“distributed_session, distributed_cache”。
parameters	Map<String, Object>	否	引用资源参数。

## 响应消息

表 4-168 响应参数说明

参数	参数类型	描述
job_id	String	Job ID，用于查询创建任务信息。

## 示例

### 请求示例

```
{
  "description": "",
  "artifacts": {
    "test-micro": {
      "storage": "swr",
      "type": "image",
      "url": "swr.roma-dev-1.roma.development.com/ns1/examples:v2",
      "auth": "iam",
      "version": "1.0.0",
      "properties": {}
    }
  },
  "refer_resources": [
    {
      "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
      "type": "cce",
      "parameters": {
        "namespace": "default"
      }
    },
    {
      "id": "default",
      "type": "cse"
    }
  ],
  "version": "1.0.2"
}
```

### 响应示例

```
{
  "job_id": "JOB66761060-f209-407c-a093-4df6f531b9dc"
}
```

## 状态码

表 4-169 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在
500	内部错误

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.5.5 删除应用组件实例

### 功能介绍

通过此API删除应用组件实例。

### URI

DELETE /v2/{project\_id}/cas/applications/{application\_id}/components/{component\_id}/instances/{instance\_id}

表 4-170 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID
component_id	String	是	组件ID
instance_id	String	是	组件实例ID

表 4-171 查询参数

参数	参数类型	必选	描述
force	boolean	否	是否强制删除。

### 请求消息

无

### 响应消息

表 4-172 响应参数说明

参数	参数类型	描述
job_id	String	Job ID

### 示例

请求示例

无

### 响应示例

```
{
  "job_id": "JOB66761060-f209-407c-a093-4df6f531b9dc"
}
```

## 状态码

表 4-173 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在
500	内部错误

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.5.6 根据实例 ID 获取实例详细信息

### 功能介绍

此API通过实例ID获取实例详细信息。

### URI

GET /v2/{project\_id}/cas/applications/{application\_id}/components/{component\_id}/instances/{instance\_id}

表 4-174 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID
component_id	String	是	组件ID
instance_id	String	是	组件实例ID

### 请求消息

无

## 响应消息

表 4-175 响应参数

参数	参数类型	描述
id	String	应用组件实例ID
name	String	应用组件实例名称
environment_id	String	应用组件环境ID
platform_type	String	运行平台类型。 应用可以在不同的平台上运行，可选用的平台的类型有以下几种：cce、vmapp
description	String	实例描述
flavor_id	String	资源规格
artifacts	Map<String, Object>	组件部署件。key为组件component_name，对于Docker多容器场景，key为容器名称,请参考 <a href="#">表 4-176</a>
version	String	应用组件版本号
configuration	Map<String, Object>	应用组件配置，如环境变量
creator	String	创建人
create_time	Integer	创建时间
update_time	Integer	修改时间
external_accesses	Array of objects	访问方式，请参考 <a href="#">表4-177</a>
refer_resources	Array of objects	部署资源，请参考 <a href="#">表4-178</a>
status_detail	Object	状态详情，请参考 <a href="#">表4-179</a>

表 4-176 artifact 参数

参数	参数类型	描述
storage	String	存储方式，支持swr和obs
type	String	类别，虚拟机部署为package，容器部署为image
url	String	软件包/镜像地址
auth	String	认证方式，支持iam和none，默认为iam

参数	参数类型	描述
version	String	版本号
properties	Map<String,String>	属性

表 4-177 external\_accesses 参数

参数	参数类型	描述
protocol	String	协议
address	String	访问地址
forward_port	Integer	应用组件进程监听端口
type	String	类型
status	String	状态
create_time	Integer	创建时间
update_time	Integer	修改时间

表 4-178 refer\_resources 参数

参数	参数类型	描述
id	String	资源ID。
type	String	基础资源：cce、ecs。 可选资源：dcs、elb等其他类型。
refer_alias	String	应用别名，dcs时才提供，支持“distributed_session”、“distributed_cache”、“distributed_session, distributed_cache”，默认值是“distributed_session, distributed_cache”。
parameters	Map<String,Object>	引用资源参数。



表 4-179 status\_detail 参数

参数	参数类型	描述
enterprise_project_id	String	企业项目ID
status	String	实例状态
available_replica	Integer	正常实例副本数
replica	Integer	实例副本数
fail_detail	String	失败描述
last_job_id	String	最近Job ID
last_job_statuses	String	最近Job的状态

## 示例

### 请求示例

无

### 响应示例

```
{
  "id": "11eddb33-140b-4e51-b1e2-6ec265373ca3",
  "name": "test-micro-test-env-7iafjk",
  "environment_id": "61b81021-21d5-42f3-b80e-0b6bd10dbf7d",
  "platform_type": "cce",
  "flavor_id": "CUSTOM-10G:250m-250m:0.5Gi-0.5Gi",
  "artifacts": {
    "test-micro": {
      "storage": "swr",
      "type": "image",
      "url": "swr.region_id.development.com/ns/examples:v2",
      "auth": "iam",
      "version": "1.0.0",
      "properties": {}
    }
  },
  "version": "1.0.2",
  "description": "",
  "configuration": {
    "env": [
      {
        "name": "PAAS_PROJECT_ID",
        "value": "e7d2e9c589e5445e808a8ff0d1235aca",
        "internal": true
      }
    ],
    "lifecycle": {
      "post-start": null,
      "pre-stop": null
    },
    "scheduler": {
      "affinity": {
        "az": [],
        "node": []
      }
    }
  }
}
```

```
    "application": [],
  },
  "anti-affinity": {
    "az": [],
    "node": [],
    "application": []
  },
  "isSynchronized": true
},
"strategy": {
  "upgrade": "RollingUpdate",
  "spec": {
    "maxSurge": 0,
    "maxUnavailable": 1
  }
},
"creator": null,
"create_time": 1610331819996,
"update_time": 1610522676792,
"external_accesses": null,
"refer_resources": [
  {
    "id": "523498f1-36c4-11eb-ae36-0255ac1000c2",
    "type": "cce",
    "parameters": {
      "namespace": "default"
    }
  },
  {
    "id": "default",
    "type": "cse"
  }
],
"status_detail": {
  "status": "RUNNING",
  "replica": 1,
  "available_replica": 1,
  "fail_detail": null,
  "last_job_id": "JOB16c33466-41e6-4c5b-a9ac-2059253a1e5c",
  "last_job_status": "SUCCEEDED",
  "enterprise_project_id": "0"
}
}
```

## 状态码

表 4-180 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.5.7 获取组件实例快照

### 功能介绍

通过此API获取应用组件实例的快照信息。

实例快照是指升级后，之前版本的实例信息，可以用于回滚。

### URI

GET /v2/{project\_id}/cas/applications/{application\_id}/components/{component\_id}/instances/{instance\_id}/snapshots

表 4-181 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
application_id	String	是	应用ID
component_id	String	是	组件ID
instance_id	String	是	组件实例ID

表 4-182 查询参数

参数	参数类型	必选	描述
limit	Integer	否	指定个数，取值[0, 100]或者1000，指定的值不在该范围内的话相当于赋值为10。默认1000，用于不分页的场景。分页的场景取值请用[0, 100]。
offset	Integer	否	指定偏移量。
order_by	String	否	排序字段，默认按创建时间排序。
order	String	否	desc/asc，默认desc。

### 请求消息

无

### 响应消息

表 4-183 响应参数说明

参数	参数类型	描述
count	Integer	快照总数。

参数	参数类型	描述
snapshots	Array of Objects	快照参数，请参考 <a href="#">表4-184</a> 。

表 4-184 snapshot 参数说明

参数	参数类型	描述
create_time	Integer	创建时间
description	String	描述
instance_id	String	应用组件实例ID
version	String	版本号

## 示例

### 请求示例

无

### 响应示例

```
{
  "snapshots": [
    {
      "version": "1.0.0",
      "description": "{\"action\":\"New Version: 1.0.0\"}",
      "instance_id": "11eddb33-140b-4e51-b1e2-6ec265373ca3",
      "create_time": 1610331830841
    }
  ],
  "count": 1
}
```

## 状态码

表 4-185 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在。
500	内部错误。

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

## 4.6 Job

### 4.6.1 获取部署任务详细信息

#### 功能介绍

通过此API获取部署任务详细信息。

#### URI

GET /v2/{project\_id}/cas/jobs/{job\_id}

表 4-186 路径参数

参数	参数类型	必选	描述
project_id	String	是	租户项目ID
job_id	String	是	部署任务ID

表 4-187 查询参数

参数	参数类型	必选	描述
instance_id	String	否	应用组件实例ID。
limit	Integer	否	指定个数。
offset	Integer	否	指定偏移量。
desc	String	否	是否降序： true=desc, false=asc。

#### 请求消息

无

## 响应消息

表 4-188 参数说明

参数	参数类型	描述
task_count	Integer	部署任务数量。
job	Object	构建工程参数，请参考表4-189。
tasks	Array<Object>	部署任务参数，请参考表4-190。

表 4-189 job 参数

参数	参数类型	描述
CREATED_BY	String	创建者
EXECUTION_STATUS	String	执行状态： <ul style="list-style-type: none"><li>• SUBMITTED：已提交但尚未处理</li><li>• SCHEDULED：计划在未来执行的</li><li>• RUNNING：当前正在处理</li><li>• FAILED：处于失败状态</li><li>• SUCCEEDED：处于成功状态</li><li>• CANCELLED：已取消</li><li>• PARTIALLY_FAILED：部分失败</li><li>• ROLLBACKING：正在回滚</li><li>• ROLLBACKED：已回滚</li><li>• UNKNOWN：未知状态</li></ul>
JOB_DESC	String	工作描述
JOB_ID	String	工作ID
JOB_NAME	String	工作名称
JOB_TYPE	String	类别： <ul style="list-style-type: none"><li>• PROVISION：提供服务任务</li><li>• DEPROVISION：取消提交服务任务</li><li>• DAILY_OPERATION：日常操作任务</li><li>• SYSTEM_INNER：系统内部任务</li><li>• UNKNOWN：未知任务</li></ul>
ORDER_ID	String	排序ID
PROJECT_ID	String	创建租户Project ID

参数	参数类型	描述
SERVICE_INSTANCE_ID	String	实例ID

表 4-190 task 参数

参数	参数类型	描述
CREATED_AT	String	创建时间
LAST_HEALTH_CHECK	String	健康检查时间
MESSAGES	String	消息
OWNER_ID	String	创建用户ID
TASK_ID	String	任务ID
TASK_INDEX	Integer	任务序号
TASK_NAME	String	任务名称
TASK_STATUS	String	任务状态： <ul style="list-style-type: none"><li>• SUBMITTED: 已提交但尚未处理</li><li>• SCHEDULED: 计划在未来执行的</li><li>• RUNNING: 当前正在处理</li><li>• FAILED: 处于失败状态</li><li>• SUCCEEDED: 处于成功状态</li><li>• CANCELLED: 已取消</li><li>• PARTIALLY_FAILED: 部分失败</li><li>• SKIPPED: 已跳过</li><li>• ROLLBACKING: 正在回滚</li><li>• ROLLBACKED: 已回滚</li><li>• UNKNOWN: 未知状态</li></ul>
TASK_TYPE	String	任务类型： <ul style="list-style-type: none"><li>• kube</li><li>• restful</li><li>• sleep</li><li>• echo</li><li>• context</li><li>• sub-job</li></ul>

