

腾讯云 TI 平台 TI-EMS

API 文档

产品文档



腾讯云

【版权声明】

©2013–2022 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100。

文档目录

API 文档

更新历史

简介

API 概览

调用方式

请求结构

公共参数

签名方法 v3

签名方法

返回结果

服务管理相关接口

创建服务

删除服务

更新服务

描述服务

暴露服务

资源管理相关接口

获取节点列表

停用资源组的伸缩组

获取资源组列表

查询伸缩组活动

删除资源组

删除节点

启用资源组的伸缩组

查询资源组的伸缩组信息

更新资源组的伸缩组

创建资源组的伸缩组

删除资源组的伸缩组

配置管理相关接口

描述服务配置

删除服务配置

创建服务配置

任务管理相关接口

更新任务

创建任务

删除任务

其他接口

描述服务运行环境

创建运行环境

删除运行环境

数据结构

错误码

API 文档

更新历史

最近更新时间：2020-09-15 08:07:36

第 7 次发布

发布时间：2020-04-01 08:39:10

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateJob](#)
 - 新增入参：LogTopicId
- [CreateRsgAsGroup](#)
 - 新增入参：InstanceType, DesiredSize
- [CreateService](#)
 - 新增入参：LogTopicId
- [UpdateRsgAsGroup](#)
 - 新增入参：DesiredSize
- [UpdateService](#)
 - 新增入参：LogTopicId

第 6 次发布

发布时间：2020-03-30 08:45:21

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [RsgAsGroup](#)
 - 新增成员：DesiredSize

第 5 次发布

发布时间：2020-03-16 08:19:19

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [UpdateRsgAsGroup](#)
 - 新增入参：MaxSize, MinSize

修改数据结构：

- [Job](#)
 - 新增成员：LogTopicId
- [JobStatus](#)
 - 新增成员：ReplicaInfos

- [ModelService](#)
 - 新增成员：LogTopicId

第 4 次发布

发布时间：2020-01-02 17:10:55

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateRsgAsGroup](#)
- [DeleteInstance](#)
- [DeleteResourceGroup](#)
- [DeleteRsgAsGroup](#)
- [DescribeResourceGroups](#)
- [DescribeRsgAsGroupActivities](#)
- [DescribeRsgAsGroups](#)
- [DisableRsgAsGroup](#)
- [EnableRsgAsGroup](#)
- [UpdateRsgAsGroup](#)

修改接口：

- [CreateJob](#)
 - 新增入参：QuantizationInput
 - 修改入参：ResourceGroupId, WorkerCount, ConfigId
- [CreateService](#)
 - 修改入参：ResourceGroupId
- [UpdateService](#)
 - 新增入参：Cpu, Memory, Gpu

新增数据结构：

- [QuantizationInput](#)
- [ReplicaInfo](#)
- [ResourceGroup](#)
- [RsgAsActivityRelatedInstance](#)
- [RsgAsGroup](#)
- [RsgAsGroupActivity](#)

修改数据结构：

- [ExposeInfo](#)
 - 新增成员：GateWayServiceId, GateWayAPIId, GateWayDomain
- [Instance](#)
 - 新增成员：RsgAsGroupId
- [Job](#)
 - 新增成员：JobType, QuantizationInput
- [ServiceStatus](#)
 - 新增成员：ReplicaInfos

第 3 次发布

发布时间：2019-11-15 17:20:20

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [ExposeService](#)

新增数据结构：

- [ExposeInfo](#)

删除数据结构：

- Expose

修改数据结构：

- [ModelService](#)
 - 修改成员：Exposes

第 2 次发布

发布时间：2019-11-14 21:54:14

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateJob](#)
- [CreateRuntime](#)
- [DeleteJob](#)
- [DeleteRuntime](#)
- [DescribeInstances](#)
- [UpdateJob](#)

修改接口：

- [CreateService](#)
 - 新增入参：Cpu, Memory, ResourceGroupId, Authentication, Gpu, GpuMemory, Description, GpuType
 - 修改入参：ServiceConfigId
 - 修改出参：Service
- [CreateServiceConfig](#)
 - 新增入参：Description
 - 删除入参：Cpu, Memory, TfloppUnits, GpuMemory
 - 修改出参：ServiceConfig
- [DeleteService](#)
 - 修改入参：ServiceId
- [DeleteServiceConfig](#)
 - 修改入参：ServiceConfigId
- [DescribeServiceConfigs](#)
 - 新增入参：PageByName
 - 修改出参：ServiceConfigs
- [DescribeServices](#)

- 修改出参：Services
- [UpdateService](#)
 - 新增入参：Description, GpuType
 - 修改入参：ServiceId, ServiceConfigId
 - 修改出参：Service

新增数据结构：

- [Config](#)
- Expose
- [Instance](#)
- [Job](#)
- [JobStatus](#)
- [ModelService](#)
- [PredictInput](#)

删除数据结构：

- Service
- ServiceConfig

修改数据结构：

- [ServiceStatus](#)
 - 新增成员：Message

第 1 次发布

发布时间：2019-08-27 16:02:42

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateService](#)
- [CreateServiceConfig](#)
- [DeleteService](#)
- [DeleteServiceConfig](#)
- [DescribeRuntimes](#)
- [DescribeServiceConfigs](#)
- [DescribeServices](#)
- [UpdateService](#)

新增数据结构：

- [Conditions](#)
- [Filter](#)
- [Option](#)
- [Runtime](#)
- [Scaler](#)
- Service
- ServiceConfig
- [ServiceStatus](#)

简介

最近更新时间：2021-12-31 11:26:17

欢迎使用 腾讯云 TI 平台 TI-EMS API 3.0 版本。全新的 API 接口文档更加规范和全面，统一的参数风格和公共错误码，统一的 SDK/CLI 版本与 API 文档严格一致，给您带来简单快捷的使用体验。支持全地域就近接入让您更快连接腾讯云产品。更多腾讯云 API 3.0 使用介绍请查看：[快速入门](#)

腾讯云 TI 平台 TI-EMS 是具备虚拟化异构算力和弹性扩缩容能力的在线推理平台，能够帮助客户解决模型部署复杂、资源浪费、手工扩展资源效率低下的问题。客户通过使用 TI-EMS 可以实现模型一键部署，自动调整弹性计算资源。同时，TI-EMS 具备多模型支持、版本管理和灰度升级等丰富完善的功能，其内置的 CPU/GPU 推理加速镜像为客户提供高性能、高性价比推理服务。

API 概览

最近更新时间：2020-11-12 08:08:44

服务管理相关接口

接口名称	接口功能
CreateService	创建服务
DeleteService	删除服务
DescribeServices	描述服务
ExposeService	暴露服务
UpdateService	更新服务

资源管理相关接口

接口名称	接口功能
CreateRsgAsGroup	创建资源组的伸缩组
DeleteInstance	删除节点
DeleteResourceGroup	删除资源组
DeleteRsgAsGroup	删除资源组的伸缩组
DescribeInstances	获取节点列表
DescribeResourceGroups	获取资源组列表
DescribeRsgAsGroupActivities	查询伸缩组活动
DescribeRsgAsGroups	查询资源组的伸缩组信息
DisableRsgAsGroup	停用资源组的伸缩组
EnableRsgAsGroup	启用资源组的伸缩组
UpdateRsgAsGroup	更新资源组的伸缩组

其他接口

接口名称	接口功能
CreateRuntime	创建运行环境
DeleteRuntime	删除运行环境
DescribeRuntimes	描述服务运行环境

任务管理相关接口

接口名称	接口功能
------	------

接口名称	接口功能
CreateJob	创建任务
DeleteJob	删除任务
UpdateJob	更新任务

配置管理相关接口

接口名称	接口功能
CreateServiceConfig	创建服务配置
DeleteServiceConfig	删除服务配置
DescribeServiceConfigs	描述服务配置

调用方式

请求结构

最近更新时间：2020-09-28 08:08:33

1. 服务地址

API 支持就近地域接入，本产品就近地域接入域名为 tiems.tencentcloudapi.com，也支持指定地域域名访问，例如广州地域的域名为 tiems.ap-guangzhou.tencentcloudapi.com。

推荐使用就近地域接入域名。根据调用接口时客户端所在位置，会自动解析到最近的某个具体地域的服务器。例如在广州发起请求，会自动解析到广州的服务器，效果和指定 tiems.ap-guangzhou.tencentcloudapi.com 是一致的。

注意：对时延敏感的业务，建议指定带地域的域名。

注意：域名是 API 的接入点，并不代表产品或者接口实际提供服务的地域。产品支持的地域列表请在调用方式/公共参数文档中查阅，接口支持的地域请在接口文档输入参数中查阅。

目前支持的域名列表为：

接入地域	域名
就近地域接入（推荐，只支持非金融区）	tiems.tencentcloudapi.com
华南地区(广州)	tiems.ap-guangzhou.tencentcloudapi.com
华东地区(上海)	tiems.ap-shanghai.tencentcloudapi.com
华北地区(北京)	tiems.ap-beijing.tencentcloudapi.com
西南地区(成都)	tiems.ap-chengdu.tencentcloudapi.com
西南地区(重庆)	tiems.ap-chongqing.tencentcloudapi.com
港澳台地区(中国香港)	tiems.ap-hongkong.tencentcloudapi.com
亚太东南(新加坡)	tiems.ap-singapore.tencentcloudapi.com
亚太东南(曼谷)	tiems.ap-bangkok.tencentcloudapi.com
亚太南部(孟买)	tiems.ap-mumbai.tencentcloudapi.com
亚太东北(首尔)	tiems.ap-seoul.tencentcloudapi.com
亚太东北(东京)	tiems.ap-tokyo.tencentcloudapi.com
美国东部(弗吉尼亚)	tiems.na-ashburn.tencentcloudapi.com
美国西部(硅谷)	tiems.na-siliconvalley.tencentcloudapi.com
北美地区(多伦多)	tiems.na-toronto.tencentcloudapi.com
欧洲地区(法兰克福)	tiems.eu-frankfurt.tencentcloudapi.com
欧洲地区(莫斯科)	tiems.eu-moscow.tencentcloudapi.com

2. 通信协议

腾讯云 API 的所有接口均通过 HTTPS 进行通信，提供高安全性的通信通道。

3. 请求方法

支持的 HTTP 请求方法:

- POST (推荐)
- GET

POST 请求支持的 Content-Type 类型:

- application/json (推荐), 必须使用签名方法 v3 (TC3-HMAC-SHA256)。
- application/x-www-form-urlencoded, 必须使用签名方法 v1 (HmacSHA1 或 HmacSHA256)。
- multipart/form-data (仅部分接口支持), 必须使用签名方法 v3 (TC3-HMAC-SHA256)。

GET 请求的请求包大小不得超过32KB。POST 请求使用签名方法 v1 (HmacSHA1、HmacSHA256) 时不得超过1MB。POST 请求使用签名方法 v3 (TC3-HMAC-SHA256) 时支持10MB。

4. 字符编码

均使用 UTF-8 编码。

公共参数

最近更新时间：2022-04-12 06:22:55

公共参数是用于标识用户和接口签名的参数，如非必要，在每个接口单独的接口文档中不再对这些参数进行说明，但每次请求均需要携带这些参数，才能正常发起请求。

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

签名方法 v3（有时也称作 TC3-HMAC-SHA256）相比签名方法 v1（有些文档可能会简称签名方法），更安全，支持更大的请求包，支持 POST JSON 格式，性能有一定提升，推荐使用该签名方法计算签名。完整介绍详见 [签名方法 v3](#)。

注意：接口文档中的示例由于目的是展示接口参数用法，简化起见，使用的是签名方法 v1 GET 请求，如果依旧想使用签名方法 v1 请参考下文章节。

使用签名方法 v3 时，公共参数需要统一放到 HTTP Header 请求头部中，如下表所示：

参数名称	类型	必选	描述
Action	String	是	HTTP 请求头：X-TC-Action。操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	HTTP 请求头：X-TC-Region。地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	HTTP 请求头：X-TC-Timestamp。当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。 注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
Version	String	是	HTTP 请求头：X-TC-Version。操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，通常为域名前缀，例如域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 tiems； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要，计算过程详见 文档 。
Token	String	否	HTTP 请求头：X-TC-Token。即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	HTTP 请求头：X-TC-Language。指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表中的前十个，接口参数设置为偏移量 Offset=0，返回数量 Limit=10，则其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Limit=10&Offset=0
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKID*****EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: ap-guangzhou
```

HTTP POST (application/json) 请求结构示例:

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKID*****EXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

{"Offset":0,"Limit":10}
```

HTTP POST (multipart/form-data) 请求结构示例 (仅特定的接口支持) :

```
https://cvm.tencentcloudapi.com/

Authorization: TC3-HMAC-SHA256 Credential=AKID*****EXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou

--58731222010402
Content-Disposition: form-data; name="Offset"

0
--58731222010402
Content-Disposition: form-data; name="Limit"

10
--58731222010402--
```

使用签名方法 v1 的公共参数

使用签名方法 v1（有时会称作 HmacSHA256 和 HmacSHA1），公共参数需要统一放到请求串中，完整介绍详见[文档](#)

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见 文档 。
Version	String	是	操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表，其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****EXAMPLE
```

Host: cvm.tencentcloudapi.com

Content-Type: application/x-www-form-urlencoded

HTTP POST 请求结构示例：

```
https://cvm.tencentcloudapi.com/
```

Host: cvm.tencentcloudapi.com

Content-Type: application/x-www-form-urlencoded

