

智能全局流量管理

API 文档



腾讯云

【版权声明】

©2013–2025 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或95716。

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

参数类型

(新)地址池相关接口

删除地址池

获取地址所属地域

地址池详情

地址池列表

修改地址池

创建地址池

配额查询

(新)实例相关接口

创建实例

实例详情

获取实例列表

修改实例配置

(新)监控器相关接口

删除监控器

新增监控器

修改监控器

获取所有监控器

查询监控器详情

(新)探测节点相关接口

获取探测节点列表

(新)策略相关接口

新建策略

删除策略

策略详情

策略列表

查询分组线路列表

修改策略

(新)套餐管理相关接口

获取实例套餐列表

探测任务套餐详情

获取探测任务套餐列表

数据结构

错误码

API 文档

更新历史

最近更新时间：2025-07-11 01:29:32

第 2 次发布

发布时间：2025-07-11 01:29:22

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [DescribeDetectPackageDetail](#)
- [DescribeDetectTaskPackageList](#)
- [DescribeInstancePackageList](#)

新增数据结构：

- [CostItem](#)
- [DetectTaskPackage](#)
- [InstancePackage](#)

第 1 次发布

发布时间：2025-06-30 11:19:19

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateAddressPool](#)
- [CreateInstance](#)
- [CreateMonitor](#)
- [CreateStrategy](#)
- [DeleteAddressPool](#)
- [DeleteMonitor](#)
- [DeleteStrategy](#)
- [DescribeAddressLocation](#)
- [DescribeAddressPoolDetail](#)
- [DescribeAddressPoolList](#)
- [DescribeDetectors](#)
- [DescribeDnsLineList](#)
- [DescribeInstanceDetail](#)
- [DescribeInstanceList](#)
- [DescribeMonitorDetail](#)
- [DescribeMonitors](#)
- [DescribeQuotas](#)
- [DescribeStrategyDetail](#)
- [DescribeStrategyList](#)
- [ModifyAddressPool](#)
- [ModifyInstanceConfig](#)
- [ModifyMonitor](#)
- [ModifyStrategy](#)

新增数据结构：

- [Address](#)
- [AddressLocation](#)

- [AddressPool](#)
- [AddressPoolDetail](#)
- [DetectorGroup](#)
- [GroupLine](#)
- [Instance](#)
- [InstanceConfig](#)
- [InstanceDetail](#)
- [InstanceInfo](#)
- [MainAddressPool](#)
- [MainPoolWeight](#)
- [MonitorDetail](#)
- [Quota](#)
- [ResourceFilter](#)
- [Source](#)
- [Strategy](#)
- [StrategyDetail](#)

简介

最近更新时间：2025-06-30 11:19:44

智能全局流量管理基于DNS技术，实现应用服务的就近访问接入和多地址负载均衡，同时对客户设定的地址进行健康检查。发现地址不可用时，能自动切换为备用地址，从而实现应用服务的同城多活、故障隔离和异地容灾。

API 概览

最近更新时间：2025-07-11 01:29:32

(新)地址池相关接口

接口名称	接口功能	频率限制（次/秒）
DeleteAddressPool	删除地址池	20
DescribeAddressLocation	获取地址所属地域	20
DescribeAddressPoolDetail	地址池详情	20
DescribeAddressPoolList	地址池列表	20
ModifyAddressPool	修改地址池	20
CreateAddressPool	创建地址池	20
DescribeQuotas	配额查询	20

(新)实例相关接口

接口名称	接口功能	频率限制（次/秒）
CreateInstance	创建实例	20
DescribeInstanceDetail	实例详情	20
DescribeInstanceList	获取实例列表	20
ModifyInstanceConfig	修改实例配置	10

(新)监控器相关接口

接口名称	接口功能	频率限制（次/秒）
DeleteMonitor	删除监控器	20
CreateMonitor	新增监控器	20
ModifyMonitor	修改监控器	20
DescribeMonitors	获取所有监控器	20
DescribeMonitorDetail	查询监控器详情	20

(新)探测节点相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeDetectors	获取探测节点列表	20

(新)策略相关接口

接口名称	接口功能	频率限制（次/秒）
CreateStrategy	新建策略	10
DeleteStrategy	删除策略	20
DescribeStrategyDetail	策略详情	20
DescribeStrategyList	策略列表	20
DescribeDnsLineList	查询分组线路列表	20
ModifyStrategy	修改策略	20

(新)套餐管理相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeInstancePackageList	获取实例套餐列表	20
DescribeDetectPackageDetail	探测任务套餐详情	20
DescribeDetectTaskPackageList	获取探测任务套餐列表	20

 注意：

以上给出的接口频率限制维度为 API + 接入地域 + 子账号，有关限频更多说明参考：[API 频率限制说明](#)