## 示例

### 请求示例

无

### 响应示例

```
{
  "job": {
    "JOB_ID": "JOB474362ef-091d-429f-9b3e-c9246b37dee6",
    "JOB_NAME": "provision/provMain",
    "JOB_TYPE": "PROVISION",
    "SERVICE_INSTANCE_ID": "429a35a0-dee8-4648-86f6-d1f8bc74ba52",
    "PROJECT_ID": "063a3158b1a34710b36ad208b9497d00",
    "ORDER_ID": "",
    "JOB_DESC": "provision/provMain",
    "EXECUTION_STATUS": "FAILED",
    "CREATED_BY": "585cc5c16cda4954b28d4eb460f06261"
  },
  "tasks": [
    {
      "MESSAGES": "{}",
      "TASK_ID": "Task-bc64b46c-1688-4787-b7ff-87cb53b346ea",
      "TASK_TYPE": "echo",
      "TASK_INDEX": 1,
      "TASK_NAME": "SEPARATOR_PREPARE_RESOURCES",
      "TASK_STATUS": "SUCCEEDED",
      "LAST_HEALTH_CHECK": "2020-02-04T04:01:03.344Z",
      "CREATED_AT": "2020-02-04T04:01:03.344Z",
      "OWNER_ID": "cas-mgr-69bb87d7c-wpqwb:30114"
    },
    {
      "MESSAGES": "{}",
      "TASK_ID": "Task-15ec252f-bca0-401e-ae5b-798a639df92e",
      "TASK_TYPE": "echo",
      "TASK_INDEX": 4,
      "TASK_NAME": "SEPARATOR_BUILDING",
      "TASK_STATUS": "SUCCEEDED",
      "LAST_HEALTH_CHECK": "2020-02-04T04:01:03.346Z",
      "CREATED_AT": "2020-02-04T04:01:03.345Z",
      "OWNER_ID": "cas-mgr-69bb87d7c-wpqwb:30114"
    },
    {
      "MESSAGES": "{}",
      "TASK_ID": "Task-52a5d1ae-2c54-46ec-8134-53e7b6a45fb7",
      "TASK_TYPE": "restful",
      "TASK_INDEX": 6,
      "TASK_NAME": "TN00091",
      "TASK_STATUS": "SUCCEEDED",
      "LAST_HEALTH_CHECK": "2020-02-04T04:01:08.924Z",
      "CREATED_AT": "2020-02-04T04:01:03.346Z",
      "OWNER_ID": "cas-mgr-69bb87d7c-wpqwb:30114"
    },
    {
      "MESSAGES": "{}",
      "TASK_ID": "Task-a149b585-d31b-4c72-a665-d820efcb0bf9",
      "TASK_TYPE": "restful",
      "TASK_INDEX": 7,
      "TASK_NAME": "TN00094",
      "TASK_STATUS": "SUCCEEDED",
      "LAST_HEALTH_CHECK": "2020-02-04T04:01:08.961Z",
      "CREATED_AT": "2020-02-04T04:01:08.925Z",
      "OWNER_ID": "cas-mgr-69bb87d7c-wpqwb:30114"
    },
    {
      "MESSAGES": "{
        \"Message\": \"ECC00011\",
        \"Detail\": \"Building failed.\"
      }"
    }
  ]
}
```



```
    },  
    "TASK_ID": "Task-a72fe8d8-8cc1-42b5-a097-0cf9e29692a7",  
    "TASK_TYPE": "restful",  
    "TASK_INDEX": 9,  
    "TASK_NAME": "TN00097",  
    "TASK_STATUS": "FAILED",  
    "LAST_HEALTH_CHECK": "2020-02-04T04:01:38.975Z",  
    "CREATED_AT": "2020-02-04T04:01:08.962Z",  
    "OWNER_ID": "cas-mgr-69bb87d7c-wpqwb:30114"  
  }  
],  
"task_count": 5  
}
```

## 状态码

表 4-191 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。
404	请求对象不存在
500	内部错误

## 错误码

错误码格式为：SVCSTG.00100.[Error\_ID]，例如：SVCSTG.00100400。错误码说明请参考[应用管理API错误码](#)。

# 5 访问 Git 仓库 API

## 5.1 获取仓库授权列表

### 功能介绍

获取Git仓库所有授权信息。

### URI

GET /v1/{project\_id}/git/auths

参数说明见[表5-1](#)。

表 5-1 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。

### 请求消息

#### 请求参数

无

### 响应消息

#### 响应参数

表 5-2 响应参数说明

参数	类型	说明
authorizations	Array<Object>	授权结构体，请参考 <a href="#">表5-3</a> 。

表 5-3 authorizations 参数说明

参数	类型	说明
name	String	授权名称。
repo_type	String	仓库类型。 取值范围：github、gitlab、gitee、bitbucket。
repo_host	String	仓库地址。
repo_home	String	仓库主页。
repo_user	String	仓库用户名。
avatar	String	头像。
token_type	String	授权方式。
create_time	Long	创建时间。
update_time	Long	修改时间。
status	Integer	状态。

## 示例

### 请求示例

无

### 响应示例

```
{
  "authorizations": [
    {
      "name": "string",
      "repo_type": "string",
      "repo_host": "string",
      "repo_home": "string",
      "repo_user": "string",
      "avatar": "string",
      "token_type": "string",
      "create_time": 0,
      "update_time": 0,
      "status": 0
    }
  ]
}
```

## 状态码

表 5-4 状态码说明

状态码	描述
200	操作成功。

状态码	描述
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.2 获取授权重定向 URL

### 功能介绍

获取授权重定向URL。

### URI

GET /v1/{project\_id}/git/auths/{repo\_type}/redirect

参数说明见[表5-5](#)。

表 5-5 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
repo_type	String	是	仓库类型。 取值范围：github、gitlab、gitee、bitbucket。

表 5-6 查询参数

参数	类型	必选	说明
tag	String	否	站点标签。

### 请求消息

请求参数

无

### 响应消息

响应参数

表 5-7 响应参数说明

参数	类型	说明
url	String	授权重定向URL。

## 示例

### 请求示例

无

### 响应示例

```
{  
  "url": "string"  
}
```

## 状态码

表 5-8 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.3 创建 OAuth 授权

### 功能介绍

创建OAuth授权。

### URI

POST /v1/{project\_id}/git/auths/{repo\_type}/oauth

参数说明见[表5-9](#)。

表 5-9 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。

参数	类型	必选	说明
repo_type	String	是	仓库类型。 取值范围：github、gitlab、gitee、bitbucket。

表 5-10 查询参数

参数	类型	必选	说明
tag	String	否	站点标签。

## 请求消息

### 请求参数

表 5-11 请求参数说明

参数	类型	必选	说明
name	String	是	授权名称。
code	String	是	git仓库授权后，重定向回来的url里面的query参数。
state	String	是	git仓库授权后，一次性的认证编码和随机串。

## 📖 说明

code和state参数值获取方法:

1. 在浏览器地址栏中输入访问[获取授权重定向URL](#)接口获取到的URL并回车。
2. 在登录页面输入您的代码仓库帐号和密码。

如果您已经登录代码仓库，则无需执行这一步。

3. 在云服务控制台登录页面输入帐号和密码，单击“登录”。

如果您已经登录云服务控制台，则无需执行这一步。

4. 等待页面显示新建授权失败后，复制地址栏的URL。例如：

```
https://console****.com/servicestage/oauth?  
repo_type=github&code=04bed2df009f3fc9***c&state=%242a  
%2410%24.uQYThj5yKFs51w9s9ajeuQ0NSGshPH7IUfXoZLPlz7J%2FoAJ8B***
```

5. 对获取到的URL进行URL解码，获得解码后的URL。示例如下：

```
https://console.****.com/servicestage/oauth?  
repo_type=github&code=04bed2df009f3fc9***c&state=$2a  
$10$.uQYThj5yKFs51w9s9ajeuQ0NSGshPH7IUfXoZLPlz7J/oAJ8B***
```

获取其中code和state对应的参数值。

获取到的state参数值为一次性的认证编码和随机串，仅支持使用一次。如需再次调用本接口创建OAuth授权，请再次执行[获取授权重定向URL](#)获取URL并参考本方法获取code和state参数值。

## 响应消息

### 响应参数

表 5-12 响应参数说明

参数	类型	说明
authorization	Object	授权结构体，请参考 <a href="#">表5-13</a> 。

表 5-13 authorization 参数说明

参数	类型	说明
name	String	授权名称。
repo_type	String	仓库类型。 取值范围：github、gitlab、gitee、bitbucket。
repo_host	String	仓库地址。
repo_home	String	仓库主页。
repo_user	String	仓库用户名。
avatar	String	头像。
token_type	String	授权方式。
create_time	Long	创建时间。

参数	类型	说明
update_time	Long	修改时间。
status	Integer	状态。

## 示例

### 请求示例

```
{
  "name": "string",
  "code": "string",
  "state": "string"
}
```

### 响应示例

```
{
  "authorization": {
    "name": "string",
    "repo_type": "string",
    "repo_host": "string",
    "repo_home": "string",
    "repo_user": "string",
    "avatar": "string",
    "token_type": "string",
    "create_time": 0,
    "update_time": 0,
    "status": 0
  }
}
```

## 状态码

表 5-14 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.4 创建私人令牌授权

### 功能介绍

创建私有令牌授权。



## URI

POST /v1/{project\_id}/git/auths/{repo\_type}/personal

参数说明见[表5-15](#)。

表 5-15 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
repo_type	String	是	仓库类型。 取值范围：github、gitlab、gitee。

## 请求消息

### 请求参数

表 5-16 请求参数说明

参数	类型	必选	说明
name	String	是	授权名称。
token	String	是	仓库Token。
host	String	否	仓库地址。

## 响应消息

### 响应参数

表 5-17 响应参数说明

参数	类型	说明
authorization	Object	授权结构体，请参考 <a href="#">表5-18</a> 。

表 5-18 authorization 参数说明

参数	类型	说明
name	String	授权名称。
repo_type	String	仓库类型。 取值范围：github、gitlab、gitee。

参数	类型	说明
repo_host	String	仓库地址。
repo_home	String	仓库主页。
repo_user	String	仓库用户名。
avatar	String	头像。
token_type	String	授权方式。
create_time	Long	创建时间。
update_time	Long	修改时间。
status	Integer	状态。

## 示例

### 请求示例

```
{  
  "name": "string",  
  "token": "string",  
  "host": "string"  
}
```

### 响应示例

```
{  
  "authorization": {  
    "name": "string",  
    "repo_type": "string",  
    "repo_host": "string",  
    "repo_home": "string",  
    "repo_user": "string",  
    "avatar": "string",  
    "token_type": "string",  
    "create_time": 0,  
    "update_time": 0,  
    "status": 0  
  }  
}
```

## 状态码

表 5-19 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.5 创建口令授权

### 功能介绍

创建Git仓库口令授权。

### URI

POST /v1/{project\_id}/git/auths/{repo\_type}/password

参数说明见[表5-20](#)。

表 5-20 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
repo_type	String	是	仓库类型。 取值范围：github、devcloud、bitbucket。

### 请求消息

#### 请求参数

表 5-21 请求参数说明

参数	类型	必选	说明
name	String	是	授权名称。
user	String	是	仓库用户名。
password	String	是	仓库密码。

### 响应消息

#### 响应参数

表 5-22 响应参数说明

参数	类型	说明
authorization	Object	授权结构体，请参考 <a href="#">表5-23</a> 。

表 5-23 authorization 参数说明

参数	类型	说明
name	String	授权名称。
repo_type	String	仓库类型。 取值范围：github、devcloud、bitbucket。
repo_host	String	仓库地址。
repo_home	String	仓库主页。
repo_user	String	仓库用户名。
avatar	String	头像。
token_type	String	授权方式。
create_time	Long	创建时间。
update_time	Long	修改时间。
status	Integer	状态。

## 示例

### 请求示例

```
{
  "name": "string",
  "user": "string",
  "password": "string"
}
```

### 响应示例

```
{
  "authorization": {
    "name": "string",
    "repo_type": "string",
    "repo_host": "string",
    "repo_home": "string",
    "repo_user": "string",
    "avatar": "string",
    "token_type": "string",
    "create_time": 0,
    "update_time": 0,
    "status": 0
  }
}
```

## 状态码

表 5-24 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.6 删除仓库授权

### 功能介绍

通过名称删除仓库授权。

### URI

DELETE /v1/{project\_id}/git/auths/{name}

参数说明见[表5-25](#)。

表 5-25 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
name	String	是	授权名称。

### 请求消息

#### 请求参数

无

### 响应消息

#### 响应参数

无

### 示例

#### 请求示例

无

响应示例

无

## 状态码

表 5-26 状态码说明

状态码	描述
204	操作成功。
500	内部服务错误。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.7 获取仓库 namespaces