```
Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****EXAMPLE
```

签名方法 v3

最近更新时间：2022-04-18 06:19:47

以下文档说明了签名方法 v3 的签名过程，但仅在您编写自己的代码来调用腾讯云 API 时才有用。我们推荐您使用 [腾讯云 API Explorer](#)，[腾讯云 SDK](#) 和 [腾讯云命令行工具（TCCLI）](#) 等开发者工具，从而无需学习如何对 API 请求进行签名。

推荐使用 API Explorer

</> 点击调试

您可以通过 API Explorer 的【签名串生成】模块查看每个接口签名的生成过程。

腾讯云 API 会对每个请求进行身份验证，用户需要使用安全凭证，经过特定的步骤对请求进行签名（Signature），每个请求都需要在公共参数中指定该签名结果并以指定的方式和格式发送请求。

为什么要进行签名

签名通过以下方式帮助保护请求：

1. 验证请求者的身份

签名确保请求是由持有有效访问密钥的人发送的。请参阅控制台 [云 API 密钥](#) 页面获取密钥相关信息。

2. 保护传输中的数据

为了防止请求在传输过程中被篡改，腾讯云 API 会使用请求参数来计算请求的哈希值，并将生成的哈希值加密后作为请求的一部分，发送到腾讯云 API 服务器。服务器会使用收到的请求参数以同样的过程计算哈希值，并验证请求中的哈希值。如果请求被篡改，将导致哈希值不一致，腾讯云 API 将拒绝本次请求。

签名方法 v3（TC3-HMAC-SHA256）功能上覆盖了以前的签名方法 v1，而且更安全，支持更大的请求，支持 JSON 格式，POST 请求支持传空数组和空字符串，性能有一定提升，推荐使用该签名方法计算签名。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v3”，可以生成签名过程进行验证，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 7 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)。

申请安全凭证

本文使用的安全凭证为密钥，密钥包括 SecretId 和 SecretKey。每个用户最多可以拥有两对密钥。

- SecretId：用于标识 API 调用者身份，可以简单类比为用户名。
- SecretKey：用于验证 API 调用者的身份，可以简单类比为密码。
- 用户必须严格保管安全凭证，避免泄露，否则将危及财产安全。如已泄露，请立刻禁用该安全凭证。

申请安全凭证的具体步骤如下：

1. 登录 [腾讯云管理中心控制台](#)。
2. 前往 [云API密钥](#) 的控制台页面。
3. 在 [云API密钥](#) 页面，单击【新建密钥】即可以创建一对密钥。

签名版本 v3 签名过程

云 API 支持 GET 和 POST 请求。对于 GET 方法，只支持 Content-Type: application/x-www-form-urlencoded 协议格式。对于 POST 方法，目前支持 Content-Type: application/json 以及 Content-Type: multipart/form-data 两种协议格式，json 格式绝大多数接口均支持，multipart 格式只有特定接口支持，此时该接口不能使用 json 格式调用，参考具体业务接口文档说明。推荐使用 POST 请求，因为两者的结果并无差异，但 GET 请求只支持 32 KB 以内的请求包。

下面以云服务器查询广州实例列表作为例子，分步骤介绍签名的计算过程。我们选择该接口是因为：

1. 云服务器默认已开通，该接口很常用；
2. 该接口是只读的，不会改变现有资源的状态；
3. 接口覆盖的参数种类较全，可以演示包含数据结构的数组如何使用。

在示例中，不论公共参数或者接口的参数，我们尽量选择容易犯错的情况。在实际调用接口时，请根据实际情况来，每个接口的参数并不相同，不要照抄这个例子的参数和值。此外，这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数（在 HTTP 头部设置，添加 X-TC- 前缀）。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsj5yKBZQpn74WFkmLPx3***** 和 Gu5t9xGARNpq86cd98joQYCN3*****。用户想查看广州区云服务器名为“未命名”的主机状态，只返回一条数据。则请求可能为：

```
curl -X POST https://cvm.tencentcloudapi.com \
-H "Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Host: cvm.tencentcloudapi.com" \
-H "X-TC-Action: DescribeInstances" \
-H "X-TC-Timestamp: 1551113065" \
-H "X-TC-Version: 2017-03-12" \
-H "X-TC-Region: ap-guangzhou" \
-d '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
```

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

```
CanonicalRequest =
HTTPRequestMethod + '\n' +
CanonicalURI + '\n' +
CanonicalQueryString + '\n' +
CanonicalHeaders + '\n' +
SignedHeaders + '\n' +
HashedRequestPayload
```

字段名称	解释
HTTPRequestMethod	HTTP 请求方法（GET、POST）。此示例取值为 POST。
CanonicalURI	URI 参数，API 3.0 固定为正斜杠（/）。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串""，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，例如：Limit=10&Offset=0。 注意：CanonicalQueryString 需要参考 RFC3986 进行 URLEncode，字符集 UTF8，推荐使用编程语言标准库，所有特殊字符均需编码，大写形式。

字段名称	解释
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。 拼接规则： 1. 头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接； 2. 多个头部，按照头部 key（小写）的 ASCII 升序进行拼接。 此示例计算结果是 content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\n。 注意：content-type 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 charset 值，如果签名时和发送时不一致，服务器会返回签名校验失败。
SignedHeaders	参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。 拼接规则： 1. 头部 key 统一转成小写； 2. 多个头部 key（小写）按照 ASCII 升序进行拼接，并且以分号（;）分隔。 此示例为 content-type;host
HashedRequestPayload	请求正文（payload，即 body，此示例为 {"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}）的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。对于 GET 请求，RequestPayload 固定为空字符串。此示例计算结果是 35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064。

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

content-type:application/json; charset=utf-8
host:cvm.tencentcloudapi.com

content-type;host
35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064
```

2. 拼接待签名字符串

按如下格式拼接待签名字符串：

```
StringToSign =
Algorithm + \n +
RequestTimestamp + \n +
CredentialScope + \n +
HashedCanonicalRequest
```

字段名称	解释
Algorithm	签名算法，目前固定为 TC3-HMAC-SHA256。
RequestTimestamp	请求时间戳，即请求头部的公共参数 X-TC-Timestamp 取值，取当前时间 UNIX 时间戳，精确到秒。此示例取值为 1551113065。

字段名称	解释
CredentialScope	凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致。此示例计算结果是 2019-02-25/cvm/tc3_request。
HashedCanonicalRequest	前述步骤拼接所得规范请求串的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。此示例计算结果是 5ffe6a04c0664d6b969fab9a13bdab201d63ee709638e2749d62a09ca18d7031。

注意：

1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨时调用必定失败。假设时间戳为 1551113065，在东八区的时间是 2019-02-26 00:44:25，但是计算得到的 Date 取 UTC+0 的日期应为 2019-02-25，而不是 2019-02-26。
2. Timestamp 必须是当前系统时间，且需确保系统时间和标准时间是同步的，如果相差超过五分钟则必定失败。如果长时间不和标准时间同步，可能导致运行一段时间后，请求必定失败，返回签名过期错误。

根据以上规则，示例中得到的待签名字符串如下：

```
TC3-HMAC-SHA256
1551113065
2019-02-25/cvm/tc3_request
5ffe6a04c0664d6b969fab9a13bdab201d63ee709638e2749d62a09ca18d7031
```

3. 计算签名

1) 计算派生签名密钥，伪代码如下：

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3*****"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

派生出的密钥 SecretDate、SecretService 和 SecretSigning 是二进制的数，可能包含不可打印字符，此处不展示中间结果。

请注意，不同的编程语言，HMAC 库函数中参数顺序可能不一样，请以实际情况为准。此处的伪代码密钥参数 key 在前，消息参数 data 在后。通常标准库函数会提供二进制格式的返回值，也可能会提供打印友好的十六进制格式的返回值，此处使用的是二进制格式。

字段名称	解释
SecretKey	原始的 SecretKey，即 Gu5t9xGARNpq86cd98joQYCN3*****。
Date	即 Credential 中的 Date 字段信息。此示例取值为 2019-02-25。
Service	即 Credential 中的 Service 字段信息。此示例取值为 cvm。

2) 计算签名，伪代码如下：

```
Signature = HexEncode(HMAC_SHA256(SecretSigning, StringToSign))
```

此示例计算结果是 2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c。

4. 拼接 Authorization

按如下格式拼接 Authorization:

```
Authorization =  
Algorithm + ' ' +  
'Credential=' + SecretId + '/' + CredentialScope + ', ' +  
'SignedHeaders=' + SignedHeaders + ', ' +  
'Signature=' + Signature
```

字段名称	解释
Algorithm	签名方法，固定为 TC3-HMAC-SHA256。
SecretId	密钥对中的 SecretId，即 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****。
CredentialScope	见上文，凭证范围。此示例计算结果是 2019-02-25/cvm/tc3_request。
SignedHeaders	见上文，参与签名的头部信息。此示例取值为 content-type;host。
Signature	签名值。此示例计算结果是 2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/  
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host, Signature=2230eefd229f582d8b1b891af7107b91597240707d778ab3738f756258d7652c  
Content-Type: application/json; charset=utf-8  
Host: cvm.tencentcloudapi.com  
X-TC-Action: DescribeInstances  
X-TC-Version: 2017-03-12  
X-TC-Timestamp: 1551113065  
X-TC-Region: ap-guangzhou  
  
{ "Limit": 1, "Filters": [ { "Values": [ "\u672a\u547d\u540d" ], "Name": "instance-name" } ] }
```

签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)

- [NodeJS](#)
- [.NET](#)
- [C++](#)

下面提供了不同产品的生成签名 demo，您可以找到对应的产品参考签名的生成：

- [Signature Demo](#)

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

Java

```
import java.nio.charset.Charset;
import java.nio.charset.StandardCharsets;
import java.security.MessageDigest;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPITC3Demo {
    private final static Charset UTF8 = StandardCharsets.UTF_8;
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";
    private final static String CT_JSON = "application/json; charset=utf-8";

    public static byte[] hmac256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
        mac.init(secretKeySpec);
        return mac.doFinal(msg.getBytes(UTF8));
    }

    public static String sha256Hex(String s) throws Exception {
        MessageDigest md = MessageDigest.getInstance("SHA-256");
        byte[] d = md.digest(s.getBytes(UTF8));
        return DatatypeConverter.printHexBinary(d).toLowerCase();
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.tencentcloudapi.com";
        String region = "ap-guangzhou";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1551113065";
```

```
//String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
// 注意时区, 否则容易出错
sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

// ***** 步骤 1: 拼接规范请求串 *****
String httpRequestMethod = "POST";
String canonicalUri = "/";
String canonicalQueryString = "";
String canonicalHeaders = "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n";
String signedHeaders = "content-type;host";

String payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}";
String hashedRequestPayload = sha256Hex(payload);
String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

// ***** 步骤 2: 拼接待签名字符串 *****
String credentialScope = date + "/" + service + "/" + "tc3_request";
String hashedCanonicalRequest = sha256Hex(canonicalRequest);
String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
System.out.println(stringToSign);

// ***** 步骤 3: 计算签名 *****
byte[] secretDate = hmac256(("TC3" + SECRET_KEY).getBytes(UTF8), date);
byte[] secretService = hmac256(secretDate, service);
byte[] secretSigning = hmac256(secretService, "tc3_request");
String signature = DatatypeConverter.printHexBinary(hmac256(secretSigning, stringToSign)).toLowerCase();
System.out.println(signature);

// ***** 步骤 4: 拼接 Authorization *****
String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Content-Type", CT_JSON);
headers.put("Host", host);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

StringBuilder sb = new StringBuilder();
sb.append("curl -X POST https://").append(host)
.append(" -H \"Authorization: ").append(authorization).append("\")")
.append(" -H \"Content-Type: application/json; charset=utf-8\"")
.append(" -H \"Host: ").append(host).append("\")")
```

```
.append(" -H \"X-TC-Action: ").append(action).append("\")
.append(" -H \"X-TC-Timestamp: ").append(timestamp).append("\")
.append(" -H \"X-TC-Version: ").append(version).append("\")
.append(" -H \"X-TC-Region: ").append(region).append("\")
.append(" -d ").append(payload).append("");
System.out.println(sb.toString());
}
}
```

Python

```
# -*- coding: utf-8 -*-
import hashlib, hmac, json, os, sys, time
from datetime import datetime

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*****"

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 1, "Filters": [{"Name": "instance-name", "Values": [u"未命名"]}]}

# ***** 步骤 1: 拼接规范请求串 *****
http_request_method = "POST"
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s\nhost:%s\n" % (ct, host)
signed_headers = "content-type;host"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

# ***** 步骤 2: 拼接待签名字符串 *****
credential_scope = date + "/" + service + "/" + "tc3_request"
```

```

hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
str(timestamp) + "\n" +
credential_scope + "\n" +
hashed_canonical_request)
print(string_to_sign)

# ***** 步骤 3: 计算签名 *****
# 计算签名摘要函数
def sign(key, msg):
    return hmac.new(key, msg.encode("utf-8"), hashlib.sha256).digest()
secret_date = sign(("TC3" + secret_key).encode("utf-8"), date)
secret_service = sign(secret_date, service)
secret_signing = sign(secret_service, "tc3_request")
signature = hmac.new(secret_signing, string_to_sign.encode("utf-8"), hashlib.sha256).hexdigest()
print(signature)

# ***** 步骤 4: 拼接 Authorization *****
authorization = (algorithm + " " +
"Credential=" + secret_id + "/" + credential_scope + ", " +
"SignedHeaders=" + signed_headers + ", " +
"Signature=" + signature)
print(authorization)

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"
+ ' -H "Content-Type: application/json; charset=utf-8"
+ ' -H "Host: ' + host + '"
+ ' -H "X-TC-Action: ' + action + '"
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '"
+ ' -H "X-TC-Version: ' + version + '"
+ ' -H "X-TC-Region: ' + region + '"
+ " -d '" + payload + "'")

```

Golang

```

package main

import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/hex"
    "fmt"
    "time"
)

func sha256hex(s string) string {
    b := sha256.Sum256([]byte(s))
    return hex.EncodeToString(b[:])
}

```

```
}

func hmacsha256(s, key string) string {
    hashed := hmac.New(sha256.New, []byte(key))
    hashed.Write([]byte(s))
    return string(hashed.Sum(nil))
}

func main() {
    secretId := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
    secretKey := "Gu5t9xGARNpq86cd98joQYCN3*****"
    host := "cvm.tencentcloudapi.com"
    algorithm := "TC3-HMAC-SHA256"
    service := "cvm"
    version := "2017-03-12"
    action := "DescribeInstances"
    region := "ap-guangzhou"
    //var timestamp int64 = time.Now().Unix()
    var timestamp int64 = 1551113065

    // step 1: build canonical request string
    httpRequestMethod := "POST"
    canonicalURI := "/"
    canonicalQueryString := ""
    canonicalHeaders := "content-type:application/json; charset=utf-8\n" + "host:" + host + "\n"
    signedHeaders := "content-type;host"
    payload := `{ "Limit": 1, "Filters": [{ "Values": [ "\u672a\u547d\u540d" ], "Name": "instance-name" } ]}`
    hashedRequestPayload := sha256hex(payload)
    canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
        httpRequestMethod,
        canonicalURI,
        canonicalQueryString,
        canonicalHeaders,
        signedHeaders,
        hashedRequestPayload)
    fmt.Println(canonicalRequest)

    // step 2: build string to sign
    date := time.Unix(timestamp, 0).UTC().Format("2006-01-02")
    credentialScope := fmt.Sprintf("%s/%s/tc3_request", date, service)
    hashedCanonicalRequest := sha256hex(canonicalRequest)
    string2sign := fmt.Sprintf("%s\n%d\n%s\n%s",
        algorithm,
        timestamp,
        credentialScope,
        hashedCanonicalRequest)
    fmt.Println(string2sign)

    // step 3: sign string
    secretDate := hmacsha256(date, "TC3"+secretKey)
    secretService := hmacsha256(service, secretDate)
```

```
secretSigning := hmacsha256("tc3_request", secretService)
signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))
fmt.Println(signature)

// step 4: build authorization
authorization := fmt.Sprintf("%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
algorithm,
secretId,
credentialScope,
signedHeaders,
signature)
fmt.Println(authorization)

curl := fmt.Sprintf(`curl -X POST https://%s\
-H "Authorization: %s"\
-H "Content-Type: application/json; charset=utf-8"\
-H "Host: %s" -H "X-TC-Action: %s"\
-H "X-TC-Timestamp: %d"\
-H "X-TC-Version: %s"\
-H "X-TC-Region: %s"\
-d '%s'`, host, authorization, host, action, timestamp, version, region, payload)
fmt.Println(curl)
}
```

PHP