### 功能介绍

获取仓库namespaces。

### URI

GET /v1/{project\_id}/git/repos/namespaces

参数说明见[表5-27](#)。

表 5-27 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。

### 请求消息

请求参数

无

### 响应消息

响应参数

表 5-28 响应参数说明

参数	类型	说明
namespaces	Array<Object>	命名空间结构体，请参考 <a href="#">表5-29</a> 。

表 5-29 namespaces 参数说明

参数	类型	说明
id	String	命名空间ID。
name	String	命名空间名称。

## 示例

### 请求示例

无

### 响应示例

```
{
  "namespaces": [
    {
      "id": "string",
      "name": "string"
    }
  ]
}
```

## 状态码

表 5-30 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.8 通过 clone url 获取仓库信息

### 功能介绍

通过clone url 获取仓库信息

### URI

GET /v1/{project\_id}/git/repos/project-info

参数说明见[表5-31](#)。

表 5-31 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。

表 5-32 查询参数

参数	类型	必选	说明
clone_url	String	是	仓库克隆URL。

### 请求消息

请求参数

无

### 响应消息

响应参数

表 5-33 响应参数说明

参数	类型	说明
namespace_id	String	命名空间ID。
namespace	String	命名空间。
project_id	String	仓库项目 ID。
project	String	仓库项目。



## 示例

### 请求示例

```
https://{Endpoint}/v1/7c6a62ddb2bf45a3920d8053d3cb8fbf/git/repos/project-info?clone_url=http://xxx.gitlab.com:8090/xxx/springcloud-xxbbon.git
```

### 响应示例

```
{
  "project": "string"
  "namespace_id": "string",
  "namespace": "string",
  "project_id": "string",
}
```

## 状态码

表 5-34 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.9 获取命名空间下所有项目

### 功能介绍

获取命名空间下所有项目

### URI

GET /v1/{project\_id}/git/repos/{namespace}/projects

参数说明见[表5-35](#)。

表 5-35 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID。

## 请求消息

### 请求参数

无

## 响应消息

### 响应参数

表 5-36 响应参数说明

参数	类型	说明
projects	Array<Object>	项目结构体，请参考 <a href="#">表5-37</a> 。

表 5-37 projects 参数说明

参数	类型	说明
id	String	项目ID。
name	String	项目名称。
clone_url	String	克隆URL。

## 示例

### 请求示例

无

### 响应示例

```
{
  "projects": [
    {
      "id": "string",
      "name": "string",
      "clone_url": "string"
    }
  ]
}
```

## 状态码

表 5-38 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.10 创建软件仓库项目

### 功能介绍

创建软件仓库项目

### URI

POST /v1/{project\_id}/git/repos/{namespace}/projects

参数说明见[表5-39](#)。

表 5-39 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID。

### 请求消息

请求参数

表 5-40 请求参数说明

参数	类型	必选	说明
name	String	是	项目名称。

### 响应消息

响应参数

表 5-41 响应参数说明

参数	类型	说明
id	String	项目ID。
name	String	项目名称。
clone_url	String	克隆URL。

## 示例

### 请求示例

无

### 响应示例

```
{
  "id": "string",
  "name": "string",
  "clone_url": "string"
}
```

## 状态码

表 5-42 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.11 获取项目分支

### 功能介绍

获取项目分支

### URI

GET /v1/{project\_id}/git/repos/{namespace}/{project}/branches

参数说明见[表5-43](#)。

表 5-43 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID。

参数	类型	必选	说明
project	String	是	仓库项目ID, 如果含有“/”, 需要将“/”替换为“:”。

## 请求消息

### 请求参数

无

## 响应消息

### 响应参数

表 5-44 响应参数说明

参数	类型	说明
branches	Array<String>	项目分支。

## 示例

### 请求示例

无

### 响应示例

```
{  
  "branches": [  
    "string"  
  ]  
}
```

## 状态码

表 5-45 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为: SVCSTG.REPO.[Error\_ID], 例如: SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.12 获取项目 tags

### 功能介绍

获取项目tags。

### URI

GET /v1/{project\_id}/git/repos/{namespace}/{project}/tags

参数说明见[表5-46](#)。

表 5-46 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。

### 请求消息

请求参数

无

### 响应消息

响应参数

表 5-47 响应参数说明

参数	类型	说明
tags	Array<String>	项目tags。

### 示例

请求示例

无

响应示例

```
{  
  "tags": [  

```

```
"string"  
]  
}
```

## 状态码

表 5-48 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage 错误码](#)。

## 5.13 创建项目 tags

### 功能介绍

创建项目tags

### URI

POST /v1/{project\_id}/git/repos/{namespace}/{project}/tags

参数说明见[表5-49](#)。

表 5-49 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。

表 5-50 查询参数

参数	类型	必选	说明
ref	String	是	分支名称或者tag名称或者commit sha。

## 请求消息

### 请求参数

表 5-51 请求参数说明

参数	类型	必选	说明
name	String	是	tags名称。
description	String	是	描述。

## 响应消息

### 响应参数

表 5-52 响应参数说明

参数	类型	说明
name	String	项目tags名称。

## 示例

### 请求示例

```
{
  "name": "string",
  "description": "string"
}
```

### 响应示例

```
{
  "name": "string"
}
```

## 状态码

表 5-53 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。



## 5.14 删除项目 tags

### 功能介绍

删除项目tags

### URI

DELETE /v1/{project\_id}/git/repos/{namespace}/{project}/tags/{tag\_name}

参数说明见[表5-54](#)。

表 5-54 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。
tag_name	String	是	tag名称。

### 请求消息

请求参数

无

### 响应消息

响应参数

无

### 示例

请求示例

无

响应示例

无

## 状态码

表 5-55 状态码说明

状态码	描述
204	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.15 获取项目 commits

### 功能介绍

获取项目最近10次commits。

### URI

GET /v1/{project\_id}/git/repos/{namespace}/{project}/commits

参数说明见[表5-56](#)。

表 5-56 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。

表 5-57 查询参数

参数	类型	必选	说明
ref	String	否	分支名称或者tag名称，如果没有提供，使用默认分支。

## 请求消息

### 请求参数

无

## 响应消息

### 响应参数

表 5-58 响应参数说明

参数	类型	说明
commits	Array<Object>	项目commits结构体，请参考 <a href="#">表5-59</a> 。

表 5-59 commits 参数说明

参数	类型	说明
sha	String	commit哈希。
message	String	commit描述。
authored_date	String	合入时间。

## 示例

### 请求示例

无

### 响应示例

```
{
  "commits": [
    {
      "sha": "string",
      "message": "string",
      "authored_date": "string"
    }
  ]
}
```

## 状态码

表 5-60 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.16 获取项目 hooks

### 功能介绍

获取项目hooks。

### URI

GET /v1/{project\_id}/git/repos/{namespace}/{project}/hooks

参数说明见[表5-61](#)。

表 5-61 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。

### 请求消息

#### 请求参数

无

### 响应消息

#### 响应参数

表 5-62 响应参数说明

参数	类型	说明
hooks	Array of objects	项目hooks，请参考 <a href="#">表5-63</a> 。

表 5-63 hooks 参数说明

参数	类型	说明
id	String	hook ID。
type	String	hook类型。
callback_url	String	回调url。

## 示例

### 请求示例

无

### 响应示例

```
{
  "hooks": [
    {
      "id": "string",
      "type": "string",
      "callback_url": "string"
    }
  ]
}
```

## 状态码

表 5-64 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.17 创建项目 hook

### 功能介绍

创建项目hook。

### URI

POST /v1/{project\_id}/git/repos/{namespace}/{project}/hooks

参数说明见表5-65。

表 5-65 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。

## 请求消息

### 请求参数

表 5-66 请求参数说明

参数	类型	必选	说明
secret	String	是	用于验证接收到的payloads的字符串。
url	String	是	hook触发时回调的url。

## 响应消息

### 响应参数

表 5-67 响应参数说明

参数	类型	说明
id	String	hook ID。
type	String	hook类型。
callback_url	String	回调url。

## 示例

### 请求示例

```
{  
  "secret": "string",  
  "url": "string"  
}
```

### 响应示例

```
{
  "id": "string",
  "type": "string",
  "callback_url": "string"
}
```

## 状态码

表 5-68 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.18 删除项目 hook

### 功能介绍

删除项目hook。

### URI

DELETE /v1/{project\_id}/git/repos/{namespace}/{project}/hooks/{hook\_id}

参数说明见[表5-69](#)。

表 5-69 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。
hook_id	String	是	hook ID。

### 请求消息

#### 请求参数

无

## 响应消息

### 响应参数

无

## 示例

### 请求示例

无

### 响应示例

无

## 状态码

表 5-70 状态码说明

状态码	描述
204	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.19 获取仓库文件目录

### 功能介绍

获取仓库文件目录。

### URI

GET /v1/{project\_id}/git/files/{namespace}/{project}/trees

参数说明见[表5-71](#)。

表 5-71 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。



参数	类型	必选	说明
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。

表 5-72 查询参数

参数	类型	必选	说明
ref	String	是	分支名称或者tag名称或者commit sha。

## 请求消息

### 请求参数

无

## 响应消息

### 响应参数

表 5-73 响应参数说明

参数	类型	说明
paths	Array<String>	文件路径。

## 示例

### 请求示例

无

### 响应示例

```
{
  "paths": [
    "string"
  ]
}
```

## 状态码

表 5-74 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.20 获取仓库文件内容

### 功能介绍

获取仓库文件内容。

### URI

GET /v1/{project\_id}/git/files/{namespace}/{project}/{path}

参数说明见[表5-75](#)。

表 5-75 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。
path	String	是	文件路径，需要将“/”替换为“.”。

表 5-76 查询参数

参数	类型	必选	说明
ref	String	是	分支名称或者tag名称或者commit sha。

## 请求消息

### 请求参数

无

## 响应消息

### 响应参数

表 5-77 响应参数说明

参数	类型	说明
path	String	文件路径。
sha	String	commit哈希。
encoding	String	编码方式：base64或者text/plain。
content	String	文件内容。

## 示例

### 请求示例

无

### 响应示例

```
{  
  "path": "string",  
  "sha": "string",  
  "encoding": "base64",  
  "content": "string"  
}
```

## 状态码

表 5-78 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.21 创建仓库文件

### 功能介绍

创建仓库文件。

### URI

POST /v1/{project\_id}/git/files/{namespace}/{project}/{path}

参数说明见[表5-79](#)。

表 5-79 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。
path	String	是	文件路径，需要将“/”替换为“.”。

表 5-80 查询参数

参数	类型	必选	说明
ref	String	是	分支名称或者tag名称或者commit哈希。

### 请求消息

#### 请求参数

表 5-81 请求参数说明

参数	类型	必选	说明
message	String	是	描述。
content	String	是	文件内容。

## 响应消息

### 响应参数

表 5-82 响应参数说明

参数	类型	说明
path	String	文件路径。

## 示例

### 请求示例

```
{
  "message": "string",
  "content": "string"
}
```

### 响应示例

```
{
  "path": "string"
}
```

## 状态码

表 5-83 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.22 修改仓库文件内容

### 功能介绍

修改仓库文件内容。

### URI

PUT /v1/{project\_id}/git/files/{namespace}/{project}/{path}

参数说明见[表5-84](#)。

表 5-84 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。
path	String	是	文件路径，需要将“/”替换为“.”。

表 5-85 查询参数

参数	类型	必选	说明
ref	String	是	分支名称或者tag名称或者commit哈希。

## 请求消息

### 请求参数

表 5-86 请求参数说明

参数	类型	必选	说明
message	String	是	描述。
content	String	是	文件内容。
sha	String	是	commit哈希。

## 响应消息

### 响应参数

表 5-87 响应参数说明

参数	类型	说明
path	String	文件路径。

## 示例

### 请求示例

```
{
  "message": "string",
  "content": "string",
  "sha": "string"
}
```

### 响应示例

```
{
  "path": "string"
}
```

## 状态码

表 5-88 状态码说明

状态码	描述
200	操作成功。
400	错误的请求。

## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

## 5.23 删除仓库文件

### 功能介绍

删除仓库文件内容。

### URI

DELETE /v1/{project\_id}/git/files/{namespace}/{project}/{path}

参数说明见[表5-89](#)。

表 5-89 路径参数

参数	类型	必选	说明
project_id	String	是	租户项目ID。
namespace	String	是	命名空间ID或者URL编码名称。

参数	类型	必选	说明
project	String	是	仓库项目ID或者URL编码名称，如果含有“/”，需要将“/”替换为“.”。
path	String	是	文件路径，需要将“/”替换为“.”。

表 5-90 查询参数

参数	类型	必选	说明
ref	String	是	分支名称或者tag名称或者commit哈希。
message	String	是	commit信息。
sha	String	是	最近commit哈希。

## 请求消息

### 请求参数

无

## 响应消息

### 响应参数

无

## 示例

### 请求示例

无

### 响应示例

无

## 状态码

表 5-91 状态码说明

状态码	描述
204	操作成功。
400	错误的请求。



## 错误码

错误码格式为：SVCSTG.REPO.[Error\_ID]，例如：SVCSTG.REPO.0401。错误码说明请参考[ServiceStage错误码](#)。

# 6 CSE API

## 6.1 调用说明

ServiceStage提供了REST ( Representational State Transfer ) 风格API，支持您通过HTTPS请求调用，访问微服务引擎专享版的CSE API方法参见[获取服务注册发现地址](#)。

## 6.2 查询指定微服务静态信息

### 功能介绍

根据serviceld查询微服务静态信息。

### URI

GET /v4/{project\_id}/registry/microservices/{serviceld}

表 6-1 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值: default。
serviceld	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 $\wedge.*\$$

### 请求消息

无

### 响应消息

响应参数

表 6-2 参数说明

参数	类型	必选	说明
service	Object	是	微服务结构体，请参考 <a href="#">MicroService</a> 。

## 示例

### 请求示例

```
GET /v4/default/registry/microservices/819706e21b7173306797d19922ce4231441c17c5
```

### 响应示例

```
{
  "service": {
    "serviceId": "819706e21b7173306797d19922ce4231441c17c5",
    "appId": "default",
    "serviceName": "SERVICECENTER",
    "version": "2.4.8",
    "level": "BACK",
    "schemas": [
      "servicecenter.grpc.api.ServiceCtrl",
      "servicecenter.grpc.api.ServiceInstanceCtrl"
    ],
    "status": "UP",
    "properties": {
      "allowCrossApp": "true"
    },
    "timestamp": "1616426688",
    "alias": "SERVICECENTER",
    "modTimestamp": "1616426688",
    "environment": "development"
  }
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.3 查询所有微服务静态信息

### 功能介绍

根据条件组合，查询满足所有条件的微服务静态信息。

### URI

```
GET /v4/{project_id}/registry/microservices
```

表 6-3 路径参数

参数	类型	必选	说明
project_id	String	是	请填写固定值：default。

## 请求消息

无

## 响应消息

### 响应参数

参数说明请参见[表6-4](#)。

表 6-4 参数说明

参数	类型	说明
services	Array of Objects	微服务列表。详情参考 <a href="#">MicroService</a> 。

## 示例

### 请求示例

```
GET /v4/default/registry/microservices
```

### 响应示例

```
{
  "services": [
    {
      "serviceId": "string",
      "environment": "string",
      "appld": "string",
      "serviceName": "string",
      "version": "string",
      "description": "string",
      "level": "string",
      "registerBy": "string",
      "schemas": [
        "string"
      ],
      "status": "UP",
      "timestamp": "string",
      "modTimestamp": "string",
      "framework": {
        "name": "string",
        "version": "string"
      },
      "paths": [
        {
          "Path": "string",
          "Property": {
            "additionalProp1": "string",
            "additionalProp2": "string",

```

```
        "additionalProp3": "string"
      }
    ]
  }
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

# 6.4 创建微服务静态信息

## 功能介绍

在注册微服务实例前需要创建服务静态信息，之后注册的微服务实例根据ServiceId这个字段与静态信息关联，一个服务对应多个实例。

ServiceId可以定制。如果定制了，系统将使用定制的ServiceId；如果没定制，系统会随机生成ServiceId。

## URI

POST /v4/{project\_id}/registry/microservices

表 6-5 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。

## 请求消息

表 6-6 请求 Body 参数

参数	类型	必选	说明
service	Object	是	微服务结构体，具体参考 <a href="#">MicroService</a> 。
rules	Array of objects	否	新增黑白名单结构体，具体参考 <a href="#">Rule</a> 。
instances	Array of objects	否	微服务实例结构体，具体参考 <a href="#">MicroServiceInstance</a> 。

参数	类型	必选	说明
tags	Map<String, String>	否	微服务标签，可以自定义Key和相应的Value。

## 响应消息

### 响应参数

参数说明请参见表6-7。

表 6-7 参数说明

参数	类型	说明
serviceld	String	微服务唯一标识。

## 示例

### 请求示例

```
{
  "service": {
    "serviceld": "string",
    "environment": "string",
    "appld": "string",
    "serviceName": "string",
    "version": "string",
    "description": "string",
    "level": "string",
    "registerBy": "string",
    "schemas": [
      "string"
    ],
    "status": "UP",
    "timestamp": "string",
    "modTimestamp": "string",
    "framework": {
      "name": "string",
      "version": "string"
    },
    "paths": [
      {
        "Path": "string",
        "Property": {
          "additionalProp1": "string",
          "additionalProp2": "string",
          "additionalProp3": "string"
        }
      }
    ]
  },
  "rules": [
    {
      "ruleId": "string",
      "ruleType": "string",
      "attribute": "string",
      "pattern": "string",

```

```
"description": "string",
"timestamp": "string",
"modTimestamp": "string"
}
],
"instances": [
{
"instanceId": "string",
"serviceId": "string",
"version": "string",
"hostName": "string",
"endpoints": [
"string"
],
"status": "string",
"healthCheck": {
"mode": "string",
"port": 0,
"interval": 0,
"times": 0
},
"dataCenterInfo": {
"name": "string",
"region": "string",
"availableZone": "string"
},
"timestamp": "string",
"modTimestamp": "string"
}
],
"tags": {
"additionalProp1": "string",
"additionalProp2": "string",
"additionalProp3": "string"
}
}
```

### 响应示例

```
{
"serviceId": "string"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.5 修改微服务静态信息

### 功能介绍

创建微服务静态信息后可对服务部分字段进行更新，每次更新需要传入微服务扩展属性的结构体。

### URI

PUT /v4/{project\_id}/registry/microservices/{serviceId}/properties

表 6-8 路径参数

参数	类型	必选	说明
project_id	String	是	请填写固定值：default。
serviceId	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 $^{\wedge}.*\$$

## 请求消息

### 请求参数

表 6-9 请求 Body 参数

参数	类型	必选	说明
properties	Map<String, String>	否	微服务扩展信息，可以自定义Key和相应的Value。

## 响应消息

### 响应参数

无

## 示例

### 请求示例

```
{
  "properties": {
    "attr1": "value1"
  }
}
```

### 响应示例

无

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。



## 6.6 查询微服务访问契约

### 功能介绍

根据指定的serviceld和schemald查询微服务的schema信息。

### URI

GET /v4/{project\_id}/registry/microservices/{serviceld}/schemas/{schemald}

表 6-10 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。
serviceld	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为^.*\$
schemald	String	是	schema的id。

### 请求消息

无

### 响应消息

#### 响应参数

参数说明请参见[表6-11](#)。

表 6-11 参数说明

参数	类型	说明
schema	String	微服务契约内容。

### 示例

#### 请求示例

```
GET /v4/default/registry/microservices/e0f0da073f2c91e8979a89ff2d7c69t6/schemas/com.test.TestService
```

#### 响应示例

```
{"schema": "---\nswagger: \"2.0\"\ninfo:\n  version: \"1.0.0\"\n  title: \"swagger definition for\ncom.service.provider.controller.ProviderImpl\"\n  x-java-interface:\n    \"cse.gen.springmvc.provider.provider.ProviderImplIntf\"\n  basePath: \"/provider\"\n  consumes:\n    \"application/json\"\n  produces:\n    \"application/json\"\n  paths:\n    /helloworld:\n      get:\n        operationId:
```

```
\\"helloworld\\"\\n  produces:\\n  - \\"application/json\\"\\n  parameters:\\n  - name: \\"name\\"\\n  in: \\"query\\"\\n  required: true\\n  type: \\"string\\"\\n  responses:\\n  200:\\n  description: \\"response of 200\\"\\n  schema:\\n  type: \\"string\\"\\n"}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

# 6.7 修改微服务访问契约

## 功能介绍

根据schemald更新微服务的访问契约内容。微服务引擎专业版默认不支持修改契约，微服务引擎专享版默认支持修改契约。

## URI

PUT /v4/{project\_id}/registry/microservices/{serviceId}/schemas/{schemald}

表 6-12 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。
serviceId	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 <code>^.*\$</code>
schemald	String	是	微服务契约唯一标识。 字符长度为1~160。 正则表达式为 <code>^[a-zA-Z0-9]{1,160}\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]{0,158}[a-zA-Z0-9]\$</code>

## 请求消息

表 6-13 请求 Body 参数

参数	类型	必选	说明
schema	String	是	微服务契约内容。
summary	String	否	微服务契约摘要。

## 响应消息

### 响应参数

无

## 示例

### 请求示例

```
{  
  "schema": "string",  
  "summary": "string"  
}
```

### 响应示例

无

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

# 6.8 创建服务间依赖关系

## 功能介绍

创建服务间的依赖关系，consumer的version必须是确定的版本，consumer的serviceName不能为\*，consumer必须是已存在的服务。provider可以是还未创建的。

## URI

PUT /v4/{project\_id}/registry/dependencies

表 6-14 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。

## 请求消息

表 6-15 请求 Body 参数

参数	类型	必选	说明
dependencies	Array of objects	是	规则的集合，请参考 <a href="#">表6-16</a> 。列表长度为1-100。

表 6-16 MicroServiceDependency 参数说明

参数	类型	必选	说明
consumer	Object	是	消费端微服务，具体参考 <a href="#">DependencyMicroService</a> 。
providers	Array of objects	否	服务端微服务，具体参考 <a href="#">DependencyMicroService</a> 。

## 响应消息

### 响应参数

无

## 示例

### 请求示例

```
{
  "dependencies": [
    {
      "consumer": {
        "environment": "string",
        "appld": "string",
        "serviceName": "string",
        "version": "string"
      },
      "providers": [
        {
          "environment": "string",
          "appld": "string",
          "serviceName": "string",
          "version": "string"
        }
      ]
    }
  ]
}
```

### 响应示例

无

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.9 查询服务 providers

### 功能介绍

根据serviceld获取该服务的所有providers。

### URI

GET /v4/{project\_id}/registry/microservices/{serviceld}/providers

表 6-17 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。
serviceld	String	是	消费者服务id。