```
<?php
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*****";
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

// step 1: build canonical request string
$httpRequestMethod = "POST";
$canonicalUri = "/";
$canonicalQueryString = "";
$canonicalHeaders = "content-type:application/json; charset=utf-8\n".host.".host.\n";
$signedHeaders = "content-type;host";
$payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]';
$hashedRequestPayload = hash("SHA256", $payload);
$canonicalRequest = $httpRequestMethod."\n"
.$canonicalUri."\n"
.$canonicalQueryString."\n"
.$canonicalHeaders."\n"
```

```
.$signedHeaders."\n"
.$hashedRequestPayload;
echo $canonicalRequest.PHP_EOL;

// step 2: build string to sign
$date = gmdate("Y-m-d", $timestamp);
$credentialScope = $date."/".$service."/tc3_request";
$hashedCanonicalRequest = hash("SHA256", $canonicalRequest);
$stringToSign = $algorithm."\n"
.$timestamp."\n"
.$credentialScope."\n"
.$hashedCanonicalRequest;
echo $stringToSign.PHP_EOL;

// step 3: sign string
$secretDate = hash_hmac("SHA256", $date, "TC3".$secretKey, true);
$secretService = hash_hmac("SHA256", $service, $secretDate, true);
$secretSigning = hash_hmac("SHA256", "tc3_request", $secretService, true);
$signature = hash_hmac("SHA256", $stringToSign, $secretSigning);
echo $signature.PHP_EOL;

// step 4: build authorization
$authorization = $algorithm
." Credential=".$secretId."/".$credentialScope
.", SignedHeaders=content-type;host, Signature=".$signature;
echo $authorization.PHP_EOL;

$curl = "curl -X POST https://".$host
.' -H "Authorization: '.$authorization.'"
.' -H "Content-Type: application/json; charset=utf-8"
.' -H "Host: '.$host.'"
.' -H "X-TC-Action: '.$action.'"
.' -H "X-TC-Timestamp: '.$timestamp.'"
.' -H "X-TC-Version: '.$version.'"
.' -H "X-TC-Region: '.$region.'"
." -d '$payload'";
echo $curl.PHP_EOL;
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'digest'
require 'json'
require 'time'
require 'openssl'

# 密钥参数
secret_id = 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****'
secret_key = 'Gu5t9xGARNpq86cd98joQYCN3*****'
```

```
service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

# ***** 步骤 1: 拼接规范请求串 *****
http_request_method = 'POST'
canonical_uri = '/'
canonical_querystring = ''
canonical_headers = "content-type:application/json; charset=utf-8\nhost:#{host}\n"
signed_headers = 'content-type;host'
# params = { 'Limit' => 1, 'Filters' => [{ 'Name' => 'instance-name', 'Values' => ['未命名'] }] }
# payload = JSON.generate(params, { 'ascii_only' => true, 'space' => ' ' })
# json will generate in random order, to get specified result in example, we hard-code it here.
payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
hashed_request_payload = Digest::SHA256.hexdigest(payload)
canonical_request = [
  http_request_method,
  canonical_uri,
  canonical_querystring,
  canonical_headers,
  signed_headers,
  hashed_request_payload,
].join("\n")

puts canonical_request

# ***** 步骤 2: 拼接待签名字符串 *****
credential_scope = date + '/' + service + '/' + 'tc3_request'
hashed_request_payload = Digest::SHA256.hexdigest(canonical_request)
string_to_sign = [
  algorithm,
  timestamp.to_s,
  credential_scope,
  hashed_request_payload,
].join("\n")
puts string_to_sign

# ***** 步骤 3: 计算签名 *****
digest = OpenSSL::Digest.new('sha256')
secret_date = OpenSSL::HMAC.digest(digest, 'TC3' + secret_key, date)
secret_service = OpenSSL::HMAC.digest(digest, secret_date, service)
secret_signing = OpenSSL::HMAC.digest(digest, secret_service, 'tc3_request')
signature = OpenSSL::HMAC.hexdigest(digest, secret_signing, string_to_sign)
```

```
puts signature
```

```
# ***** 步骤 4: 拼接 Authorization *****
```

```
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{signed_headers}, Signature=#{signature}"
```

```
puts authorization
```

```
puts 'curl -X POST ' + endpoint \  
+ ' -H "Authorization: ' + authorization + '" \  
+ ' -H "Content-Type: application/json; charset=utf-8" \  
+ ' -H "Host: ' + host + '" \  
+ ' -H "X-TC-Action: ' + action + '" \  
+ ' -H "X-TC-Timestamp: ' + timestamp.to_s + '" \  
+ ' -H "X-TC-Version: ' + version + '" \  
+ ' -H "X-TC-Region: ' + region + '" \  
+ " -d '" + payload + '"
```

DotNet

```
using System;  
using System.Collections.Generic;  
using System.Security.Cryptography;  
using System.Text;  
  
public class Application  
{  
    public static string SHA256Hex(string s)  
    {  
        using (SHA256 algo = SHA256.Create())  
        {  
            byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));  
            StringBuilder builder = new StringBuilder();  
            for (int i = 0; i < hashbytes.Length; ++i)  
            {  
                builder.Append(hashbytes[i].ToString("x2"));  
            }  
            return builder.ToString();  
        }  
    }  
  
    public static byte[] HmacSHA256(byte[] key, byte[] msg)  
    {  
        using (HMACSHA256 mac = new HMACSHA256(key))  
        {  
            return mac.ComputeHash(msg);  
        }  
    }  
  
    public static Dictionary<String, String> BuildHeaders(string secretid,  
        string secretkey, string service, string endpoint, string region,
```

```
string action, string version, DateTime date, string requestPayload)
{
    string datestr = date.ToString("yyyy-MM-dd");
    DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);
    long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 1000;
    // ***** 步骤 1: 拼接规范请求串 *****
    string algorithm = "TC3-HMAC-SHA256";
    string httpRequestMethod = "POST";
    string canonicalUri = "/";
    string canonicalQueryString = "";
    string contentType = "application/json";
    string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n" + "host:" + endpoint + "\n";
    string signedHeaders = "content-type;host";
    string hashedRequestPayload = SHA256Hex(requestPayload);
    string canonicalRequest = httpRequestMethod + "\n"
        + canonicalUri + "\n"
        + canonicalQueryString + "\n"
        + canonicalHeaders + "\n"
        + signedHeaders + "\n"
        + hashedRequestPayload;
    Console.WriteLine(canonicalRequest);

    // ***** 步骤 2: 拼接待签名字符串 *****
    string credentialScope = datestr + "/" + service + "/" + "tc3_request";
    string hashedCanonicalRequest = SHA256Hex(canonicalRequest);
    string stringToSign = algorithm + "\n" + requestTimestamp.ToString() + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
    Console.WriteLine(stringToSign);

    // ***** 步骤 3: 计算签名 *****
    byte[] tc3SecretKey = Encoding.UTF8.GetBytes("TC3" + secretkey);
    byte[] secretDate = HmacSHA256(tc3SecretKey, Encoding.UTF8.GetBytes(datestr));
    byte[] secretService = HmacSHA256(secretDate, Encoding.UTF8.GetBytes(service));
    byte[] secretSigning = HmacSHA256(secretService, Encoding.UTF8.GetBytes("tc3_request"));
    byte[] signatureBytes = HmacSHA256(secretSigning, Encoding.UTF8.GetBytes(stringToSign));
    string signature = BitConverter.ToString(signatureBytes).Replace("-", "").ToLower();
    Console.WriteLine(signature);

    // ***** 步骤 4: 拼接 Authorization *****
    string authorization = algorithm + " "
        + "Credential=" + secretid + "/" + credentialScope + ", "
        + "SignedHeaders=" + signedHeaders + ", "
        + "Signature=" + signature;
    Console.WriteLine(authorization);

    Dictionary<string, string> headers = new Dictionary<string, string>();
    headers.Add("Authorization", authorization);
    headers.Add("Host", endpoint);
    headers.Add("Content-Type", contentType + "; charset=utf-8");
    headers.Add("X-TC-Timestamp", requestTimestamp.ToString());
    headers.Add("X-TC-Version", version);
}
```

```
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}

public static void Main(string[] args)
{
    // 密钥参数
    string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
    string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";

    string service = "cvm";
    string endpoint = "cvm.tencentcloudapi.com";
    string region = "ap-guangzhou";
    string action = "DescribeInstances";
    string version = "2017-03-12";

    // 此处由于示例规范的原因，采用时间戳2019-02-26 00:44:25，此参数作为示例，如果在项目中，您应当使用：
    // DateTime date = DateTime.UtcNow;
    // 注意时区，建议此时间统一采用UTC时间戳，否则容易出错
    DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
    string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}";

    Dictionary<string, string> headers = BuildHeaders(SECRET_ID, SECRET_KEY, service
, endpoint, region, action, version, date, requestPayload);

    Console.WriteLine("POST https://cvm.tencentcloudapi.com");
    foreach (KeyValuePair<string, string> kv in headers)
    {
        Console.WriteLine(kv.Key + ": " + kv.Value);
    }
    Console.WriteLine();
    Console.WriteLine(requestPayload);
}
}
```

NodeJS

```
const crypto = require('crypto');

function sha256(message, secret = '', encoding) {
    const hmac = crypto.createHmac('sha256', secret)
    return hmac.update(message).digest(encoding)
}

function getHash(message, encoding = 'hex') {
    const hash = crypto.createHash('sha256')
    return hash.update(message).digest(encoding)
}

function getDate(timestamp) {
```

```
const date = new Date(timestamp * 1000)
const year = date.getUTCFullYear()
const month = ('0' + (date.getUTCMonth() + 1)).slice(-2)
const day = ('0' + date.getUTCDate()).slice(-2)
return `${year}-${month}-${day}`
}

function main(){
// 密钥参数
const SECRET_ID = "AKIDz8krbsj5yKBZQpn74WFkmLPx3*****"
const SECRET_KEY = "Gu5t9xGARnpq86cd98joQYCN3*****"

const endpoint = "cvm.tencentcloudapi.com"
const service = "cvm"
const region = "ap-guangzhou"
const action = "DescribeInstances"
const version = "2017-03-12"
//const timestamp = getTime()
const timestamp = 1551113065
//时间处理, 获取世界时间日期
const date = getDate(timestamp)

// ***** 步骤 1: 拼接规范请求串 *****
const signedHeaders = "content-type;host"

const payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}"

const hashedRequestPayload = getHash(payload);
const httpRequestMethod = "POST"
const canonicalUri = "/"
const canonicalQueryString = ""
const canonicalHeaders = "content-type:application/json; charset=utf-8\\n" + "host:" + endpoint + "\\n"

const canonicalRequest = httpRequestMethod + "\\n"
+ canonicalUri + "\\n"
+ canonicalQueryString + "\\n"
+ canonicalHeaders + "\\n"
+ signedHeaders + "\\n"
+ hashedRequestPayload
console.log(canonicalRequest)

// ***** 步骤 2: 拼接待签名字符串 *****
const algorithm = "TC3-HMAC-SHA256"
const hashedCanonicalRequest = getHash(canonicalRequest);
const credentialScope = date + "/" + service + "/" + "tc3_request"
const stringToSign = algorithm + "\\n" +
timestamp + "\\n" +
credentialScope + "\\n" +
hashedCanonicalRequest
console.log(stringToSign)
```

```
// ***** 步骤 3: 计算签名 *****
const kDate = sha256(date, 'TC3' + SECRET_KEY)
const kService = sha256(service, kDate)
const kSigning = sha256('tc3_request', kService)
const signature = sha256(stringToSign, kSigning, 'hex')
console.log(signature)

// ***** 步骤 4: 拼接 Authorization *****
const authorization = algorithm + " " +
"Credential=" + SECRET_ID + "/" + credentialScope + ", " +
"SignedHeaders=" + signedHeaders + ", " +
"Signature=" + signature
console.log(authorization)

const curlcmd = 'curl -X POST ' + "https://" + endpoint
+ ' -H "Authorization: ' + authorization + '"
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + endpoint + '"
+ ' -H "X-TC-Action: ' + action + '"
+ ' -H "X-TC-Timestamp: ' + timestamp.toString() + '"
+ ' -H "X-TC-Version: ' + version + '"
+ ' -H "X-TC-Region: ' + region + '"
+ " -d '" + payload + '"'
console.log(curlcmd)
}
main()
```

C++

```
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <stdio.h>
#include <time.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

using namespace std;

string get_data(int64_t &timestamp)
{
    string utcDate;
    char buff[20] = {0};
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);
    utcDate = string(buff);
    return utcDate;
}
```

```
}

string int2str(int64_t n)
{
    std::stringstream ss;
    ss << n;
    return ss.str();
}

string sha256Hex(const string &str)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str.c_str(), str.size());
    SHA256_Final(hash, &sha256);
    std::string NewString = "";
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        snprintf(buf, sizeof(buf), "%02x", hash[i]);
        NewString = NewString + buf;
    }
    return NewString;
}

string HmacSha256(const string &key, const string &input)
{
    unsigned char hash[32];

    HMAC_CTX *h;
    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX hmac;
    HMAC_CTX_init(&hmac);
    h = &hmac;
    #else
    h = HMAC_CTX_new();
    #endif

    HMAC_Init_ex(h, &key[0], key.length(), EVP_sha256(), NULL);
    HMAC_Update(h, ( unsigned char* )&input[0], input.length());
    unsigned int len = 32;
    HMAC_Final(h, hash, &len);

    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX_cleanup(h);
    #else
    HMAC_CTX_free(h);
    #endif

    std::stringstream ss;
```

```
ss << std::setfill('0');
for (int i = 0; i < len; i++)
{
    ss << hash[i];
}

return (ss.str());
}

string HexEncode(const string &input)
{
    static const char* const lut = "0123456789abcdef";
    size_t len = input.length();

    string output;
    output.reserve(2 * len);
    for (size_t i = 0; i < len; ++i)
    {
        const unsigned char c = input[i];
        output.push_back(lut[c >> 4]);
        output.push_back(lut[c & 15]);
    }
    return output;
}

int main()
{
    // 密钥参数
    string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
    string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";

    string service = "cvm";
    string host = "cvm.tencentcloudapi.com";
    string region = "ap-guangzhou";
    string action = "DescribeInstances";
    string version = "2017-03-12";
    int64_t timestamp = 1551113065;
    string date = get_data(timestamp);

    // ***** 步骤 1: 拼接规范请求串 *****
    string httpRequestMethod = "POST";
    string canonicalUri = "/";
    string canonicalQueryString = "";
    string canonicalHeaders = "content-type:application/json; charset=utf-8\nhost:" + host + "\n";
    string signedHeaders = "content-type;host";
    string payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}] }";
    string hashedRequestPayload = sha256Hex(payload);
    string canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryString + "\n"
    + canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
    cout << canonicalRequest << endl;
```