### 请求消息

无

### 响应消息

#### 响应参数

参数说明请参见[表6-18](#)。

表 6-18 参数说明

参数	类型	说明
providers	Array of objects	微服务结构体，具体参考 <a href="#">MicroService</a> 。

### 示例

#### 请求示例

```
GET /v4/default/registry/microservices/e5788b23b640232730c7a99a6d806b4f31be453e/providers
```

#### 响应示例

```
{
  "providers": [{
    "serviceld": "string",
    "appld": "string",
    "serviceName": "string",
    "version": "string",
    "description": "string",
    "level": "string",
```

```
"status": "up",
"timestamp": "string",
"framework": {
  "name": "string",
  "version": "string"
}
}]
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

# 6.10 查询微服务唯一标识信息

## 功能介绍

可通过指定条件，查询微服务serviceId或schema的唯一标识信息。

## URI

GET /v4/{project\_id}/registry/existence

表 6-19 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。

表 6-20 Query 参数

参数	类型	必选	说明
type	String	是	请求类型： <ul style="list-style-type: none"><li>microservice</li><li>schema</li></ul>
env	String	否	微服务的环境信息，取值为development testing acceptance production，默认为development。
appld	String	是	资源类型为microservice时，需传入应用app唯一标识。 字符长度为1~160。 正则表达式为^[a-zA-Z0-9]*\$ ^([a-zA-Z0-9][a-zA-Z0-9_\-]*)[a-zA-Z0-9]\$

参数	类型	必选	说明
serviceName	String	是	资源类型为microservice时，需传入微服务名称。 字符长度为1~128 正则表达式为^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$
version	String	是	资源类型为microservice时，需传入微服务版本。 字符长度为1~64。 正则表达式为^[0-9]*\$ ^[0-9]+(\.[0-9]+)*\$
serviceId	String	是	资源类型为schema时，需传入微服务唯一标识。 字符长度为1~64。 正则表达式为^.*\$
schemald	String	是	资源类型为schema时，需传入schema唯一标识。 字符长度为1~160。 正则表达式为^[a-zA-Z0-9]{1,160}\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]{0,158}[a-zA-Z0-9]\$

## 请求消息

无

## 响应消息

### 响应参数

参数说明请参见[表6-21](#)。

表 6-21 参数说明

参数	类型	说明
serviceId	String	微服务唯一标识，当查询微服务时返回该字段。
schemald	String	访问契约唯一标识，当查询契约时返回该字段。

## 示例

### 请求示例

```
GET /v4/default/registry/existence
```

### 响应示例

```
{  
  "serviceId": "string"  
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

# 6.11 注册微服务实例

## 功能介绍

创建微服务静态信息后可以注册该微服务的实例。注册微服务实例时，需提供该微服务实例相关的信息。

instanceID可定制，如果定制，则使用定制的instanceID注册实例。如果没定制，系统自动生成id。

## URI

POST /v4/{project\_id}/registry/microservices/{serviceId}/instances

表 6-22 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。
serviceId	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 $^{\wedge}.*^{\$}$

## 请求消息

### 请求参数

表 6-23 参数说明

参数	类型	必选	说明
instance	Object	是	注册微服务实例结构体，具体参考 <a href="#">MicroServiceInstance</a> 。

## 响应消息

### 响应参数

参数说明请参见[表6-24](#)。



表 6-24 参数说明

参数	类型	说明
instanceId	String	微服务实例唯一标识。

## 示例

### 请求示例

```
{
  "instance": {
    "instanceId": "string",
    "serviceId": "string",
    "version": "string",
    "hostName": "string",
    "endpoints": [
      "string"
    ],
    "status": "string",
    "healthCheck": {
      "mode": "string",
      "port": 0,
      "interval": 0,
      "times": 0
    },
    "dataCenterInfo": {
      "name": "string",
      "region": "string",
      "availableZone": "string"
    },
    "timestamp": "string",
    "modTimestamp": "string"
  }
}
```

### 响应示例

```
{
  "instanceId": "string"
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.12 查询微服务实例（根据 serviceId）

### 功能介绍

实例注册后可以根据service\_id发现该微服务的所有实例。

### URI

GET /v4/{project\_id}/registry/microservices/{serviceId}/instances

表 6-25 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。
serviceId	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 $^.*\$$

## 请求消息

表 6-26 请求 Header 参数

参数	类型	必选	说明
X-ConsumerId	String	否	服务消费者唯一标识。

## 响应消息

### 响应参数

表 6-27 参数说明

参数	类型	说明
instances	Array of objects	微服务实例的集合，具体参考 <a href="#">MicroServiceInstance</a> 。

## 示例

### 请求示例

```
GET /v4/default/registry/microservices/e0f0da073f2c91e8979a89ff2d7c69t6/instances
```

### 响应示例

```
{
  "instances": [
    {
      "instanceId": "string",
      "serviceId": "string",
      "version": "string",
      "hostName": "string",
      "endpoints": [
        "string"
      ],
      "status": "string",
      "healthCheck": {
        "mode": "string",
```

```
"port": 0,
"interval": 0,
"times": 0
},
"dataCenterInfo": {
"name": "string",
"region": "string",
"availableZone": "string"
},
"timestamp": "string",
"modTimestamp": "string"
}
]
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.13 注销微服务实例

### 功能介绍

实例注册后可以根据instance\_id进行实例注销。

### URI

DELETE /v4/{project\_id}/registry/microservices/{serviceld}/instances/{instanceld}

表 6-28 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。
serviceld	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 $^{\wedge}.*\$$
instanceld	String	是	微服务实例唯一标识。 字符长度不超过64位。 正则表达式为 $^{\wedge}[A-Za-z0-9_.-]*\$$

### 请求消息

无

## 响应消息

### 响应参数

无

## 示例

### 请求示例

```
DELETE /v4/default/registry/microservices/a0c572b9ede946712698081a87552a7a87169bcb/instances/17dd265a8b6f11ebb23100ff14de2ec3
```

### 响应示例

无

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

# 6.14 查询微服务实例详细信息

## 功能介绍

实例注册后可以根据serviceid和instanceId获取该实例的详细信息。

## URI

```
GET /v4/{project_id}/registry/microservices/{serviceid}/instances/{instanceId}
```

表 6-29 路径参数

参数	类型	必选	说明
project_id	String	是	请填写固定值：default。
serviceid	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 <code>^.*\$</code>
instanceId	String	是	微服务实例唯一标识。 字符长度不超过64位。 正则表达式为 <code>^[A-Za-z0-9_-]*\$</code>

## 请求消息

### 请求参数

无

## 响应消息

### 响应参数

参数说明请参见[表6-30](#)。

表 6-30 参数说明

参数	类型	说明
instance	Object	微服务实例，具体参考 <a href="#">MicroServiceInstance</a> 。

## 示例

### 请求示例

```
GET /v4/default/registry/microservices/a0c572b9ede946712698081a87552a7a87169bcb/instances/2370d7938b7111ebb23100ff14de2ec3
```

### 响应示例

```
{
  "instance": {
    "instanceId": "string",
    "serviceId": "string",
    "version": "string",
    "hostName": "string",
    "endpoints": [
      "string"
    ],
    "status": "string",
    "healthCheck": {
      "mode": "string",
      "port": 0,
      "interval": 0,
      "times": 0
    },
    "dataCenterInfo": {
      "name": "string",
      "region": "string",
      "availableZone": "string"
    },
    "timestamp": "string",
    "modTimestamp": "string"
  }
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.15 修改微服务实例扩展信息

### 功能介绍

实例注册后可以根据instance\_id进行添加/更新一个微服务实例扩展信息。

### URI

PUT /v4/{project\_id}/registry/microservices/{serviceld}/instances/{instanceld}/properties

表 6-31 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。
serviceld	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为^.*\$
instanceld	String	是	微服务实例唯一标识。 字符长度不超过64位。 正则表达式为^[A-Za-z0-9_-.]*\$

### 请求消息

#### 请求参数

表 6-32 请求 Body 参数

参数	类型	必选	说明
properties	Map<String, String>	否	微服务实例扩展属性，可以自定义Key和相应的Value。

### 响应消息

#### 响应参数

无

### 示例

#### 请求示例

```
{  
  "properties": {
```

```
    "attr1": "value1"  
  }  
}
```

#### 响应示例

无

#### 状态码

状态码请参见[状态码](#)。

#### 错误码

错误码请参见[CSE错误码](#)。

## 6.16 修改微服务实例状态

#### 功能介绍

实例注册后可以根据instance\_id进行更新一个微服务实例状态。

#### URI

PUT /v4/{project\_id}/registry/microservices/{serviceId}/instances/{instanceId}/status

表 6-33 路径参数

参数	类型	必选	说明
project_id	String	是	请填写固定值：default。
serviceId	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 <code>^.*\$</code>
instanceId	String	是	微服务实例唯一标识。 字符长度不超过64位。 正则表达式为 <code>^[A-Za-z0-9_-]*\$</code>

表 6-34 Query 参数

参数	类型	必选	说明
value	String	是	实例状态。 <ul style="list-style-type: none"><li>• UP表示在线</li><li>• OUTFSERVICE表示摘机</li><li>• STARTING表示正在启动</li><li>• DOWN表示下线</li><li>• TESTING表示拨测</li></ul>

## 请求消息

### 请求参数

无

## 响应消息

### 响应参数

无

## 示例

### 请求示例

```
PUT /v4/default/registry/microservices/a0c572b9ede946712698081a87552a7a87169bcb/instances/831113358b7811ebb23100ff14de2ec3/status?value=UP
```

### 响应示例

无

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.17 发送心跳信息

### 功能介绍

服务提供端需要向服务中心发送心跳信息，以保证服务中心知道服务实例是否健康。



## URI

PUT /v4/{project\_id}/registry/microservices/{serviceld}/instances/{instanceId}/heartbeat

表 6-35 路径参数

参数	类型	必选	说明
project_id	String	是	请填写固定值：default。
serviceld	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 $^{\wedge}.*\$$
instanceId	String	是	微服务实例唯一标识。 字符长度不超过64位。 正则表达式为 $^{\wedge}[A-Za-z0-9_-\wedge]*\$$

## 请求消息

无

## 响应消息

### 响应参数

无

## 示例

### 请求示例

```
PUT /v4/default/registry/microservices/a0c572b9ede946712698081a87552a7a87169bcb/instances/2370d7938b7111ebb23100ff14de2ec3/heartbeat
```

### 响应示例

无

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.18 按条件查询微服务实例

### 功能介绍

实例注册后可以根据微服务版本规则或字段条件发现该微服务的实例。

### URI

GET /v4/{project\_id}/registry/instances

表 6-36 路径参数

参数	类型	必选	说明
project_id	String	是	请填写固定值：default。

表 6-37 Query 参数

参数	类型	必选	说明
appId	String	是	应用app唯一标识。 字符长度为1~160。 正则表达式为^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$
serviceName	String	是	微服务名称。 字符长度为1~128。 正则表达式为^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$
version	String	是	版本规则。包括： <ul style="list-style-type: none"><li>• 1 表示精确版本匹配，如：0.0.1</li><li>• 2 表示后续版本匹配，如：0.0.1+</li><li>• 3 表示最新版本，如：latest</li><li>• 4 表示版本范围，如：0.1.0-0.2.0</li></ul>
env	String	否	取值为development testing acceptance production。
tags	String	否	Tag标签过滤服务，多个时逗号分隔。 正则表达式为^[a-zA-Z][a-zA-Z0-9_\.]{0,63}\$

## 请求消息

表 6-38 请求 Header 参数

参数	类型	必选	说明
X-ConsumerId	String	否	服务消费者唯一标识。

## 响应消息

### 响应参数

参数说明请参见[表6-39](#)。

表 6-39 参数说明

参数	类型	说明
instances	Array of objects	微服务实例的集合。每一项为 <a href="#">MicroServiceInstance</a> 。

## 示例

### 请求示例

```
GET /v4/default/registry/instances
```

### 响应示例

```
{
  "instances": [
    {
      "instanceId": "string",
      "serviceId": "string",
      "version": "string",
      "hostName": "string",
      "endpoints": [
        "string"
      ],
      "status": "string",
      "healthCheck": {
        "mode": "string",
        "port": 0,
        "interval": 0,
        "times": 0
      },
      "dataCenterInfo": {
        "name": "string",
        "region": "string",
        "availableZone": "string"
      },
      "timestamp": "string",
      "modTimestamp": "string"
    }
  ]
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.19 查询配置

### 功能业务

从配置管理中心查询配置信息。

### URI

GET /v3/{project\_id}/configuration/items

表 6-40 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。

表 6-41 Query 参数

参数	类型	必选	说明
dimensionsInfo	String	是	服务信息，为服务名、应用、版本的拼接（服务名、应用必填，版本选填），拼接格式为（请注意“@”和“#”添加到http url时需转义，“@”转义为“%40”，“#”转义为“%23”）： {serviceName}@{appId}#{version} 或 {serviceName}@{appId} 不能包含\$%^&+/\特殊字符和空格。
keyFilter	String	否	key的筛选条件，支持正则表达式。涉及特殊字符的正则表达式需要自行转义。
revision	String	否	配置项版本。 若revision值与当前配置的revision一致，则返回304状态码，无body体返回；若不一致，则返回当前配置和revision。