```
// ***** 步骤 2: 拼接待签名字符串 *****
string algorithm = "TC3-HMAC-SHA256";
string RequestTimestamp = int2str(timestamp);
string credentialScope = date + "/" + service + "/" + "tc3_request";
string hashedCanonicalRequest = sha256Hex(canonicalRequest);
string stringToSign = algorithm + "\n" + RequestTimestamp + "\n" + credentialScope + "\n" + hashedCanonicalRequest;
cout << stringToSign << endl;

// ***** 步骤 3: 计算签名 *****
string kKey = "TC3" + SECRET_KEY;
string kDate = HmacSha256(kKey, date);
string kService = HmacSha256(kDate, service);
string kSigning = HmacSha256(kService, "tc3_request");
string signature = HexEncode(HmacSha256(kSigning, stringToSign));
cout << signature << endl;

// ***** 步骤 4: 拼接 Authorization *****
string authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
cout << authorization << endl;

string curlcmd = "curl -X POST https://" + host + "\n"
+ " -H \"Authorization: \" + authorization + "\n"
+ " -H \"Content-Type: application/json; charset=utf-8\" + "\n"
+ " -H \"Host: \" + host + "\n"
+ " -H \"X-TC-Action: \" + action + "\n"
+ " -H \"X-TC-Timestamp: \" + RequestTimestamp + "\n"
+ " -H \"X-TC-Version: \" + version + "\n"
+ " -H \"X-TC-Region: \" + region + "\n"
+ " -d \"" + payload;
cout << curlcmd << endl;
return 0;
};
```

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

错误码	错误描述
AuthFailure.SignatureExpire	签名过期。Timestamp 与服务器接收到请求的时间相差不得超过五分钟。
AuthFailure.SecretIdNotFound	密钥不存在。请到控制台查看密钥是否被禁用，是否少复制了字符或者多了字符。
AuthFailure.SignatureFailure	签名错误。可能是签名计算错误，或者签名与实际发送的内容不相符合，也有可能是密钥 SecretKey 错误导致的。
AuthFailure.TokenFailure	临时证书 Token 错误。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。

签名方法

最近更新时间：2022-04-12 06:22:56

签名方法 v1 简单易用，但是功能和安全性都不如签名方法 v3，推荐使用签名方法 v3。

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v1”，可以生成签名过程进行验证，并提供了部分编程语言的签名示例，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 7 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)。

推荐使用 API Explorer

<> 点击调试

您可以通过 API Explorer 的【签名串生成】模块查看每个接口签名的生成过程。

腾讯云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。

签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往 [云API密钥页面](#) 申请，否则无法调用云 API 接口。

1. 申请安全凭证

在第一次使用云 API 之前，请前往 [云 API 密钥页面](#) 申请安全凭证。

安全凭证包括 SecretId 和 SecretKey：

- SecretId 用于标识 API 调用者身份
- SecretKey 用于加密签名字符串和服务器端验证签名字符串的密钥。
- 用户必须严格保管安全凭证，避免泄露。

申请安全凭证的具体步骤如下：

- 登录 [腾讯云管理中心控制台](#)。
- 前往 [云 API 密钥](#) 的控制台页面
- 在 [云 API 密钥](#) 页面，单击【新建密钥】即可以创建一对 SecretId/SecretKey。

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

有了安全凭证 SecretId 和 SecretKey 后，就可以生成签名串了。以下是使用签名方法 v1 生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3*****

注意：这里只是示例，请根据用户实际申请的 SecretId 和 SecretKey 进行后续操作！

以云服务器查看实例列表(DescribeInstances)请求为例，当用户调用这一接口时，其请求参数可能如下：

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥 ID	AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886

参数名称	中文	参数值
Region	实例所在区域	ap-guangzhou
InstanceId.0	待查询的实例 ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

这里只展示了部分公共参数和接口输入参数，用户可以根据实际需要添加其他参数，例如 Language 和 Token 公共参数。

2.1. 对参数排序

首先对所有请求参数按参数名的字典序（ASCII 码）升序排序。注意：1）只按参数名进行排序，参数值保持对应即可，不参与比大小；2）按 ASCII 码比大小，如 InstanceIds.2 要排在 InstanceIds.12 后面，不是按字母表，也不是按数值。用户可以借助编程语言中的相关排序函数来实现这一功能，如 PHP 中的 ksort 函数。上述示例参数的排序结果如下：

```
{
  'Action': 'DescribeInstances',
  'InstanceId.0': 'ins-09dx96dg',
  'Limit': 20,
  'Nonce': 11886,
  'Offset': 0,
  'Region': 'ap-guangzhou',
  'SecretId': 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****',
  'Timestamp': 1465185768,
  'Version': '2017-03-12',
}
```

使用其它程序设计语言开发时，可对上面示例中的参数进行排序，得到的结果一致即可。

2.2. 拼接请求字符串

此步骤生成请求字符串。

将把上一步排序好的请求参数格式化成“参数名称=参数值”的形式，如对 Action 参数，其参数名称为 "Action"，参数值为 "DescribeInstances"，因此格式化后即为 Action=DescribeInstances。

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

```
Action=DescribeInstances&InstanceId.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机: 查看实例列表(DescribeInstances)的请求域名为: cvm.tencentcloudapi.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 /。

4. 请求字符串: 即上一步生成的请求字符串。

签名原文串的拼接规则为: 请求方法 + 请求主机 + 请求路径 + ? + 请求字符串。

示例的拼接结果为:

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****&Timestamp=1465185768&Version=2017-03-12
```

2.4. 生成签名串

此步骤生成签名串。

首先使用 HMAC-SHA1 算法对上一步中获得的**签名原文字符串**进行签名, 然后将生成的签名串使用 Base64 进行编码, 即可获得最终的签名串。

具体代码如下, 以 PHP 语言为例:

```
$secretKey = 'Gu5t9xGARNpq86cd98joQYCN3*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****&Timestamp=1465185768&Version=2017-03-12';
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey, true));
echo $signStr;
```

最终得到的签名串为:

```
zmmjn35mikh6pM3V7sUEuX4wyYM=
```

使用其它程序设计语言开发时, 可用上面示例中的原文进行签名验证, 得到的签名串与例子中的一致即可。

3. 签名串编码

生成的签名串并不能直接作为请求参数, 需要对其进行 URL 编码。

如上一步生成的签名串为 zmmjn35mikh6pM3V7sUEuX4wyYM=, 最终得到的签名串请求参数 (Signature) 为: zmmjn35mikh6pM3V7sUEuX4wyYM%3D, 它将用于生成最终的请求 URL。

注意: 如果用户的请求方法是 GET, 或者请求方法为 POST 同时 Content-Type 为 application/x-www-form-urlencoded, 则发送请求时所有请求参数的值均需要做 URL 编码, 参数键和=符号不需要编码。非 ASCII 字符在 URL 编码前需要先用 UTF-8 进行编码。

注意: 有些编程语言的库会自动为所有参数进行 urlencode, 在这种情况下, 就不需要对签名串进行 URL 编码了, 否则两次 URL 编码会导致签名失败。

注意: 其他参数值也需要进行编码, 编码采用 [RFC 3986](#)。使用 %XY 对特殊字符例如汉字进行百分比编码, 其中 “X” 和 “Y” 为十六进制字符 (0-9 和大写字母 A-F), 使用小写将引发错误。

4. 签名失败

根据实际情况, 存在以下签名失败的错误码, 请根据实际情况处理。

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在

错误代码	错误描述
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

在实际调用 API 3.0 时，推荐使用配套的腾讯云 SDK 3.0，SDK 封装了签名的过程，开发时只关注产品提供的具体接口即可。详细信息参见 [SDK 中心](#)。当前支持的编程语言有：

- [Python](#)
- [Java](#)
- [PHP](#)
- [Go](#)
- [NodeJS](#)
- [.NET](#)
- [C++](#)

下面提供了不同产品的生成签名 demo，您可以找到对应的产品参考签名的生成：

- [Signature Demo](#)

为了更清楚的解释签名过程，下面以实际编程语言为例，将上述的签名过程具体实现。请求的域名、调用的接口和参数的取值都以上述签名过程为准，代码只为解释签名过程，并不具备通用性，实际开发请尽量使用 SDK。

最终输出的 url 可能为：https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&Signature=zmmjn35mikh6pM3V7sUEuX4wyYM%3D&Timestamp=1465185768&Version=2017-03-12。

注意：由于示例中的密钥是虚构的，时间戳也不是系统当前时间，因此如果将此 url 在浏览器中打开或者用 curl 等命令调用时会返回鉴权错误：签名过期。为了得到一个可以正常返回的 url，需要修改示例中的 SecretId 和 SecretKey 为真实的密钥，并使用系统当前时间戳作为 Timestamp。

注意：在下面的示例中，不同编程语言，甚至同一语言每次执行得到的 url 可能都有所不同，表现为参数的顺序不同，但这并不影响正确性。只要所有参数都在，且签名计算正确即可。

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，即使是旧版的 API，由于存在细节差异也会导致签名计算错误，请以对应的实际文档为准。

Java

```
import java.io.UnsupportedEncodingException;
import java.net.URLEncoder;
import java.util.Random;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.xml.bind.DatatypeConverter;

public class TencentCloudAPIDemo {
    private final static String CHARSET = "UTF-8";
```

```
public static String sign(String s, String key, String method) throws Exception {
    Mac mac = Mac.getInstance(method);
    SecretKeySpec secretKeySpec = new SecretKeySpec(key.getBytes(CHARSET), mac.getAlgorithm());
    mac.init(secretKeySpec);
    byte[] hash = mac.doFinal(s.getBytes(CHARSET));
    return DatatypeConverter.printBase64Binary(hash);
}

public static String getStringToSign(TreeMap<String, Object> params) {
    StringBuilder s2s = new StringBuilder("GETcvm.tencentcloudapi.com/?");
    // 签名时要求对参数进行字典排序, 此处用TreeMap保证顺序
    for (String k : params.keySet()) {
        s2s.append(k).append("=").append(params.get(k).toString()).append("&");
    }
    return s2s.toString().substring(0, s2s.length() - 1);
}

public static String getUrl(TreeMap<String, Object> params) throws UnsupportedEncodingException {
    StringBuilder url = new StringBuilder("https://cvm.tencentcloudapi.com/?");
    // 实际请求的url中对参数顺序没有要求
    for (String k : params.keySet()) {
        // 需要对请求串进行urlencode, 由于key都是英文字母, 故此处仅对其value进行urlencode
        url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
    }
    return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
    // 实际调用时应当使用随机数, 例如: params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
    params.put("Nonce", 11886); // 公共参数
    // 实际调用时应当使用系统当前时间, 例如: params.put("Timestamp", System.currentTimeMillis() / 1000);
    params.put("Timestamp", 1465185768); // 公共参数
    params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"); // 公共参数
    params.put("Action", "DescribeInstances"); // 公共参数
    params.put("Version", "2017-03-12"); // 公共参数
    params.put("Region", "ap-guangzhou"); // 公共参数
    params.put("Limit", 20); // 业务参数
    params.put("Offset", 0); // 业务参数
    params.put("InstanceId", "ins-09dx96dg"); // 业务参数
    params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3*****", "HmacSHA1")); // 公共参数
    System.out.println(getUrl(params));
}
}
```

Python

注意: 如果是在 Python 2 环境中运行, 需要先安装 requests 依赖包: `pip install requests`。

```
# -*- coding: utf8 -*-
import base64
```

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*****"

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.tencentcloudapi.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceId.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'ap-guangzhou',
        'SecretId': secret_id,
        'Timestamp': 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

Golang

```
package main

import (
    "bytes"
    "crypto/hmac"
    "crypto/sha1"
    "encoding/base64"
    "fmt"
    "sort"
    "strconv"
```

```
)

func main() {
secretId := "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
secretKey := "Gu5t9xGARNpq86cd98joQYCN3*****"
params := map[string]string{
"Nonce": "11886",
"Timestamp": strconv.Itoa(1465185768),
"Region": "ap-guangzhou",
"SecretId": secretId,
"Version": "2017-03-12",
"Action": "DescribeInstances",
"InstanceId.0": "ins-09dx96dg",
"Limit": strconv.Itoa(20),
"Offset": strconv.Itoa(0),
}

var buf bytes.Buffer
buf.WriteString("GET")
buf.WriteString("cvm.tencentcloudapi.com")
buf.WriteString("/")
buf.WriteString("?")

// sort keys by ascii asc order
keys := make([]string, 0, len(params))
for k, _ := range params {
keys = append(keys, k)
}
sort.Strings(keys)

for i := range keys {
k := keys[i]
buf.WriteString(k)
buf.WriteString("=")
buf.WriteString(params[k])
buf.WriteString("&")
}
buf.Truncate(buf.Len() - 1)

hashed := hmac.New(sha1.New, []byte(secretKey))
hashed.Write(buf.Bytes())

fmt.Println(base64.StdEncoding.EncodeToString(hashed.Sum(nil)))
}
```

PHP

```
<?php
$secretId = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
$secretKey = "Gu5t9xGARNpq86cd98joQYCN3*****";
```

```
$param["Nonce"] = 11886;//rand();
$param["Timestamp"] = 1465185768;//time();
$param["Region"] = "ap-guangzhou";
$param["SecretId"] = $secretId;
$param["Version"] = "2017-03-12";
$param["Action"] = "DescribeInstances";
$param["InstanceId.0"] = "ins-09dx96dg";
$param["Limit"] = 20;
$param["Offset"] = 0;

ksort($param);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $param as $key => $value ) {
    $signStr = $signStr . $key . "=" . $value . "&";
}
$signStr = substr($signStr, 0, -1);

$signature = base64_encode(hash_hmac("sha1", $signStr, $secretKey, true));
echo $signature.PHP_EOL;
// need to install and enable curl extension in php.ini
// $param["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($param);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
require 'time'
require 'openssl'
require 'base64'

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3*****"

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
  'Action' => 'DescribeInstances',
  'InstanceId.0' => 'ins-09dx96dg',
  'Limit' => 20,
  'Nonce' => 11886,
  'Offset' => 0,
  'Region' => 'ap-guangzhou',
```

```
'SecretId' => secret_id,
'Timestamp' => 1465185768, # Time.now.to_i
'Version' => '2017-03-12',
}
sign = method + endpoint + '/'?
params = []
data.sort.each do |item|
  params << "#{item[0]}=#{item[1]}"
end
sign += params.join('&')
digest = OpenSSL::Digest.new('sha1')
data['Signature'] = Base64.encode64(OpenSSL::HMAC.digest(digest, secret_key, sign))
puts data['Signature']

# require 'net/http'
# uri = URI('https://' + endpoint)
# uri.query = URI.encode_www_form(data)
# p uri
# res = Net::HTTP.get_response(uri)
# puts res.body
```

DotNet