### 请求消息

请求参数

表 6-42 请求 Header 参数

参数	类型	必选	说明
x-environment	String	否	指定查询配置的environment，取值 development testing acceptance production。其他值均被作为空值处理。

## 响应消息

### 响应参数

参数说明请参见表6-43。（若传入的revision值与当前配置的revision一致，则无响应消息）

表 6-43 参数说明

参数	参数类型	描述
{dimension} (参数名称和参数个数不固定)	Map<String, String>	配置内容
revision	表6-44	revision

表 6-44 revision

参数	参数类型	描述
version	String	配置版本

## 示例

### 请求示例

```
GET /v3/default/configuration/items?dimensionsInfo=service%40app%231.0.0
```

### 响应示例

```
{
  "service@app": {
    "k1": "v1"
  },
  "service@app#1.0.0": {
    "k2": "v2"
  }
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.20 删除微服务信息

### 功能介绍

删除一个微服务及其相关信息，同时注销其所有实例信息。

### URI

DELETE /v4/{project\_id}/registry/microservices/{serviceId}

表 6-45 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。
serviceId	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 $^{\wedge}.*\$$

表 6-46 Query 参数

参数	类型	必选	说明
force	Boolean	否	是否强制删除。包括： <ul style="list-style-type: none"><li>• true表示强制删除。</li><li>• false表示非强制删除。</li></ul> 选择强制删除会自动注销所有服务实例，并且删除相关的服务依赖关系，如果未传此参数，那么在服务拥有实例时，无法被删除。

### 请求消息

无

### 响应消息

响应参数

无

## 示例

### 请求示例

```
DELETE /v4/default/registry/microservices/e0f0da073f2c91e8979a89ff2d7c69t6
```

### 响应示例

无

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.21 批量删除微服务静态信息

### 功能介绍

批量删除指定的微服务定义及其相关信息，同时注销其所有实例信息。

### URI

```
DELETE /v4/{project_id}/registry/microservices
```

表 6-47 路径参数

参数	类型	必选	说明
project_id	String	是	请填写固定值：default。

### 请求消息

#### 请求参数

表 6-48 请求 Body 参数

参数	类型	必选	说明
serviceIds	Array of strings	是	微服务ID列表。

参数	类型	必选	说明
force	Boolean	否	是否强制删除。包括： <ul style="list-style-type: none"><li>• false表示非强制删除。</li><li>• true表示强制删除。</li></ul> 默认值：false 选择强制删除会自动注销所有服务实例，并且删除相关的服务依赖关系，如果未传此参数，那么在服务被依赖或者拥有实例时，无法被删除。

## 响应消息

### 响应参数

参数说明请参见[表6-49](#)。

表 6-49 参数说明

参数	类型	说明
services	Array of objects	具体参考 <a href="#">DelServicesResponse</a> 。

## 示例

### 请求示例

```
{
  "serviceIds": [
    "string"
  ],
  "force": boolean
}
```

### 响应示例

```
{
  "services": [
    {
      "serviceId": "string"
    }
  ]
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。



## 6.22 批量查询微服务实例

### 功能介绍

实例注册后可以根据微服务版本规则或字段条件批量发现多个微服务的实例。

### URI

POST /v4/{project\_id}/registry/instances/action

表 6-50 路径参数

参数	类型	必选	说明
project_id	String	是	请填写固定值：default。

表 6-51 Query 参数

参数	类型	必选	说明
type	String	是	操作类型。 填固定值：query。

### 请求消息

表 6-52 请求 Header 参数

参数	类型	必选	说明
X-ConsumerId	String	否	服务消费者唯一标识。

表 6-53 参数说明

参数	类型	必选	说明
services	Array of objects	否	微服务请求结构体，具体参考 <a href="#">表6-54</a> 。
instances	Array of objects	否	微服务实例请求结构体，具体参考 <a href="#">表6-55</a> 。

表 6-54 FindService 参数说明

参数	类型	必选	说明
service	Object	是	微服务请求结构体，具体参考 <a href="#">DependencyMicroService</a> 。
rev	String	否	客户端缓存的版本号。

表 6-55 FindInstance 参数说明

参数	类型	必选	说明
instance	Object	是	微服务实例请求结构体，具体参考 <a href="#">HeartbeatSetElement</a> 。
rev	String	否	客户端缓存的版本号。

## 响应消息

### 响应参数

参数说明请参见[表6-56](#)。

表 6-56 参数说明

参数	类型	说明
services	BatchFindResult	微服务发现的响应结构体。
instances	BatchFindResult	微服务实例发现的响应结构体。

表 6-57 BatchFindResult 参数说明

参数	类型	说明
failed	Array of objects	部分微服务发现失败的响应结构体，具体参考 <a href="#">表6-58</a> 。
notModified	Array of integers	实例没有发生变化的索引数组，每一项的值对应请求结构体中微服务数组的索引值。
updated	Array of objects	实例发生变化的响应结构体，具体参考 <a href="#">表6-59</a> 。

表 6-58 FindFailedResult 参数说明

参数	类型	说明
indexes	Array of integers	微服务发现失败的索引数组，每一项的值对应请求结构体中微服务数组的索引值。
error	Object	错误信息结构体，具体参考 <a href="#">Error</a> 。

表 6-59 FindResult 参数说明

参数	类型	说明
index	Integer	实例发生变化的索引数组，每一项的值对应请求结构体中微服务数组的索引值。
rev	String	返回给客户端缓存的版本号。
instances	Array of objects	微服务实例的集合，具体参考 <a href="#">MicroServiceInstance</a> 。

## 示例

### 请求示例

```
curl -X POST "/v4/default/registry/instances/action?type=query" -H "accept: application/json" -H "Content-Type: application/json" -d '{
  "services": [
    {
      "service": {
        "environment": "string",
        "appld": "string",
        "serviceName": "string",
        "version": "string"
      },
      "rev": "string"
    }
  ],
  "instances": [
    {
      "instance": {
        "serviceId": "string",
        "instanceId": "string"
      },
      "rev": "string"
    }
  ]
}
```

### 响应示例

```
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Server: SERVICECENTER/0.0.1
Vary: Accept-Encoding
Vary: Origin
X-Response-Status: 200
Date: Fri, 21 Dec 2018 02:36:48 GMT
Content-Length: 1952
```

```
{
  "services": {
    "failed": [
      {
        "indexes": [
          0
        ],
        "error": {
          "errorCode": "string",
          "errorMessage": "string",
          "detail": "string"
        }
      }
    ],
    "notModified": [
      0
    ],
    "updated": [
      {
        "index": 0,
        "rev": "string",
        "instances": [
          {
            "instanceId": "string",
            "serviceId": "string",
            "version": "string",
            "hostName": "string",
            "endpoints": [
              "string"
            ],
            "status": "string",
            "healthCheck": {
              "mode": "string",
              "port": 0,
              "interval": 0,
              "times": 0
            },
            "dataCenterInfo": {
              "name": "string",
              "region": "string",
              "availableZone": "string"
            },
            "timestamp": "string",
            "modTimestamp": "string"
          }
        ]
      }
    ]
  },
  "instances": {
    "failed": [
      {
        "indexes": [
          0
        ],
        "error": {
          "errorCode": "string",
          "errorMessage": "string",
          "detail": "string"
        }
      }
    ],
    "notModified": [
      0
    ],
    "updated": [
      {
        "index": 0,
```

```
"rev": "string",
"instances": [
  {
    "instanceId": "string",
    "serviceId": "string",
    "version": "string",
    "hostName": "string",
    "endpoints": [
      "string"
    ],
    "status": "string",
    "healthCheck": {
      "mode": "string",
      "port": 0,
      "interval": 0,
      "times": 0
    },
    "dataCenterInfo": {
      "name": "string",
      "region": "string",
      "availableZone": "string"
    },
    "timestamp": "string",
    "modTimestamp": "string"
  }
]
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

## 6.23 查询微服务的所有契约

### 功能介绍

查询一个微服务的所有契约信息，包括schemaId和summary信息。

### URI

GET /v4/{project\_id}/registry/microservices/{serviceId}/schemas

表 6-60 路径参数

参数	类型	必选	说明
project_id	String	是	请填固定值：default。
serviceId	String	是	微服务唯一标识。 字符长度为1~64。 正则表达式为 $^{\wedge}.*$$

表 6-61 Query 参数

参数	类型	必选	说明
withSchema	String	否	是否查询schema内容，默认为0。包括： <ul style="list-style-type: none"><li>0表示不显示schema内容只有schemald和summary。</li><li>1表示显示schemald、summary和schema。</li></ul>

## 请求消息

### 请求参数

无

## 响应消息

### 响应参数

参数说明请参见[表6-62](#)。

表 6-62 参数说明

参数	类型	说明
schemas	Array of objects	成功获取微服务所有schema结构体的列表，具体参考 <a href="#">Schema</a> 。

## 示例

### 请求示例

无

### 响应示例