```
using System;
using System.Collections.Generic;
using System.Net;
using System.Security.Cryptography;
using System.Text;

public class Application {
  public static string Sign(string signKey, string secret)
  {
    string signRet = string.Empty;
    using (HMACSHA1 mac = new HMACSHA1(Encoding.UTF8.GetBytes(signKey)))
    {
      byte[] hash = mac.ComputeHash(Encoding.UTF8.GetBytes(secret));
      signRet = Convert.ToBase64String(hash);
    }
    return signRet;
  }

  public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, string requestHost, string requestPath)
  {
    string retStr = "";
    retStr += requestMethod;
    retStr += requestHost;
    retStr += requestPath;
    retStr += "?";
    string v = "";
    foreach (string key in requestParams.Keys)
```

```
{
v += string.Format("{0}={1}&", key, requestParams[key]);
}
retStr += v.TrimEnd('&');
return retStr;
}

public static void Main(string[] args)
{
// 密钥参数
string SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****";
string SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****";

string endpoint = "cvm.tencentcloudapi.com";
string region = "ap-guangzhou";
string action = "DescribeInstances";
string version = "2017-03-12";
double RequestTimestamp = 1465185768; // 时间戳 2019-02-26 00:44:25,此参数作为示例，以实际为准
// long timestamp = ToTimestamp() / 1000;
// string requestTimestamp = timestamp.ToString();
Dictionary<string, string> param = new Dictionary<string, string>();
param.Add("Limit", "20");
param.Add("Offset", "0");
param.Add("InstanceId.0", "ins-09dx96dg");
param.Add("Action", action);
param.Add("Nonce", "11886");
// param.Add("Nonce", Math.Abs(new Random().Next()).ToString());

param.Add("Timestamp", RequestTimestamp.ToString());
param.Add("Version", version);

param.Add("SecretId", SECRET_ID);
param.Add("Region", region);
SortedDictionary<string, string> headers = new SortedDictionary<string, string>(param, StringComparer.Ordinal);
string sigInParam = MakeSignPlainText(headers, "GET", endpoint, "/");
string sigOutParam = Sign(SECRET_KEY, sigInParam);
Console.WriteLine(sigOutParam);
}
}
```

NodeJS

```
const crypto = require('crypto');

function get_req_url(params, endpoint){
params['Signature'] = escape(params['Signature']);
const url_strParam = sort_params(params)
return "https://" + endpoint + "?" + url_strParam.slice(1);
}
```

```
function formatSignString(reqMethod, endpoint, path, strParam){
  let strSign = reqMethod + endpoint + path + "?" + strParam.slice(1);
  return strSign;
}

function sha1(secretKey, strsign){
  let signMethodMap = {'HmacSHA1': 'sha1'};
  let hmac = crypto.createHmac(signMethodMap['HmacSHA1'], secretKey || "");
  return hmac.update(Buffer.from(strsign, 'utf8')).digest('base64')
}

function sort_params(params){
  let strParam = "";
  let keys = Object.keys(params);
  keys.sort();
  for (let k in keys) {
    //k = k.replace(/_/g, '.');
    strParam += ("&" + keys[k] + "=" + params[keys[k]]);
  }
  return strParam
}

function main(){
  // 密钥参数
  const SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****"
  const SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3*****"

  const endpoint = "cvm.tencentcloudapi.com"
  const Region = "ap-guangzhou"
  const Version = "2017-03-12"
  const Action = "DescribeInstances"
  const Timestamp = 1465185768 // 时间戳 2016-06-06 12:02:48, 此参数作为示例, 以实际为准
  // const Timestamp = Math.round(Date.now() / 1000)
  const Nonce = 11886 // 随机正整数
  //const nonce = Math.round(Math.random() * 65535)

  let params = {};
  params['Action'] = Action;
  params['InstanceId.0'] = 'ins-09dx96dg';
  params['Limit'] = 20;
  params['Offset'] = 0;
  params['Nonce'] = Nonce;
  params['Region'] = Region;
  params['SecretId'] = SECRET_ID;
  params['Timestamp'] = Timestamp;
  params['Version'] = Version;

  // 1. 对参数排序,并拼接请求字符串
  strParam = sort_params(params)

  // 2. 拼接签名原字符串
  const reqMethod = "GET";
```

```
const path = "/";
strSign = formatSignString(reqMethod, endpoint, path, strParam)
// console.log(strSign)

// 3. 生成签名串
params['Signature'] = sha1(SECRET_KEY, strSign)
console.log(params['Signature'])

// 4. 进行url编码并拼接请求url
// const req_url = get_req_url(params, endpoint)
// console.log(params['Signature'])
// console.log(req_url)
}
main()
```

返回结果

最近更新时间：2021-07-20 08:12:49

注意：目前只要请求被服务端正常处理了，响应的 HTTP 状态码均为200。例如返回的消息体里的错误码是签名失败，但 HTTP 状态码是200，而不是401。

正确返回结果

以云服务器的接口查看实例状态列表 (DescribeInstancesStatus) 2017-03-12 版本为例，若调用成功，其可能的返回如下为：

```
{
  "Response": {
    "TotalCount": 0,
    "InstanceStatusSet": [],
    "RequestId": "b5b41468-520d-4192-b42f-595cc34b6c1c"
  }
}
```

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。
- 除了固定的字段外，其余均为具体接口定义的字段，不同的接口所返回的字段参见接口文档中的定义。此例中的 TotalCount 和 InstanceStatusSet 均为 DescribeInstancesStatus 接口定义的字段，由于调用请求的用户暂时还没有云服务器实例，因此 TotalCount 在此情况下的返回值为 0，InstanceStatusSet 列表为空。

错误返回结果

若调用失败，其返回值示例如下为：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码。完整的错误码列表请参考本产品“API 文档”目录下的“错误码”页面。

服务管理相关接口

创建服务

最近更新时间：2022-04-13 07:17:21

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

创建服务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateService。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Scaler	是	Scaler	扩缩容配置
ServiceConfigId	是	String	服务配置Id
Name	是	String	服务名称
ScaleMode	是	String	扩缩容方式，支持AUTO, MANUAL，分别表示自动扩缩容和手动扩缩容
ResourceGroupId	是	String	部署要使用的资源组Id，默认为共享资源组
Cpu	是	Integer	处理器配置, 单位为1/1000核；范围[100, 256000]
Memory	是	Integer	内存配置, 单位为1M；范围[100, 256000]
Cluster	否	String	集群，不填则使用默认集群
Authentication	否	String	默认为空，表示不需要鉴权，TOKEN 表示选择 Token 鉴权方式
Gpu	否	Integer	GPU算力配置，单位为1/1000 卡，范围 [0, 256000]
GpuMemory	否	Integer	显存配置, 单位为1M，范围 [0, 256000]
Description	否	String	备注
GpuType	否	String	GPU类型
LogTopicId	否	String	Cls日志主题ID

3. 输出参数

参数名称	类型	描述
Service	ModelService	服务
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建服务

输入示例

```
https://tiems.tencentcloudapi.com/?Action=CreateService
&ServiceConfigId=asjqn7wtdgrd7gt
&Name=test
&ScaleMode=MANUAL
&ResourceGroupId=ap-beijing
&Cpu=100
&Memory=100
&Description=test
&Scaler.StartReplicas=1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "48d37fb1-15fa-4320-9c59-d673c653e0d4",
    "Service": {
      "Id": "ank95gbm4dwfhmds",
      "Cluster": "ap-beijing",
      "Region": "ap-beijing",
      "Name": "test",
      "Runtime": "pmm1",
      "ModelUri": "cos://ti-ems-12502019.cos.ap-beijing.myqcloud.com/models/pmm1/",
      "Cpu": 100,
      "Memory": 100,
      "Gpu": 0,
      "GpuMemory": 0,
      "CreateTime": "2019-11-07T15:14:19+08:00",
      "UpdateTime": "2019-11-07T15:14:19+08:00",
      "ScaleMode": "MANUAL",
      "Scaler": {
        "MaxReplicas": 1,
        "MinReplicas": 1,
        "StartReplicas": 1,
        "HpaMetrics": []
      },
      "Status": {
```

```
"DesiredReplicas": 0,
"CurrentReplicas": 1,
"Replicas": [],
"Status": "Waiting",
"Message": "",
"Conditions": []
},
"AccessToken": "",
"ConfigName": "test1",
"ConfigVersion": "1.0",
"ServeSeconds": 1,
"ConfigId": "asjqn7wtdgrd7gt",
"ResourceGroupId": "ap-beijing",
"ResourceGroupName": "公共资源组",
"Exposes": [],
"Description": "test",
"GpuType": ""
}
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
FailedOperation.AlreadyExists	已经存在同名资源。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

删除服务

最近更新时间：2022-04-13 07:17:20

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

删除服务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteService。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ServiceId	是	String	服务Id

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除服务

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DeleteServiceConfig
&ServiceId=xxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "bb70a498-78e1-4905-bc73-170b66edabb9"
  }
}
```

```
}  
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

更新服务

最近更新时间：2022-04-13 07:17:20

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

更新服务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：UpdateService。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ServiceId	是	String	服务Id
Scaler	否	Scaler	扩缩容配置
ServiceConfigId	否	String	服务配置Id
ScaleMode	否	String	支持AUTO, MANUAL，分别表示自动扩缩容，手动扩缩容
ServiceAction	否	String	支持STOP(停止) RESUME(重启)
Description	否	String	备注
GpuType	否	String	GPU卡类型
Cpu	否	Integer	处理器配置，单位为 1/1000 核
Memory	否	Integer	内存配置，单位为1M
Gpu	否	Integer	显卡配置，单位为 1/1000 卡
LogTopicId	否	String	Cls日志主题ID

3. 输出参数

参数名称	类型	描述
Service	ModelService	服务
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 更新服务

输入示例

```
https://tiems.tencentcloudapi.com/?Action=UpdateService
&ServiceId=ank95gbm4dwfhmds
&Scaler.StartReplicas=2
&Scaler.MaxReplicas=2
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "9c7e1137-91a3-4c3b-8ff8-d695276a9b13",
    "Service": {
      "Id": "ank95gbm4dwfhmds",
      "Cluster": "ap-beijing",
      "Region": "ap-beijing",
      "Name": "test",
      "Runtime": "pmml",
      "ModelUri": "cos://ti-ems-125502019.cos.ap-beijing.myqcloud.com/models/pmml/",
      "Cpu": 100,
      "Memory": 100,
      "Gpu": 0,
      "GpuMemory": 0,
      "CreateTime": "2019-11-07T15:14:19+08:00",
      "UpdateTime": "2019-11-07T15:14:19+08:00",
      "ScaleMode": "MANUAL",
      "Scaler": {
        "MaxReplicas": 2,
        "MinReplicas": 0,
        "StartReplicas": 2,
        "HpaMetrics": []
      },
      "Status": {
        "DesiredReplicas": 1,
        "CurrentReplicas": 1,
        "Replicas": [
          "ank95gbm4dwfhmds-fccc99cdd-ftff5"
        ],
        "Status": "Normal",
        "Message": "",
        "Conditions": []
      },
      "AccessToken": "",
      "ConfigName": "test1",
      "ConfigVersion": "1.0",
      "ServeSeconds": 307,
```

```
"ConfigId": "asjcqn7wtdgrd7gt",
"ResourceGroupId": "ap-beijing",
"ResourceGroupName": "公共资源组",
"Exposes": [],
"Description": "test",
"GpuType": ""
}
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

描述服务

最近更新时间：2022-04-13 07:17:20

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

描述服务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeServices。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Filters.N	否	Array of Filter	筛选选项，支持筛选的字段：id, region, zone, cluster, status, runtime, rsg_id
Offset	否	Integer	偏移量，默认为0
Limit	否	Integer	返回数量，默认为20，最大值为100
Order	否	String	输出列表的排列顺序。取值范围：ASC：升序排列 DESC：降序排列
OrderField	否	String	排序的依据字段，取值范围 "CREATE_TIME" "UPDATE_TIME"

3. 输出参数

参数名称	类型	描述
Services	Array of ModelService	服务列表
TotalCount	Integer	服务总数
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 描述服务

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DescribeServices
&Limit=1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "7bf9843d-9153-4647-99a0-05c70ba47603",
    "Services": [
      {
        "Id": "ank95gbm4dwfhmds",
        "Cluster": "ap-beijing",
        "Region": "ap-beijing",
        "Name": "test",
        "Runtime": "pmml",
        "ModelUri": "cos://ti-ems-125502019.cos.ap-beijing.myqcloud.com/models/pmml/",
        "Cpu": 100,
        "Memory": 100,
        "Gpu": 0,
        "GpuMemory": 0,
        "CreateTime": "2019-11-07T15:14:19+08:00",
        "UpdateTime": "2019-11-07T15:14:19+08:00",
        "ScaleMode": "MANUAL",
        "Scaler": {
          "MaxReplicas": 2,
          "MinReplicas": 0,
          "StartReplicas": 2,
          "HpaMetrics": []
        },
        "Status": {
          "DesiredReplicas": 2,
          "CurrentReplicas": 2,
          "Replicas": [
            "ank95gbm4dwfhmds-fccc99cdd-ftff5",
            "ank95gbm4dwfhmds-fccc99cdd-fvskl"
          ],
          "Status": "Normal",
          "Message": "",
          "Conditions": []
        },
        "AccessToken": "",
        "ConfigName": "test1",
        "ConfigVersion": "1.0",
        "ServeSeconds": 470,
        "ConfigId": "asjqn7wtdgrd7gt",
        "ResourceGroupld": "ap-beijing",
        "ResourceGroupName": "公共资源组",
        "Exposes": [],
```

```
"Description": "test",
"GpuType": ""
},
],
"TotalCount": 2
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
FailedOperation.AlreadyExists	已经存在同名资源。
InternalServerError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

暴露服务

最近更新时间：2022-04-13 07:17:20

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

暴露服务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ExposeService。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ServiceId	是	String	服务Id
ExposeType	是	String	暴露方式，支持 EXTERNAL（外网暴露），VPC（VPC内网打通）
VpcId	否	String	暴露方式为 VPC 时，填写需要打通的私有网络Id
SubnetId	否	String	暴露方式为 VPC 时，填写需要打通的子网Id

3. 输出参数

参数名称	类型	描述
Expose	ExposeInfo	暴露方式
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 暴露模型服务

输入示例

```
https://tiems.tencentcloudapi.com/?Action=ExposeService
&ServiceId=ank95gbm4dwfhmds
&ExposeType=EXTERNAL
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "2b6049dc-f99a-47e9-99fa-f701ecc4b76a",
    "Expose": {
      "ExposeType": "EXTERNAL",
      "VpcId": "",
      "SubnetId": "",
      "Ip": "pending"
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
FailedOperation.AlreadyExists	已经存在同名资源。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。

错误码	描述
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

资源管理相关接口

获取节点列表

最近更新时间：2022-04-13 07:17:19

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

获取节点列表

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeInstances。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Filters.N	否	Array of Filter	筛选选项
Offset	否	Integer	偏移量，默认为0
Limit	否	Integer	返回数量，默认为20，最大值为200
Order	否	String	输出列表的排列顺序。取值范围：ASC：升序排列 DESC：降序排列
OrderField	否	String	排序的依据字段，取值范围 "CREATE_TIME", "UPDATE_TIME", "NAME"
ResourceGroupId	否	String	要查询的资源组 ID