```
{
  "schemas": [
    {
      "schemald": "xxxx",
      "schema": "xxxx", // 当url带withSchema=1参数才返回，默认不返回
      "summary": "abcd"
    }
  ]
}
```

## 状态码

状态码请参见[状态码](#)。

## 错误码

错误码请参见[CSE错误码](#)。

# 7 数据结构

## 7.1 参数说明

### 7.1.1 HealthCheck

表 7-1 参数说明

参数	类型	必选	说明
mode	String	是	check模式, push/pull。
port	Integer	否	端口。 字符长度不超过65536。 正则表达式为^[0-9]*\$
interval	Integer	是	心跳间隔(秒), 当值小于5秒时, 按5秒注册。 字符长度不超过4294967296。 正则表达式为^[0-9]+\$
times	Integer	是	retry times。 字符长度不超过4294967296。 正则表达式为^[0-9]+\$



## 7.1.2 MicroServiceInstance

表 7-2 参数说明

参数	类型	必选	说明
instanceId	String	否	实例id, 唯一标识。创建实例, instanceId 由service-center产生。
serviceId	String	否	微服务唯一标识, 创建实例时, 以url里面的为准。
version	String	否	微服务版本号
hostName	String	是	计算机名称, 可以为本机的域名, 或IP地址等。
endpoints	Array of strings	是	例: rest:127.0.0.1:8080。
status	String	是	实例状态。包括: <ul style="list-style-type: none"><li>• UP</li><li>• DOWN</li><li>• STARTING</li><li>• OUTOFSERVICE</li></ul>
properties	Map<String, String>	否	扩展属性。可以自定义KEY和相应的Value。
healthCheck	<a href="#">HealthCheck</a>	否	服务健康模式检查。定义微服务的健康检查模式, 可选的模式为主动推送 (push) 和被动拉取 (pull)。interval定义拉取时间间隔 (秒)、times定义最大尝试请求次数。在被动拉取模式下需定义port, 由注册中心主动向该port进行链接。默认为push模式, interval默认为30, times默认为3
timestamp	String	否	实例创建时间戳, 自动生成。
modTimestamp	String	否	更新时间
dataCenterInfo	<a href="#">DataCenterInfo</a>	否	时区。

## 7.1.3 HeartbeatSetElement

表 7-3 参数说明

参数	类型	必选	说明
serviceld	String	是	微服务id 字符长度为1~64。 正则表达式为 $^{\wedge}.*\$$
instanceid	String	是	微服务实例id 字符长度不超过64位。 正则表达式为 $^{\wedge}[A-Za-z0-9_-.]*\$$

## 7.1.4 InstanceHbRst

表 7-4 参数说明

参数	类型	必选	说明
serviceld	String	否	微服务id
instanceid	String	否	微服务实例id
errorMessage	String	否	错误信息，成功为空，不成功，则为具体的错误信息。

## 7.1.5 DelServicesResponse

表 7-5 参数说明

参数	类型	必选	说明
serviceld	String	否	微服务id

## 7.1.6 MetricData

表 7-6 参数说明

参数	类型	必选	说明
name	String	是	服务名称。
appld	String	是	应用Id。

参数	类型	必选	说明
serviceId	String	否	服务Id。
version	String	是	服务版本号。
instanceId	String	否	实例Id。
instance	String	是	实例名称。
thread	String	否	线程数。
cpu	String	否	CPU使用率。
memory	String	否	内存。
interfaces	<a href="#">表7-7</a>	否	实例包含的详细接口数据。

#### 说明

serviceId和instanceId两个参数必须同时存在或者同时不存在。

表 7-7 items 参数说明

参数	类型	必选	说明
items	FunctionMetricInfo	否	接口级别聚合指标数据。

## 7.1.7 FunctionMetricInfo

表 7-8 参数说明

参数	类型	必选	说明
<a href="#">AggregateMetricInfo</a>	Object	否	聚合指标数据。
l995	Integer	否	99.5%的请求低于此值。
l99	Integer	否	99%的请求低于此值。
l90	Integer	否	90%的请求低于此值。
l75	Integer	否	75%的请求低于此值。
l50	Integer	否	50%的请求低于此值。
l25	Integer	否	25%的请求低于此值。
l5	Integer	否	5%的请求低于此值。

## 7.1.8 InstanceMetricInfo

表 7-9 参数说明

参数	类型	必选	说明
<b>AggregateMetricInfo</b>	Object	否	聚合指标数据。
thread	Integer	否	线程数。
cpu	number	否	CPU使用率。
memory	Object	否	内存信息。

## 7.2 公共请求参数

### 7.2.1 MicroService

表 7-10 参数说明

参数	类型	必选	说明
serviceId	String	否	微服务唯一标识。 字符长度为1~64。 正则表达式为 <code>^.*\$</code>
environment	String	否	用于区分服务阶段，取值为development testing acceptance production。 当配置为development、testing或acceptance时，可以新增或者修改已存在的Schema；当配置为production时，则不可以新增或者修改Schema。 默认值：development
appId	String	是	应用App唯一标识。 字符长度为1~160。 正则表达式为 <code>^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$</code>
serviceName	String	是	微服务名称，同一个App要保证唯一。 字符长度为1~128。 正则表达式为 <code>^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$</code>

参数	类型	必选	说明
version	String	是	微服务版本号。 字符长度为1~64。 正则表达式为 <code>^[0-9]*\$ ^[0-9]+(\.[0-9]+)*\$</code>
description	String	否	微服务描述信息。 字符长度不超过256。
level	String	否	微服务层级。包括： <ul style="list-style-type: none"><li>• FRONT</li><li>• MIDDLE</li><li>• BACK</li></ul>
timestamp	String	否	微服务注册时间。
schemas	String	否	微服务访问的契约内容。支持数字、字母，支持使用括号内字符做连接符( <code>_.</code> )，长度1-160字节，数组长度最大100个契约
status	String	否	微服务状态，UP表示上线，DOWN表示下线。默认值UP
paths	ServicePath	否	服务路由信息。
framework	Framework	否	微服务开发框架。
registerBy	String	否	微服务注册方式。包括： <ul style="list-style-type: none"><li>• SDK</li><li>• PLATFORM</li><li>• SIDECAR</li><li>• UNKNOWN</li></ul>
modTimestamp	String	否	最后修改UTC时间
properties	Map<String, String>	否	微服务扩展信息，可以自定义Key和相应的Value。
alias	String	否	微服务别名，字符长度为1-128。 正则表达式为 <code>^[a-zA-Z0-9_-\.:]*\$</code>

表 7-11 ServicePath 参数说明

参数	类型	必选	说明
Path	String	否	路由地址。支持数字、字母和括号内字符(.,?'\/+&%\$#=#~_-@{}), 长度1-160字节
Property	<b>Properties</b>	否	扩展属性。

表 7-12 Framework 参数说明

参数	类型	必选	说明
name	String	否	微服务开发框架，默认值为 UNKNOWN
version	String	否	微服务开发框架版本号，支持任意字符，长度1-64字节

## 7.2.2 Properties

表 7-13 参数说明

参数	类型	必选	说明
< * >	String	否	微服务扩展属性，可以自定义KEY和相应的Value。长度最小1字节。

## 7.2.3 DependencyMicroService

表 7-14 参数说明

参数	类型	必选	说明
appId	String	是	应用app唯一标识。 字符长度为1~160。 正则表达式为^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$

参数	类型	必选	说明
serviceName	String	是	微服务名称。 作为provider支持为*，表示依赖同一租户下的所有服务。当服务名称为*时，appId和version可以省略。 consumer不支持*。 字符长度为1~128。 正则表达式为 <code>^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$</code>
version	String	是	微服务版本。 作为provider支持+、固定版本和latest（当前最新版本），如1.0.1+，表示1.0.1以上的版本，包括1.0.1。 作为consumer只能为固定版本。 字符长度为1~64。 正则表达式为 <code>^[0-9]*\$ ^[0-9]+(\.[0-9]+)*\$</code>
environment	String	否	微服务的环境信息。包括： <ul style="list-style-type: none"> <li>development</li> <li>testing</li> <li>acceptance</li> <li>production</li> </ul>

## 7.2.4 Rule

表 7-15 参数说明

参数	类型	必选	说明
ruleId	String	否	黑白名单ID
ruleType	String	是	类型，WHITE表示白名单、BLACK表示黑名单
attribute	String	是	如果是tag_xxx开头，则按Tag过滤attribute属性。否则，则按ServiceId、AppId、ServiceName、Version、Description、Level、Status过滤。
pattern	String	是	匹配规则，正则表达式，长度1到64个字符。
timestamp	String	否	创建rule的时间，只有获取rule时返回使用。

参数	类型	必选	说明
description	String	否	Rule的描述信息，字符长度不超过256。
modTimestamp	String	否	最后修改UTC时间

## 7.2.5 AddOrUpdateRule

表 7-16 参数说明

参数	类型	必选	说明
ruleType	String	否	类型，WHITE表示白名单、BLACK表示黑名单
attribute	String	否	如果是tag_xxx开头，则按Tag过滤attribute属性。否则，则按ServiceId、Appld、ServiceName、Version、Description、Level、Status过滤。
pattern	String	否	匹配规则，正则表达式，长度1到64个字符。
description	String	否	rule的描述。 字符长度不超过256个字符。

## 7.2.6 DataCenterInfo

表 7-17 参数说明

参数	类型	必选	说明
name	String	是	数据中心名称，支持数字、字母，支持使用字符_-做连接符，长度1-128字节。
region	String	是	数据中心所属区域名称，支持数字、字母，支持使用字符_-做连接符，长度1-128字节。
availableZone	String	是	数据中心所属可用区名称，支持数字、字母，支持使用字符_-做连接符，长度1-128字节。



## 7.2.7 Schema

表 7-18 参数说明

参数	类型	必选	说明
schemald	String	否	微服务契约ID。支持数字、字母和_-.字符，长度1-160字节。
schema	String	否	微服务契约内容。任意字符、长度1-2048字节。
summary	String	否	微服务契约摘要。支持数字、字母，长度0-128字节。

## 7.2.8 environment\_setting

表 7-19 environment\_setting 参数说明

参数	类型	描述
cluster_id	String	k8s的集群id
language	String	语言类型和版本，至少提供一个，如：Java 8
node_label_selector	String	过滤节点标签

## 7.2.9 phases

表 7-20 phases 参数说明

参数	类型	描述
build	<a href="#">表7-21</a>	构建语法结构体

表 7-21 build 参数说明

参数	类型	描述
commands	<a href="#">表7-22</a>	构建命令结构体

表 7-22 commands 参数说明

参数	类型	描述
commands	String	软件编译命令

## 7.3 公共响应参数

### 7.3.1 WatchMicroServiceKey

表 7-23 参数说明

参数	类型	必选	说明
appld	String	否	应用app唯一标识。 字符长度为1~160。 正则表达式为^[a-zA-Z0-9]*\$ ^ [a-zA-Z0-9][a-zA-Z0-9_\.]* [a-zA-Z0-9]\$
serviceName	String	否	微服务名称。 作为provider支持为*，表示依赖同一租户下的所有服务。当服务名称为*时，appld和version可以省略。 consumer不支持*。
version	String	否	微服务版本。 作为provider支持+、固定版本和latest（当前最新版本），如1.0.1+，表示1.0.1以上的版本，包括1.0.1。 作为consumer只能为固定版本。

### 7.3.2 SuccessdResponse

表 7-24 参数说明

参数	类型	必选	说明
Result	String	是	返回结果。

### 7.3.3 ServiceInfo

表 7-25 参数说明

参数	类型	必选	说明
serviceId	String	是	微服务唯一标识。字符为uuid。 正则表达式为 <code>^.*\$</code>
appId	String	是	应用app唯一标识。 字符长度为1~160。 正则表达式为 <code>^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$</code>
serviceName	String	是	微服务名称，同一个App要保证唯一。 字符长度为1~128。 正则表达式为 <code>^[a-zA-Z0-9]*\$ ^[a-zA-Z0-9][a-zA-Z0-9_\.]*[a-zA-Z0-9]\$</code>
version	String	是	微服务版本号。 字符长度为1~64。 正则表达式为 <code>^[0-9]*\$ ^[0-9]+(\.[0-9]+)*\$</code>
level	String	是	微服务层级。包括： <ul style="list-style-type: none"><li>• FRONT</li><li>• MIDDLE</li><li>• BACK</li></ul>
status	String	是	服务状态。 <ul style="list-style-type: none"><li>• UP</li><li>• DOWN</li></ul>
timestamp	String	是	时间戳。

### 7.3.4 AggregateMetricInfo

表 7-26 参数说明

参数	类型	必选	说明
time	Integer	否	上报时间。
name	String	否	接口名称。

参数	类型	必选	说明
qps	Number	否	当前qps。
latency	Number	否	平均时延。
rate	Integer	否	成功率。
total	Integer	否	总请求数。
isCircuitBreakerOpen	Boolean	否	断路器是否打开。
failure	Integer	否	总失败数。
shortCircuited	Integer	否	总短路个数。
semaphoreRejected	Integer	否	总信号量拒绝个数。
threadPoolRejected	Integer	否	总线程拒绝个数。
countTimeout	Integer	否	总超时个数。

### 7.3.5 ServiceDetail

表 7-27 参数说明

参数	类型	必选	说明
microService	<a href="#">MicroService</a>	否	服务信息
instances	<a href="#">MicroService Instance</a>	否	实例信息
schemaInfos	<a href="#">Schema</a>	否	微服务的API详细信息
rules	<a href="#">Rule</a>	否	黑白名单信息
providers	<a href="#">MicroService</a>	否	所有的provider信息
consumers	<a href="#">MicroService</a>	否	所有的consumer信息
tags	Tags	否	微服务所有的标签信息
microServiceVersions	Array, 每一项为String	否	微服务的所有版本信息

表 7-28 Tags 参数说明

参数	类型	必选	说明
tags	<b>Properties</b>	否	微服务扩展属性请求结构体

## 7.3.6 TenantProject

表 7-29 TenantProject 参数

参数	位于	参数类型	描述
tenant	body	String	租户名称
projectId	body	String	id

## 7.3.7 Error

表 7-30 参数说明

参数	类型	必选	说明
errorCode	String	是	错误代码。
errorMessage	String	是	错误信息。
detail	String	否	详细定位信息。

# 8 权限策略说明

## 8.1 策略及授权项说明

如果您需要对您所拥有的ServiceStage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），如果帐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ServiceStage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或角色，才能使用户组中的用户获得相应的权限，这一过程称为授权。授权后，用户就可以基于已有权限对云服务进行操作。

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。

帐号具备所有接口的调用权限，如果使用帐号下的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。

## 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，管理员权限用户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。
- 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。
- 对应API接口：自定义策略实际调用的API接口。

表 8-1 ServiceStage 授权项明细

权限	授权项	对应API接口
创建应用	servicestage:app:create	POST /v2/{project_id}/cas/applications
根据应用ID删除应用	servicestage:app:delete	DELETE /v2/{project_id}/cas/applications/{application_id}
修改应用信息	servicestage:app:modify	PUT /v2/{project_id}/cas/applications/{application_id}
获取所有应用	servicestage:app:list	GET /v2/{project_id}/cas/applications
根据应用ID获取应用详细信息	servicestage:app:get	GET /v2/{project_id}/cas/applications/{application_id}
查询指定微服务静态信息	cse:registry:get	GET /v4/{project_id}/registry/microservices/{serviceId}
查询所有微服务静态信息	cse:registry:get	GET /v4/{project_id}/registry/microservices
创建微服务静态信息	cse:registry:modify	POST /v4/{project_id}/registry/microservices
修改微服务静态信息	cse:registry:modify	PUT /v4/{project_id}/registry/microservices/{serviceId}/properties
查询微服务访问契约	cse:registry:get	GET /v4/{project_id}/registry/microservices/{serviceId}/schemas/{schemaId}
修改微服务访问契约	cse:registry:modify	PUT /v4/{project_id}/registry/microservices/{serviceId}/schemas/{schemaId}
创建服务间依赖关系	cse:registry:modify	PUT /v4/{project_id}/registry/dependencies
查询服务providers	cse:registry:get	GET /v4/{project_id}/registry/microservices/{serviceId}/providers
查询微服务唯一标识信息	cse:registry:get	GET /v4/{project_id}/registry/existence
注册微服务实例	cse:registry:modify	POST /v4/{project_id}/registry/microservices/{serviceId}/instances
根据serviceId查询微服务实例	cse:registry:get	GET /v4/{project_id}/registry/microservices/{serviceId}/instances

权限	授权项	对应API接口
注销微服务实例	cse:registry:modify	DELETE /v4/{project_id}/registry/microservices/{serviceId}/instances/{instanceId}
查询微服务实例详细信息	cse:registry:get	GET /v4/{project_id}/registry/microservices/{serviceId}/instances/{instanceId}
修改微服务实例扩展信息	cse:registry:modify	PUT /v4/{project_id}/registry/microservices/{serviceId}/instances/{instanceId}/properties
修改微服务实例状态	cse:registry:modify	PUT /v4/{project_id}/registry/microservices/{serviceId}/instances/{instanceId}/status
发送心跳信息	cse:registry:modify	PUT /v4/{project_id}/registry/microservices/{serviceId}/instances/{instanceId}/heartbeat
按条件查询微服务实例	cse:registry:get	GET /v4/{project_id}/registry/instances
查询配置	cse:config:get	GET /v3/{project_id}/configuration/items
删除微服务信息	cse:registry:modify	DELETE /v4/{project_id}/registry/microservices/{serviceId}
批量删除微服务静态信息	cse:registry:modify	DELETE /v4/{project_id}/registry/microservices
批量查询微服务实例	cse:registry:modify	POST /v4/{project_id}/registry/instances/action
查询微服务的所有契约	cse:registry:get	GET /v4/{project_id}/registry/microservices/{serviceId}/schemas



# 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	编码	错误码说明
200	-	操作成功。
400	BadRequest	非法请求。 建议直接修改该请求，不要重试该请求。
401	Unauthorized	在客户端提供认证信息后，返回该状态码，表明服务端指出客户端所提供的认证信息不正确或非法。
404	NotFound	所请求的资源不存在。 建议直接修改该请求，不要重试该请求。
422	UnprocessableEntity	请求格式正确，但是由于含有语义错误，无法响应。
500	InternalServerError	表明服务端能被请求访问到，但是不能理解用户的请求。

## 9.2 ServiceStage 错误码

在向系统发出请求后，如果遇到错误，会在响应中包含响应的错误码描述错误信息。错误码如下所示。

## 应用管理 API 错误码

表 9-2 错误码列表

状态码	错误码	错误信息	描述	处理措施
400	SVCSTG.00100400	parameter invalid	错误请求	检查参数是否正确，稍后重试。
	SVCSTG.00100402	component was deployed, uninstall first	组件已部署，请先卸载	卸载组件后重试。
	SVCSTG.00100414	job id can not empty	job id不能为空	请指定job id进行查询
	SVCSTG.00100418	component instance action not defined	未定义组件实例操作	请指定合法的组件实例操作
	SVCSTG.00100420	runtime not supported	不支持运行时	请指定支持的运行时
	SVCSTG.00100422	base resource can not empty	基础资源不能为空	请设置基础资源
	SVCSTG.00100425	instance artifacts can not be empty	实例的软件包或者镜像为空	报文中的artifact参数不能为空
	SVCSTG.00100449	environment name is already used	环境名已被占用	重新提供一个新的环境名
404	SVCSTG.00100403	application not found	应用未找到	请指定已经存在的应用
	SVCSTG.00100404	component not found	组件未找到	请指定已经存在的组件
	SVCSTG.00100408	request not found	请求未找到	请指定已经存在的请求
	SVCSTG.00100411	environment not found	环境未找到	请指定存在的环境
	SVCSTG.00100412	instance not found	实例未找到	请对存在的实例进行操作
	SVCSTG.00100424	elb instance not found	elb实例未找到	请指定已经存在的elb实例
	SVCSTG.00100429	job not found	构建工程未找到	请指定已经存在的构建工程

状态码	错误码	错误信息	描述	处理措施
	SVCSTG.00100433	application release not found	发布任务未找到	请选择已经存在的发布任务
	SVCSTG.00100436	snapshot not found	实例快照未找到	请指定已经存在的实例快照
	SVCSTG.00100438	template image not found	模板镜像未找到	请指定已经存在的模板镜像
	SVCSTG.00100441	instance version is already used by other instances in the same environment	组件实例的版本号已被相同环境的其他实例所占用。	重新提供一个新的版本号
405	SVCSTG.00100421	account is limit suspended	帐户已被暂停限额	恢复帐号至正常状态
	SVCSTG.00100427	permission denied	没有权限	重新授权
408	SVCSTG.00100428	request timeout	系统处理超时	系统繁忙，请稍后再试
409	SVCSTG.00100401	application not empty	应用不为空	删除应用下的组件再删除应用
	SVCSTG.00100413	instance name exists	实例名已经存在	更换实例名
	SVCSTG.00100419	can not operate in current status	当前状态不能操作	当前操作未完成，请稍后再试
	SVCSTG.00100423	component instance is not deletable	组件实例当前状态不可删除	当前操作未完成，请稍后再试
	SVCSTG.00100426	environment is in use	环境已经被使用	删除部署在环境上的实例后再删除环境
	SVCSTG.00100432	component in releasing	组件当前已在发布任务中	请等待当前发布任务执行结束后再重试

状态码	错误码	错误信息	描述	处理措施
	SVCSTG.00100435	release can not delete	发布任务当前状态不能删除	当前操作未完成，请稍后再试
	SVCSTG.00100439	elb is already used in other instance of component	elb实例已经在组件中被使用	请使用其他elb实例重试
500	SVCSTG.00100500	service internal error	服务内部错误	系统繁忙，请稍后重试
	SVCSTG.00100501	remote service error	远程服务错误	系统繁忙，请稍后重试

## 访问 Git 仓库 API 错误码

表 9-3 错误码说明

状态码	错误码	错误信息	描述	处理措施
400	SVCSTG.REP.O.0400	parameter invalid	错误请求	检查参数是否正确，稍后重试。
	SVCSTG.REP.O.0401	repository not implemented	错误请求	请选择其他仓库类型。
	SVCSTG.REP.O.0402	oauth code or state invalid	错误请求	请重新进行OAuth2授权，获取正确的认证码和随机串。
	SVCSTG.REP.O.0403	resource already exist	错误请求	调用查询接口，检查是否存在重复的资源。
	SVCSTG.REP.O.0450	no repository authorization	错误请求	检查请求header是否输入正确授权名称。
	SVCSTG.REP.O.0451	repository authorization invalid	错误请求	重新授权。
	SVCSTG.REP.O.0451	clone url invalid	错误请求	检查查询参数clone_url是否正确。 示例: https://user-name@github.com/user-name/demo.git。
500	SVCSTG.REP.O.0500	service internal error	服务内部错误	网络异常等原因，稍后重试。

状态码	错误码	错误信息	描述	处理措施
	SVCSTG.REP O.0501	remote repository error	服务内部 错误	检查请求参数是否正确，或者远端GIT的仓库文件已经被修改。

## 9.3 CSE 错误码

在向系统发出请求后，如果遇到错误，会在响应中包含响应的错误码描述错误信息。错误码如表9-4所示。

表 9-4 错误码列表

分类	状态码	错误码	描述	处理措施
配置管理	400	SVCSTG.00300607	dimension无效	根据提示，修改dimensionsInfo参数，确保符合参数规范要求。
		SVCSTG.00400641	配置项格式错误	请填写符合格式要求： <ul style="list-style-type: none"> <li>配置项支持数字、字母和.-_:[]六个字符，长度1-2048位。</li> <li>值支持数字、字母和括号内的特殊字符（~!@#\$%^&amp;*()_{} :"&lt;&gt;?'-[]./=），微服务专业版支持长度1-2048位，专享版支持长度1-131072位。</li> </ul>
		SVCSTG.00400642	配置配额超限	<ul style="list-style-type: none"> <li>微服务引擎专享版：删除部分配置。</li> <li>微服务引擎专业版：删除部分配置，或申请微服务引擎专享版。</li> </ul>
	401	SVCSTG.00300401	认证不通过	输入正确的token。
	500	SVCSTG.00300605	服务内部错误	内部错误请联系运维支持。
		SVCSTG.00300500	服务内部错误	内部错误请联系运维支持。
		SVCSTG.00300501	服务内部错误	内部错误请联系运维支持。
微服务	400	400001	非法参数	根据错误提示中的规则，修改提示中的参数。

分类	状态码	错误码	描述	处理措施
		400010	服务已存在	修改创建微服务body体中的servicId或微服务描述信息。
		400012	服务不存在	请输入有效的servicId。
		400013	无法删除该微服务，该微服务已部署实例	请先将实例下线，再删除微服务；或强制删除微服务（url中添加query参数“force=true”）。
		400014	schemald不存在	请输入有效的schemald。
		400015	schema不允许修改	该schema已注册，不支持修改。
		400016	schema不存在	请先注册schema再查询。
		400017	实例不存在	请输入有效的instanceId。
		400023	无法删除该微服务，该微服务被其他微服务依赖	可选择强制删除微服务（url中添加query参数“force=true”）。
		400026	微服务版本不存在	请输入正确的版本号或版本号范围。
		400100	配额不足	对应的资源（如微服务、实例、schema）配额不足，请删除部分资源再创建。
	401	401002	认证不通过	输入正确的token。
	500	500003	内部错误	内部错误请联系运维支持。
		500011	后端错误	内部错误请联系运维支持。

## 9.4 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中需要填入项目编号，所以需要获取到项目编号。项目编号获取步骤如下：

- 步骤1** 注册并登录管理控制台。
- 步骤2** 单击用户名，在下拉列表中单击“我的凭证”。
- 步骤3** 在“项目列表”中查看项目ID。

----结束


## 9.5 获取帐号 ID

在调用接口的时候，部分URL中需要填入帐号ID（domain-id），所以需要先获取到帐号ID。帐号ID获取步骤如下：

- 步骤1** 注册并登录管理控制台。
- 步骤2** 鼠标移动到登录用户名，在下拉列表中单击“我的凭证”。  
查看帐号ID。  
----结束

## 9.6 获取服务注册发现地址

在调用微服务引擎专享版的API时，需要通过微服务引擎专享版的服务注册发现地址来访问。服务注册发现地址获取步骤如下：

- 步骤1** 登录ServiceStage控制台，选择“基础设施 > 微服务引擎(CSE)”。
- 步骤2** 选择待查看的微服务引擎，可以查看或单击  复制服务注册发现地址。  
----结束