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	资源组下节点总数
Instances	Array of Instance	资源组下节点列表
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 展示用户当前地域下所有节点

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DescribeInstances
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "Instances": [
      {
        "AbnormalReason": "",
        "Cpu": 24,
        "CpuRequested": 0,
        "Created": "2019-11-06T16:18:22+08:00",
        "DeadlineTime": "2020-02-06T16:18:36+08:00",
        "Gpu": 0,
        "GpuRequested": 0,
        "Id": "ins-19a065ev",
        "InstanceChargeType": "PREPAID",
        "InstanceType": "sv_tiems_instance_24c48g",
        "Memory": 48,
        "MemoryRequested": 0,
        "Region": "ap-beijing",
        "RenewFlag": "DISABLE_NOTIFY_AND_MANUAL_RENEW",
        "ResourceGroupId": "fl2hrhgp97jcg8pf",
        "State": "Running",
        "Updated": "2019-11-07T20:50:34+08:00",
        "Zone": "ap-beijing-2"
      },
      {
        "AbnormalReason": "",
        "Cpu": 24,
        "CpuRequested": 0,
        "Created": "2019-11-06T16:18:22+08:00",
        "DeadlineTime": "2020-02-06T16:18:29+08:00",
        "Gpu": 0,
        "GpuRequested": 0,
        "Id": "ins-19a066i9",
        "InstanceChargeType": "PREPAID",
        "InstanceType": "sv_tiems_instance_24c48g",
        "Memory": 48,
        "MemoryRequested": 0,
        "Region": "ap-beijing",
        "RenewFlag": "DISABLE_NOTIFY_AND_MANUAL_RENEW",
        "ResourceGroupId": "pf2kdwlmn9s2wq5m",
        "State": "Running",
        "Updated": "2019-11-07T20:50:34+08:00",
        "Zone": "ap-beijing-2"
      }
    ]
  }
}
```

```
],  
  "RequestId": "675e9ba8-f0ad-4727-83cc-384ee8089ac6",  
  "TotalCount": 2  
}  
}
```

示例2 展示用户当前地域下某资源组的所有节点

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DescribeInstances  
&ResourceGroupId=pf2kdwlmn9s2wq5m  
&<公共请求参数>
```

输出示例

```
{  
  "Response": {  
    "Instances": [  
      {  
        "AbnormalReason": "",  
        "Cpu": 24,  
        "CpuRequested": 0,  
        "Created": "2019-11-06T16:18:22+08:00",  
        "DeadlineTime": "2020-02-06T16:18:29+08:00",  
        "Gpu": 0,  
        "GpuRequested": 0,  
        "Id": "ins-19a066i9",  
        "InstanceChargeType": "PREPAID",  
        "InstanceType": "sv_tiems_instance_24c48g",  
        "Memory": 48,  
        "MemoryRequested": 0,  
        "Region": "ap-beijing",  
        "RenewFlag": "DISABLE_NOTIFY_AND_MANUAL_RENEW",  
        "ResourceGroupId": "pf2kdwlmn9s2wq5m",  
        "State": "Running",  
        "Updated": "2019-11-07T20:50:34+08:00",  
        "Zone": "ap-beijing-2"  
      }  
    ],  
    "RequestId": "6c6ebb92-5ede-491e-ab88-d49cb3dc22a1",  
    "TotalCount": 1  
  }  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。

停用资源组的伸缩组

最近更新时间：2022-04-13 07:17:18

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

停用资源组的伸缩组

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DisableRsgAsGroup。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Id	是	String	伸缩组 ID

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 停用资源组的伸缩组

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DisableRsgAsGroup
&Id=asg-xxxxxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "0e21c605-3d85-4fc0-bc91-4c61ca250ad5"
```

```
}  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnsupportedOperation	操作不支持。

获取资源组列表

最近更新时间：2022-04-13 07:17:19

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

获取资源组列表

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeResourceGroups。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Filters.N	否	Array of Filter	筛选选项
Offset	否	Integer	偏移量，默认为0
Limit	否	Integer	返回数量，默认为20，最大值为200
Order	否	String	输出列表的排列顺序。取值范围：ASC：升序排列 DESC：降序排列
OrderField	否	String	排序的依据字段，取值范围 "CREATE_TIME", "UPDATE_TIME", "NAME"

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	资源组总数
ResourceGroups	Array of ResourceGroup	资源组列表 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 展示用户当前地域下的可用资源组（包括公共资源组）

输入示例

<https://tiems.tencentcloudapi.com/?Action=DescribeResourceGroups>
&<公共请求参数>

输出示例

```
{
  "Response": {
    "RequestId": "0e21c605-3d85-4fc0-bc91-4c61ca250ad5",
    "ResourceGroups": [
      {
        "Region": "ap-beijing",
        "Cluster": "ap-beijing",
        "Id": "ap-beijing",
        "Name": "公共资源组",
        "Description": "公共资源组",
        "Created": "",
        "Updated": "",
        "InstanceCount": 0,
        "ServiceCount": 0,
        "JobCount": 0,
        "InstanceType": "sv_tiems_instance_4c20g1p4",
        "Gpu": 0,
        "Cpu": 0,
        "Memory": 0,
        "Status": "Ready",
        "Zone": "ap-beijing-4",
        "GpuType": [
          "P4"
        ],
        "Public": true
      },
      {
        "Cluster": "ap-beijing",
        "Cpu": 336,
        "Created": "2019-10-22T22:32:38+08:00",
        "Description": "",
        "Gpu": 0,
        "GpuType": [],
        "Id": "kd8rgp2qpxf5h46f",
        "InstanceCount": 14,
        "InstanceType": "sv_tiems_instance_24c48g",
        "JobCount": 2,
        "Memory": 0,
        "Name": "rsg-kd8rgp2qpxf5h46f",
        "Public": false,
        "Region": "ap-beijing",
        "ServiceCount": 10,
        "Status": "Ready",
        "Updated": "2019-10-22T22:32:38+08:00",
```

```
"Zone": "ap-beijing-2"
}
],
"TotalCount": 2
}
}
```

示例2 展示用户当前地域下的可用资源组（不包括公共资源组）

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DescribeResourceGroups
&Filters.0.Name=public
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "0e21c605-3d85-4fc0-bc91-4c61ca250ad5",
    "ResourceGroups": [
      {
        "Cluster": "ap-beijing",
        "Cpu": 336,
        "Created": "2019-10-22T22:32:38+08:00",
        "Description": "",
        "Gpu": 0,
        "GpuType": [],
        "Id": "kd8rgp2qpxf5h46f",
        "InstanceCount": 14,
        "InstanceType": "sv_tiems_instance_24c48g",
        "JobCount": 2,
        "Memory": 0,
        "Name": "rsg-kd8rgp2qpxf5h46f",
        "Public": false,
        "Region": "ap-beijing",
        "ServiceCount": 10,
        "Status": "Ready",
        "Updated": "2019-10-22T22:32:38+08:00",
        "Zone": "ap-beijing-2"
      }
    ],
    "TotalCount": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。

查询伸缩组活动

最近更新时间：2022-04-13 07:17:18

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

查询伸缩组活动

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeRsgAsGroupActivities。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Id	是	String	伸缩组 ID
StartTime	否	String	查询活动的开始时间
EndTime	否	String	查询互动的结束时间
Filters.N	否	Array of Filter	筛选选项
Offset	否	Integer	偏移量，默认为 0
Limit	否	Integer	返回数量，默认为 20，最大值为 200
Order	否	String	输出列表的排列顺序。取值范围："ASC", "DESC"
OrderField	否	String	排序的依据字段，取值范围 "CREATE_TIME", "UPDATE_TIME", "NAME"

3. 输出参数

参数名称	类型	描述
RsgAsGroupActivitySet	Array of RsgAsGroupActivity	伸缩组活动数组 注意：此字段可能返回 null，表示取不到有效值。
TotalCount	Integer	所查询的伸缩组活动总数目
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询伸缩组活动

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DescribeRsgAsGroupActivities
&Id=asg-xxxxxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "206bf53c-c658-44cb-b50a-5eb9904c49fd",
    "RsgAsGroupActivitySet": [
      {
        "Id": "asa-xxxxxxx",
        "RsgAsGroupId": "asg-xxxxxxx",
        "ActivityType": "SCALE_OUT",
        "StatusCode": "FAILED",
        "StatusMessage": "选择的机型在当前可用区已售罄",
        "Cause": "因匹配期望实例数",
        "Description": "因匹配期望实例数，扩容1台",
        "StartTime": "2019-12-30T10:11:26+08:00",
        "EndTime": "2019-12-30T10:11:27+08:00",
        "CreateTime": "2019-12-30T10:11:26+08:00",
        "StatusMessageSimplified": "云主机售罄"
      }
    ],
    "TotalCount": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)

- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnsupportedOperation	操作不支持。

删除资源组

最近更新时间：2022-04-13 07:17:19

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

删除资源组

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteResourceGroup。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ResourceGroupId	是	String	要删除的资源组 ID

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除资源组

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DeleteResourceGroup
&ResourceGroupId=rsg-xxxxxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "0e21c605-3d85-4fc0-bc91-4c61ca250ad5"
```

```
}  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnsupportedOperation	操作不支持。

删除节点

最近更新时间：2022-04-13 07:17:19

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

删除资源组中的节点。目前仅支持删除已经到期的预付费节点，和按量付费节点。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteInstance。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
InstanceId	是	String	要删除的节点 ID

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除节点

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DeleteInstance
&InstanceId=ins-xxxxxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "0e21c605-3d85-4fc0-bc91-4c61ca250ad5"
```

```
}  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 API Inspector 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 API Explorer 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

启用资源组的伸缩组

最近更新时间：2022-04-13 07:17:18

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

启用资源组的伸缩组

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：EnableRsgAsGroup。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Id	是	String	伸缩组 ID

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 启用资源组的伸缩组

输入示例

```
https://tiems.tencentcloudapi.com/?Action=EnableRsgAsGroup
&Id=asg-xxxxxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "0e21c605-3d85-4fc0-bc91-4c61ca250ad5"
```

```
}  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnsupportedOperation	操作不支持。

查询资源组的伸缩组信息

最近更新时间：2022-04-13 07:17:18

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

查询资源组的伸缩组信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeRsgAsGroups。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Filters.N	否	Array of Filter	筛选选项
Offset	否	Integer	偏移量，默认为 0
Limit	否	Integer	返回数量，默认为 20，最大值为 200
Order	否	String	输出列表的排列顺序。取值范围："ASC", "DESC"
OrderField	否	String	排序的依据字段，取值范围 "CREATE_TIME", "UPDATE_TIME", "NAME"

3. 输出参数

参数名称	类型	描述
RsgAsGroupSet	Array of RsgAsGroup	所查询的伸缩组数组
TotalCount	Integer	伸缩组数组总数目
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询资源组的伸缩组信息

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DescribeRsgAsGroups
Filters.0.Name=rsg_id
&Filters.0.Values.0=rsg-xxxxxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "3d96e8c0-97a9-43a2-8268-99ff505fbd84",
    "RsgAsGroupSet": [
      {
        "Id": "asg-xxxxxxx",
        "Region": "ap-beijing",
        "Zone": "ap-beijing-5",
        "Cluster": "ap-beijing",
        "RsgId": "rsg-xxxxxxx",
        "Name": "rsg-xxxxxxx",
        "MaxSize": 3,
        "MinSize": 0,
        "CreateTime": "2019-12-24T17:39:40+08:00",
        "UpdateTime": "2019-12-24T17:39:40+08:00",
        "Status": "Enabled",
        "InstanceType": "sv_tiems_instance_8c32g1t4",
        "InstanceCount": 1
      }
    ],
    "TotalCount": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)

- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnsupportedOperation	操作不支持。

更新资源组的伸缩组

最近更新时间：2022-04-13 07:17:18

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

更新资源组的伸缩组

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：UpdateRsgAsGroup。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Id	是	String	伸缩组 ID
Name	否	String	重命名名称
MaxSize	否	Integer	伸缩组最大节点数
MinSize	否	Integer	伸缩组最小节点数
DesiredSize	否	Integer	伸缩组期望的节点数

3. 输出参数

参数名称	类型	描述
RsgAsGroup	RsgAsGroup	资源组的伸缩组
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 更新资源组的伸缩组

输入示例

```
https://tiems.tencentcloudapi.com/?Action=UpdateRsgAsGroup
&Id=asg-xxxxxxx
```

```
$Name=<name>
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "3d96e8c0-97a9-43a2-8268-99ff505fbd84",
    "RsgAsGroup": {
      "Id": "asg-xxxxxxx",
      "Region": "ap-beijing",
      "Zone": "ap-beijing-5",
      "Cluster": "ap-beijing",
      "RsgId": "rsg-xxxxxxx",
      "Name": "<name>",
      "MaxSize": 5,
      "MinSize": 0,
      "CreateTime": "2019-12-24T17:39:40+08:00",
      "UpdateTime": "2019-12-24T17:39:40+08:00",
      "Status": "Enabled",
      "InstanceType": "sv_tiems_instance_8c32g1t4",
      "InstanceCount": 1
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

• [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnsupportedOperation	操作不支持。

创建资源组的伸缩组

最近更新时间：2022-04-13 07:17:20

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

创建资源组的伸缩组。当前一个资源组仅允许创建一个伸缩组。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateRsgAsGroup。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
RsgId	是	String	资源组 ID
MaxSize	是	Integer	伸缩组允许的最大节点数
MinSize	是	Integer	伸缩组允许的最小节点数
InstanceType	是	String	伸缩组的节点规格
Cluster	否	String	资源组所在的集群名
Name	否	String	伸缩组名称
DesiredSize	否	Integer	伸缩组期望的节点数

3. 输出参数

参数名称	类型	描述
RsgAsGroup	RsgAsGroup	所创建的资源组的伸缩组
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建资源组的伸缩组

输入示例

```
https://tiems.tencentcloudapi.com/?Action=CreateRsgAsGroup
&Cluster=ap-beijing
&RsgId=rsg-xxxxxxx
&Name=asg-xxxxxxx
&MaxSize=5
&MinSize=0
&InstanceType=sv_tiems_instance_8c32g1t4
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "3d96e8c0-97a9-43a2-8268-99ff505fbd84",
    "RsgAsGroup": {
      "Id": "asg-xxxxxxx",
      "Region": "ap-beijing",
      "Zone": "ap-beijing-5",
      "Cluster": "ap-beijing",
      "RsgId": "rsg-xxxxxxx",
      "Name": "xxxx",
      "MaxSize": 5,
      "MinSize": 0,
      "CreateTime": "2019-12-24T17:39:40+08:00",
      "UpdateTime": "2019-12-24T17:39:40+08:00",
      "Status": "Enabled",
      "InstanceType": "sv_tiems_instance_8c32g1t4",
      "InstanceCount": 0
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)

- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
FailedOperation.AlreadyExists	已经存在同名资源。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

删除资源组的伸缩组

最近更新时间：2022-04-13 07:17:19

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

伸缩

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteRsgAsGroup。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Id	是	String	伸缩组 ID

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除资源组的伸缩组

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DeleteRsgAsGroup
&Id=asg-xxxxxxx
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "0e21c605-3d85-4fc0-bc91-4c61ca250ad5"
```

```
}  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。

配置管理相关接口

描述服务配置

最近更新时间：2022-04-13 07:17:17

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

描述服务配置

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeServiceConfigs。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Filters.N	否	Array of Filter	筛选选项，支持按照name等进行筛选
Offset	否	Integer	偏移量，默认为0
Limit	否	Integer	返回数量，默认为20，最大值为1000
Order	否	String	输出列表的排列顺序。取值范围：ASC：升序排列 DESC：降序排列
OrderField	否	String	排序的依据字段，取值范围 "CREATE_TIME", "UPDATE_TIME", "NAME"
PageByName	否	Boolean	是否按照配置名分页

3. 输出参数

参数名称	类型	描述
ServiceConfigs	Array of Config	服务配置
TotalCount	Integer	服务配置总数
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 描述服务配置

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DescribeServiceConfigs
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "4a3d63f8-fb12-4dd5-b444-02c0a91af0d2",
    "ServiceConfigs": [
      {
        "Id": "asjcqn7wtdgrd7t",
        "Name": "test1",
        "Runtime": "pmml",
        "ModelUri": "cos://ti-ems-example.cos.ap-beijing.myqcloud.com/models/pmml/",
        "CreateTime": "2019-11-04T17:34:58+08:00",
        "UpdateTime": "2019-11-04T17:34:58+08:00",
        "Version": "1.0",
        "Description": "字符串"
      }
    ],
    "TotalCount": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

• [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

删除服务配置

最近更新时间：2022-04-13 07:17:17

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

删除服务配置

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteServiceConfig。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ServiceConfigId	否	String	服务配置Id
ServiceConfigName	否	String	服务配置名称

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除服务配置

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DeleteServiceConfig
&ServiceConfigName=test
&<公共请求参数>
```

输出示例

```
{
  "Response": {
```

```
"RequestId": "9120fa1d-f19c-4b11-a9af-956911fbc85b"
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

创建服务配置

最近更新时间：2022-04-13 07:17:17

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

创建服务配置

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateServiceConfig。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Name	是	String	配置名称
Runtime	是	String	运行环境
ModelUri	是	String	模型地址，支持cos路径，格式为 cos://bucket名-appid.cos.region名.myqcloud.com/模型文件夹路径。为模型文件的上一层文件夹地址。
Description	否	String	配置描述

3. 输出参数

参数名称	类型	描述
ServiceConfig	Config	服务配置
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建服务配置

输入示例

```
https://tiems.tencentcloudapi.com/?Action=CreateServiceConfig
&Name=test
&Runtime=pmml
&ModelUri=cas://ti-example-125550209.cos.ap-beijing.myqcloud.com/models/pmml/
```

&Description=test
&<公共请求参数>

输出示例

```
{
  "Response": {
    "RequestId": "63dccc5a-0084-4a3e-a9ae-cefb019fce9d",
    "ServiceConfig": {
      "Id": "aktst2r9fkmtpdf8",
      "Name": "test",
      "Runtime": "pmml",
      "ModelUri": "cos://ti-example-125550209.cos.ap-beijing.myqcloud.com/models/pmml/",
      "CreateTime": "2019-11-07T15:01:12+08:00",
      "UpdateTime": "2019-11-07T15:01:12+08:00",
      "Version": "1.0",
      "Description": "test"
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
FailedOperation.AlreadyExists	已经存在同名资源。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

任务管理相关接口

更新任务

最近更新时间：2022-04-13 07:17:21

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

更新任务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：UpdateJob。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	任务 Id
JobAction	否	String	任务更新动作，支持：Cancel
Description	否	String	备注

3. 输出参数

参数名称	类型	描述
Job	Job	任务 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 更新任务

输入示例

```
https://tiems.tencentcloudapi.com/?Action=UpdateJob
&JobId=aktwbxqmfr8g44v
```

&Description=123
&<公共请求参数>

输出示例

```
{
  "Response": {
    "RequestId": "a31293c1-d8ba-4dbb-828e-6f16016fbb88",
    "Job": {
      "Id": "aktwbxqmfkr8g44v",
      "Cluster": "ap-beijing",
      "Region": "ap-beijing",
      "Name": "test",
      "Runtime": "pmml",
      "Description": "123",
      "ConfigId": "0",
      "PredictInput": {
        "InputDataFormat": "json",
        "OutputDataFormat": "json",
        "InputPath": "cos://data-test-125502019.cos.ap-guangzhou.myqcloud.com/test/pmminput",
        "OutputPath": "cos://data-test-125502019.cos.ap-guangzhou.myqcloud.com/test/",
        "BatchSize": 32,
        "SignatureName": ""
      },
      "Status": {
        "DesiredWorkers": 1,
        "CurrentWorkers": 1,
        "Replicas": [
          "aktwbxqmfkr8g44v-mwjbb"
        ],
        "Status": "Cancelled",
        "Message": ""
      },
      "CreateTime": "2019-11-07T15:41:35+08:00",
      "StartTime": "2019-11-07T15:41:35+08:00",
      "EndTime": "",
      "CancelTime": "",
      "ResourceGroupId": "ap-beijing",
      "ResourceGroupName": "公共资源组",
      "Cpu": 100,
      "Memory": 100,
      "Gpu": 0,
      "GpuMemory": 0,
      "GpuType": "",
      "ConfigName": "test1",
      "ConfigVersion": "1.0"
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
FailedOperation.AlreadyExists	已经存在同名资源。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。

创建任务

最近更新时间：2022-04-13 07:17:22

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

创建任务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateJob。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Name	是	String	任务名称
ResourceGroupId	是	String	使用的资源组 Id，默认使用共享资源组
Cpu	是	Integer	处理器配置，单位为1/1000核；范围[100, 256000]
Memory	是	Integer	内存配置，单位为1M；范围[100, 256000]
Cluster	否	String	运行集群
PredictInput	否	PredictInput	预测输入
Description	否	String	任务描述
WorkerCount	否	Integer	同时处理任务的 Worker 个数
ConfigId	否	String	使用的配置 Id
Gpu	否	Integer	GPU算力配置，单位为1/1000 卡，范围 [0, 256000]
GpuMemory	否	Integer	显存配置，单位为1M，范围 [0, 256000]
GpuType	否	String	GPU类型
QuantizationInput	否	QuantizationInput	量化输入
LogTopicId	否	String	Cls日志主题ID

3. 输出参数

参数名称	类型	描述
Job	Job	任务
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建任务

输入示例

```
https://tiems.tencentcloudapi.com/?Action=CreateJob
&Name=test
&Description=job
&WorkerCount=1
&ConfigId=asjqn7wtdgrd7gt
&ResourceGroupId=ap-beijing
&Cpu=100
&Memory=100
&PredictInput.InputDataFormat=json
&PredictInput.OutputDataFormat=json
&PredictInput.InputPath=cos://data-test-1255502019.cos.ap-guangzhou.myqcloud.com/test/pmminput
&PredictInput.OutputPath=cos://data-test-1255502019.cos.ap-guangzhou.myqcloud.com/test/
&PredictInput.BatchSize=32
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "0448ae0a-d593-4a48-a025-bede58e1f841",
    "Job": {
      "Id": "aktwbxqmfkr8g44v",
      "Cluster": "ap-beijing",
      "Region": "ap-beijing",
      "Name": "test",
      "Runtime": "pmm1",
      "Description": "job",
      "ConfigId": "asjqn7wtdgrd7gt",
      "PredictInput": {
        "InputDataFormat": "json",
        "OutputDataFormat": "json",
        "InputPath": "cos://data-test-1255502019.cos.ap-guangzhou.myqcloud.com/test/pmminput",
        "OutputPath": "cos://data-test-1255502019.cos.ap-guangzhou.myqcloud.com/test/",
        "BatchSize": 32,
        "SignatureName": ""
      },
      "Status": {
        "DesiredWorkers": 0,
        "CurrentWorkers": 0,
```

```
"Replicas": [],
"Status": "Waiting",
"Message": ""
},
"CreateTime": "2019-11-07T15:41:35+08:00",
"StartTime": "",
"EndTime": "",
"CancelTime": "",
"ResourceGroupId": "ap-beijing",
"ResourceGroupName": "公共资源组",
"Cpu": 100,
"Memory": 100,
"Gpu": 0,
"GpuMemory": 0,
"GpuType": "",
"ConfigName": "test1",
"ConfigVersion": "1.0"
}
}
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
FailedOperation.AlreadyExists	已经存在同名资源。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

删除任务

最近更新时间：2022-04-13 07:17:22

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

删除任务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteJob。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
JobId	是	String	任务 Id

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除任务

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DeleteJob
&JobId=aktwbxqmfkr8g44v
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "f478e4dd-02e5-4468-9a5d-baf762b8e36d"
```

```
}  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
InternalServerError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。

其他接口

描述服务运行环境

最近更新时间：2022-04-13 07:17:21

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

描述服务运行环境

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeRuntimes。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
Runtimes	Array of Runtime	TIEMS支持的运行环境列表
UserAccess	Integer	用户对runtime对权限 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 描述服务运行环境

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DescribeRuntimes
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "efcdba3f-080f-4913-a7a2-5f47e0baf67a",
    "Runtimes": [
      {
        "Name": "pmml",
        "Framework": "pmml",
        "Description": "pmml-description",
        "Public": true,
        "HealthCheckOn": true,
        "Image": "**",
        "CreateTime": "2019-11-01T17:03:26+08:00"
      },
      {
        "Name": "angel",
        "Framework": "angel",
        "Description": "angel-description",
        "Public": true,
        "HealthCheckOn": true,
        "Image": "**",
        "CreateTime": "2019-11-01T17:03:26+08:00"
      },
      {
        "Name": "tf-serving",
        "Framework": "tf-serving",
        "Description": "tf-serving-description",
        "Public": true,
        "HealthCheckOn": true,
        "Image": "**",
        "CreateTime": "2019-11-01T17:03:26+08:00"
      },
      {
        "Name": "test",
        "Framework": "test",
        "Description": "字符串",
        "Public": false,
        "HealthCheckOn": false,
        "Image": "ccr.ccs.tencentyun.com/mla-library/tool",
        "CreateTime": "2019-11-07T14:47:49+08:00"
      }
    ],
    "UserAccess": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。

创建运行环境

最近更新时间：2022-04-13 07:17:21

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

创建运行环境

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateRuntime。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Name	是	String	全局唯一的运行环境名称
Image	是	String	运行环境镜像地址
Framework	否	String	运行环境框架
Description	否	String	运行环境描述
HealthCheckOn	否	Boolean	是否支持健康检查，默认为False

3. 输出参数

参数名称	类型	描述
Runtime	Runtime	运行环境
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建自定义运行环境

输入示例

```
https://tiems.tencentcloudapi.com/?Action=CreateRuntime
&Name=test
&Framework=test
```

```
&Image=ccr.ccs.tencentyun.com/testrepo/testimage:v1
&Description=test
&HealthCheckOn=false
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "e8717195-6b62-4fcc-99fe-e29d2aecb21e",
    "Runtime": {
      "Name": "test",
      "Framework": "test",
      "Description": "字符串",
      "Public": false,
      "HealthCheckOn": false,
      "Image": "ccr.ccs.tencentyun.com/testrepo/testimage:v1",
      "CreateTime": "2019-11-07T14:47:49+08:00"
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

[腾讯云 API 平台](#) 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
FailedOperation.AlreadyExists	已经存在同名资源。
InternalError	内部错误。
InvalidParameter	参数错误。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

删除运行环境

最近更新时间：2022-04-13 07:17:21

1. 接口描述

接口请求域名：tiems.tencentcloudapi.com。

删除运行环境

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

API Explorer 提供了在线调用、签名验证、SDK 代码生成和快速检索接口等能力。您可查看每次调用的请求内容和返回结果以及自动生成 SDK 调用示例。

2. 输入参数

以下请求参数列表仅列出了接口请求参数和部分公共参数，完整公共参数列表见 [公共请求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteRuntime。
Version	是	String	公共参数 ，本接口取值：2019-04-16。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Runtime	是	String	要删除的Runtime名

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除运行环境

输入示例

```
https://tiems.tencentcloudapi.com/?Action=DeleteRuntime
&Runtime=test
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "RequestId": "754a8f06-3e10-4282-8c77-b0efc6fbb132"
```

```
}  
}
```

5. 开发者资源

腾讯云 API 平台

腾讯云 API 平台 是综合 API 文档、错误码、API Explorer 及 SDK 等资源的统一查询平台，方便您从同一入口查询及使用腾讯云提供的所有 API 服务。

API Inspector

用户可通过 [API Inspector](#) 查看控制台每一步操作关联的 API 调用情况，并自动生成各语言版本的 API 代码，也可前往 [API Explorer](#) 进行在线调试。

SDK

云 API 3.0 提供了配套的开发工具集（SDK），支持多种编程语言，能更方便的调用 API。

- [Tencent Cloud SDK 3.0 for Python](#)
- [Tencent Cloud SDK 3.0 for Java](#)
- [Tencent Cloud SDK 3.0 for PHP](#)
- [Tencent Cloud SDK 3.0 for Go](#)
- [Tencent Cloud SDK 3.0 for NodeJS](#)
- [Tencent Cloud SDK 3.0 for .NET](#)
- [Tencent Cloud SDK 3.0 for C++](#)
- [Tencent Cloud SDK 3.0 for Ruby](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

错误码	描述
FailedOperation	操作失败。
InternalError	内部错误。
InvalidParameter	参数错误。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

数据结构

最近更新时间：2020-07-22 08:19:24

Conditions

状态

被如下接口引用：CreateService, DescribeServices, UpdateService。

名称	类型	描述
Reason	String	原因
Count	Integer	具有相同原因的副本个数

Config

配置

被如下接口引用：CreateServiceConfig, DescribeServiceConfigs。

名称	类型	描述
Id	String	Id
Name	String	配置名
ModelUri	String	模型地址
CreateTime	String	创建时间
Runtime	String	运行环境
Version	String	配置版本
UpdateTime	String	更新时间
Description	String	配置描述 注意：此字段可能返回 null，表示取不到有效值。

ExposeInfo

暴露信息

被如下接口引用：CreateService, DescribeServices, ExposeService, UpdateService。

名称	类型	描述
ExposeType	String	暴露方式，支持 EXTERNAL（外网暴露），VPC（VPC内网打通）
Ip	String	暴露Ip。暴露方式为 EXTERNAL 为外网 Ip，暴露方式为 VPC 时为指定 Vpc 下的Vip
VpcId	String	暴露方式为 VPC 时，打通的私有网络Id 注意：此字段可能返回 null，表示取不到有效值。
SubnetId	String	暴露方式为 VPC 时，打通的子网Id 注意：此字段可能返回 null，表示取不到有效值。
GateWayServiceId	String	GATEWAY 服务id，ExposeType = GATEWAY 时返回 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
GateWayAPIId	String	GATEWAY api id, ExposeType = GATEWAY 时返回 注意：此字段可能返回 null，表示取不到有效值。
GateWayDomain	String	GATEWAY domain, ExposeType = GATEWAY 时返回 注意：此字段可能返回 null，表示取不到有效值。

Filter

筛选项

被如下接口引用：DescribeInstances, DescribeResourceGroups, DescribeRsgAsGroupActivities, DescribeRsgAsGroups, DescribeServiceConfigs, DescribeServices。

名称	类型	必选	描述
Name	String	是	名称
Values	Array of String	是	取值

Instance

节点

被如下接口引用：DescribeInstances。

名称	类型	描述
Id	String	节点 ID
Zone	String	节点所在地区
InstanceType	String	节点类型
InstanceChargeType	String	节点充值类型
Cpu	Integer	Cpu 核数
Memory	Integer	内存
Gpu	Integer	Gpu 核数
State	String	节点状态
AbnormalReason	String	节点故障信息
Created	String	创建时间
Updated	String	更新时间
DeadlineTime	String	到期时间
ResourceGroupId	String	所属资源组 ID
RenewFlag	String	自动续费标签
Region	String	节点所在地域
CpuRequested	Integer	当前 Cpu 申请使用量
MemoryRequested	Integer	当前 Memory 申请使用量

名称	类型	描述
GpuRequested	Integer	当前 Gpu 申请使用量
RsgAsGroupId	String	节点所在伸缩组 ID

Job

任务

被如下接口引用：CreateJob, UpdateJob。

名称	类型	描述
Id	String	任务 Id
Cluster	String	集群名 注意：此字段可能返回 null，表示取不到有效值。
Region	String	Region 名
Name	String	任务名称
Runtime	String	Worker 使用的运行环境 注意：此字段可能返回 null，表示取不到有效值。
Description	String	任务描述 注意：此字段可能返回 null，表示取不到有效值。
ConfigId	String	配置 Id 注意：此字段可能返回 null，表示取不到有效值。
PredictInput	PredictInput	预测输入 注意：此字段可能返回 null，表示取不到有效值。
Status	JobStatus	任务状态
CreateTime	String	任务创建时间
StartTime	String	任务开始时间 注意：此字段可能返回 null，表示取不到有效值。
EndTime	String	任务结束时间 注意：此字段可能返回 null，表示取不到有效值。
CancelTime	String	任务取消时间 注意：此字段可能返回 null，表示取不到有效值。
ResourceGroupId	String	任务使用资源组 Id 注意：此字段可能返回 null，表示取不到有效值。
Cpu	Integer	处理器配置, 单位为1/1000核; 范围[100, 256000] 注意：此字段可能返回 null，表示取不到有效值。
Memory	Integer	内存配置, 单位为1M; 范围[100, 256000] 注意：此字段可能返回 null，表示取不到有效值。
Gpu	Integer	GPU算力配置, 单位为1/1000 卡, 范围 [0, 256000] 注意：此字段可能返回 null，表示取不到有效值。
GpuMemory	Integer	显存配置, 单位为1M, 范围 [0, 256000] 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
ResourceGroupName	String	任务使用资源组名称 注意：此字段可能返回 null，表示取不到有效值。
GpuType	String	GPU类型 注意：此字段可能返回 null，表示取不到有效值。
ConfigName	String	配置名 注意：此字段可能返回 null，表示取不到有效值。
ConfigVersion	String	配置版本 注意：此字段可能返回 null，表示取不到有效值。
JobType	String	Job类型 注意：此字段可能返回 null，表示取不到有效值。
QuantizationInput	QuantizationInput	量化输入 注意：此字段可能返回 null，表示取不到有效值。
LogTopicId	String	Cls日志主题ID 注意：此字段可能返回 null，表示取不到有效值。

JobStatus

任务状态

被如下接口引用：CreateJob, UpdateJob。

名称	类型	描述
Status	String	任务状态
Message	String	错误时为错误描述 注意：此字段可能返回 null，表示取不到有效值。
DesiredWorkers	Integer	预期Worker数量 注意：此字段可能返回 null，表示取不到有效值。
CurrentWorkers	Integer	当前Worker数量 注意：此字段可能返回 null，表示取不到有效值。
Replicas	Array of String	副本名 注意：此字段可能返回 null，表示取不到有效值。
ReplicaInfos	Array of ReplicaInfo	副本实例 注意：此字段可能返回 null，表示取不到有效值。

ModelService

模型服务

被如下接口引用：CreateService, DescribeServices, UpdateService。

名称	类型	描述
Id	String	服务ID
Cluster	String	运行集群 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
Name	String	服务名称
Runtime	String	运行环境
ModelUri	String	模型地址
Cpu	Integer	处理器配置, 单位为1/1000核
Memory	Integer	内存配置, 单位为1M
Gpu	Integer	GPU 配置, 单位为1/1000 卡
GpuMemory	Integer	显存配置, 单位为1M
CreateTime	String	创建时间
UpdateTime	String	更新时间
ScaleMode	String	支持AUTO, MANUAL
Scaler	Scaler	弹性伸缩配置
Status	ServiceStatus	服务状态
AccessToken	String	访问密钥 注意：此字段可能返回 null，表示取不到有效值。
ConfigId	String	服务配置Id
ConfigName	String	服务配置名
ServeSeconds	Integer	服务运行时长
ConfigVersion	String	配置版本 注意：此字段可能返回 null，表示取不到有效值。
ResourceGroupId	String	服务使用资源组 Id 注意：此字段可能返回 null，表示取不到有效值。
Exposes	Array of ExposeInfo	暴露方式 注意：此字段可能返回 null，表示取不到有效值。
Region	String	Region 名 注意：此字段可能返回 null，表示取不到有效值。
ResourceGroupName	String	服务使用资源组名称 注意：此字段可能返回 null，表示取不到有效值。
Description	String	备注 注意：此字段可能返回 null，表示取不到有效值。
GpuType	String	GPU类型 注意：此字段可能返回 null，表示取不到有效值。
LogTopicId	String	Cls日志主题Id 注意：此字段可能返回 null，表示取不到有效值。

Option

配置项

被如下接口引用：CreateService, DescribeServices, UpdateService。

名称	类型	必选	描述
Name	String	是	名称
Value	Integer	是	取值

PredictInput

预测输入

被如下接口引用：CreateJob, UpdateJob。

名称	类型	必选	描述
InputPath	String	是	输入路径，支持 cos 格式路径文件夹或文件
OutputPath	String	是	输出路径，支持 cos 格式路径
InputDataFormat	String	否	输入数据格式，目前支持：JSON
OutputDataFormat	String	否	输出数据格式，目前支持：JSON
BatchSize	Integer	否	预测批大小，默认为 64
SignatureName	String	否	模型签名 注意：此字段可能返回 null，表示取不到有效值。

QuantizationInput

量化输入

被如下接口引用：CreateJob, UpdateJob。

名称	类型	必选	描述
InputPath	String	是	量化输入路径
OutputPath	String	是	量化输出路径
BatchSize	Integer	否	量化批大小
Precision	String	否	量化精度，支持：FP32, FP16, INT8
ConvertType	String	否	转换类型

ReplicaInfo

实例信息

被如下接口引用：CreateJob, CreateService, DescribeServices, UpdateJob, UpdateService。

名称	类型	描述
Name	String	实例名称
Enilp	String	弹性网卡模式时，弹性网卡Ip 注意：此字段可能返回 null，表示取不到有效值。
Status	String	Normal: 正常运行中; Abnormal: 异常; Waiting: 等待中

名称	类型	描述
Message	String	当 status为 Abnormal 的时候，一些额外的信息 注意：此字段可能返回 null，表示取不到有效值。
StartTime	String	启动时间 注意：此字段可能返回 null，表示取不到有效值。
CreateTime	String	创建时间 注意：此字段可能返回 null，表示取不到有效值。
Restarted	Integer	重启次数

ResourceGroup

资源组

被如下接口引用：DescribeResourceGroups。

名称	类型	描述
Id	String	资源组 Id
Region	String	地域
Cluster	String	集群 注意：此字段可能返回 null，表示取不到有效值。
Name	String	资源组名称
Description	String	资源组描述 注意：此字段可能返回 null，表示取不到有效值。
Created	String	创建时间
Updated	String	更新时间
InstanceCount	Integer	资源组主机数量 注意：此字段可能返回 null，表示取不到有效值。
ServiceCount	Integer	使用资源组的服务数量 注意：此字段可能返回 null，表示取不到有效值。
JobCount	Integer	使用资源组的任务数量 注意：此字段可能返回 null，表示取不到有效值。
Public	Boolean	资源组是否为公共资源组 注意：此字段可能返回 null，表示取不到有效值。
InstanceType	String	机器类型 注意：此字段可能返回 null，表示取不到有效值。
Status	String	资源组状态 注意：此字段可能返回 null，表示取不到有效值。
Gpu	Integer	显卡总张数 注意：此字段可能返回 null，表示取不到有效值。
Cpu	Integer	处理器总核数 注意：此字段可能返回 null，表示取不到有效值。
Memory	Integer	内存总量，单位为G 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
Zone	String	可用区 注意：此字段可能返回 null，表示取不到有效值。
GpuType	Array of String	Gpu类型 注意：此字段可能返回 null，表示取不到有效值。
HasPrepaid	Boolean	该资源组下是否有预付费资源 注意：此字段可能返回 null，表示取不到有效值。
PayMode	String	资源组是否允许预付费或后付费模式 注意：此字段可能返回 null，表示取不到有效值。

RsgAsActivityRelatedInstance

伸缩组活动关联的节点

被如下接口引用：DescribeRsgAsGroupActivities。

名称	类型	描述
InstanceId	String	节点 ID
InstanceStatus	String	节点状态

RsgAsGroup

资源组的伸缩组

被如下接口引用：CreateRsgAsGroup, DescribeRsgAsGroups, UpdateRsgAsGroup。

名称	类型	描述
Id	String	伸缩组 ID
Region	String	伸缩组所在地域
Zone	String	伸缩组所在可用区
Cluster	String	伸缩组所在集群
RsgId	String	伸缩组所在资源组 ID
Name	String	伸缩组名称
MaxSize	Integer	伸缩组允许的最大节点个数
MinSize	Integer	伸缩组允许的最小节点个数
CreateTime	String	伸缩组创建时间
UpdateTime	String	伸缩组更新时间
Status	String	伸缩组状态
InstanceType	String	伸缩组节点类型
InstanceCount	Integer	伸缩组内节点个数
DesiredSize	Integer	伸缩组起始节点数

RsgAsGroupActivity

伸缩组活动信息

被如下接口引用：DescribeRsgAsGroupActivities。

名称	类型	描述
Id	String	伸缩组活动 ID
RsgAsGroupId	String	关联的伸缩组 ID
ActivityType	String	活动类型
StatusCode	String	状态的编码
StatusMessage	String	状态的消息
Cause	String	活动原因
Description	String	活动描述
StartTime	String	活动开始时间
EndTime	String	活动结束时间
CreateTime	String	活动创建时间
RsgAsActivityRelatedInstance	Array of RsgAsActivityRelatedInstance	活动相关联的节点
StatusMessageSimplified	String	简略的状态消息

Runtime

运行环境

被如下接口引用：CreateRuntime, DescribeRuntimes。

名称	类型	描述
Name	String	运行环境名称
Framework	String	运行环境框架
Description	String	运行环境描述
Public	Boolean	是否为公开运行环境 注意：此字段可能返回 null，表示取不到有效值。
HealthCheckOn	Boolean	是否打开健康检查 注意：此字段可能返回 null，表示取不到有效值。
Image	String	镜像地址 注意：此字段可能返回 null，表示取不到有效值。
CreateTime	String	创建时间 注意：此字段可能返回 null，表示取不到有效值。

Scaler

扩缩容配置

被如下接口引用：CreateService, DescribeServices, UpdateService。

名称	类型	必选	描述
MaxReplicas	Integer	是	最大副本数，ScaleMode 为 MANUAL 时此值会被置为 StartReplicas 取值
MinReplicas	Integer	是	最小副本数，ScaleMode 为 MANUAL 时此值会被置为 StartReplicas 取值
StartReplicas	Integer	是	起始副本数
HpaMetrics	Array of Option	否	扩缩容指标，选择自动扩缩容时至少需要选择一个指标，支持CPU-UTIL、MEMORY-UTIL

ServiceStatus

服务状态

被如下接口引用：CreateService, DescribeServices, UpdateService。

名称	类型	描述
DesiredReplicas	Integer	预期副本数
CurrentReplicas	Integer	当前副本数
Status	String	Normal：正常运行中；Abnormal：服务异常，例如容器启动失败等；Waiting：服务等待中，例如容器下载镜像过程等；Stopped：已停止 Stopping 停止中；Resuming：重启中；Updating：服务更新中
Conditions	Array of Conditions	服务处于当前状态的原因集合 注意：此字段可能返回 null，表示取不到有效值。
Replicas	Array of String	副本名称 注意：此字段可能返回 null，表示取不到有效值。
Message	String	运行状态对额外信息 注意：此字段可能返回 null，表示取不到有效值。
ReplicaInfos	Array of ReplicaInfo	副本信息 注意：此字段可能返回 null，表示取不到有效值。

错误码

最近更新时间：2022-02-28 08:18:18

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

```
{
  "Response": {
    "Error": {
      "Code": "AuthFailure.SignatureFailure",
      "Message": "The provided credentials could not be validated. Please check your signature is correct."
    },
    "RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
  }
}
```

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
ActionOffline	接口已下线。
AuthFailure.InvalidAuthorization	请求头部的 Authorization 不符合腾讯云标准。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。请在 控制台 检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务端时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。签名计算错误，请对照调用方式中的签名方法文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未授权。请参考 CAM 文档对鉴权的说明。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误（包括参数格式、类型等错误）。
InvalidParameterValue	参数取值错误。

错误码	说明
InvalidRequest	请求 body 的 multipart 格式错误。
IpInBlacklist	IP地址在黑名单中。
IpNotInWhitelist	IP地址不在白名单中。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数。
NoSuchProduct	产品不存在
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
RequestLimitExceeded.GlobalRegionUinLimitExceeded	主账号超过频率限制。
RequestLimitExceeded.IPLimitExceeded	IP限频。
RequestLimitExceeded.UinLimitExceeded	主账号限频。
RequestSizeLimitExceeded	请求包超过限制大小。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
ResponseSizeLimitExceeded	返回包超过限制大小。
ServiceUnavailable	当前服务暂时不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误，用户多传未定义的参数会导致错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s) 请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
FailedOperation.AlreadyExists	已经存在同名资源。