调用方式

请求结构

最近更新时间：2025-06-30 11:19:44

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 `igtm.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `igtm.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 `igtm.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）	<code>igtm.tencentcloudapi.com</code>
华南地区（广州）	<code>igtm.ap-guangzhou.tencentcloudapi.com</code>
华东地区（上海）	<code>igtm.ap-shanghai.tencentcloudapi.com</code>
华东地区（南京）	<code>igtm.ap-nanjing.tencentcloudapi.com</code>
华北地区（北京）	<code>igtm.ap-beijing.tencentcloudapi.com</code>
西南地区（成都）	<code>igtm.ap-chengdu.tencentcloudapi.com</code>
西南地区（重庆）	<code>igtm.ap-chongqing.tencentcloudapi.com</code>
港澳台地区（中国香港）	<code>igtm.ap-hongkong.tencentcloudapi.com</code>
亚太东南（新加坡）	<code>igtm.ap-singapore.tencentcloudapi.com</code>
亚太东南（雅加达）	<code>igtm.ap-jakarta.tencentcloudapi.com</code>
亚太东南（曼谷）	<code>igtm.ap-bangkok.tencentcloudapi.com</code>
亚太东北（首尔）	<code>igtm.ap-seoul.tencentcloudapi.com</code>
亚太东北（东京）	<code>igtm.ap-tokyo.tencentcloudapi.com</code>
美国东部（弗吉尼亚）	<code>igtm.na-ashburn.tencentcloudapi.com</code>
美国西部（硅谷）	<code>igtm.na-siliconvalley.tencentcloudapi.com</code>
南美地区（圣保罗）	<code>igtm.sa-saopaulo.tencentcloudapi.com</code>
欧洲地区（法兰克福）	<code>igtm.eu-frankfurt.tencentcloudapi.com</code>

注意：由于**金融区**和非金融区是隔离不通的，因此当访问金融区服务时（公共参数 `Region` 为金融区地域），需要同时指定带金融区地域的域名，最好和 `Region` 的地域保持一致。

金融区接入地域	金融区域名
华东地区（上海金融）	<code>igtm.ap-shanghai-fsi.tencentcloudapi.com</code>
华南地区（深圳金融）	<code>igtm.ap-shenzhen-fsi.tencentcloudapi.com</code>

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：2025-06-30 11:19:44

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：出于简化的目的，部分接口文档中的示例使用的是签名方法 v1 GET 请求，而不是更安全的签名方法 v3。

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。取值参考接口文档输入参数章节关于公共参数 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。取值参考接口文档中输入参数章节关于公共参数 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=AKID***/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，AKID*** 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 为具体产品名，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 igtm；tc3_request 为固定字符串； - 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*****/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*****/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*****/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章节关于公共参数 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 。

参数名称	类型	必选	描述
Version	String	是	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。
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*****

Host: cvm.tencentcloudapi.com
```

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*****
***
```

签名方法 v3

最近更新时间：2025-06-30 11:19:45

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”，可以对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 8 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++](#)、[Ruby](#)。

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：

- 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】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：

- 云服务器默认已开通，该接口很常用；
- 该接口是只读的，不会改变现有资源的状态；
- 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：AKID***** 和 *****。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=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f" \
-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 中间号 (?) 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 编码（特殊字符编码后需大写字母），字符集 UTF-8。推荐使用编程语言标准库进行编码。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入其他头部参与签名以提高自身请求的唯一性和安全性，此示例额外增加了接口名头部。 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\nx-tc-action:describeinstances\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;x-tc-action
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

字段名称	解释
	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
x-tc-action:describeinstances

content-type;host;x-tc-action
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。
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 7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84。

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，可能运行一段时间后，请求失败，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
7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84
```

3. 计算签名

1) 计算派生签名密钥，伪代码如下：

```
SecretKey = "*****"  
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，将其转为十六进制字符串打印的输出分别为：
da98fb70dcf6b112dc21038d1eeeb3a95c74b4dcb12c1131f864f6066bd02be0，
8d70cbefb03939f929db64d32dc2ba89b1095620119fe3e050e2b18c5bd2752f，
b596b923aad85185e2d1f6659d2a062e0a86731226e021e61bfe06f7ed05f5af。

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 *****。
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 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

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 AKID*****。
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;x-tc-action。
Signature	签名值。此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
```

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
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"}]}
```

⚠ 注意：

请求发送时的 HTTP 头部（Header）和请求体（Payload）必须和签名计算过程中的内容完全一致，否则会返回签名不一致错误。可以通过打印实际请求内容，网络抓包等方式对比排查。

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++](#)
- [Ruby](#)

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](#)

为了更清楚地解释签名过程，下面以实际编程语言为例，将上述的签名过程完整实现。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;
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    private final static String SECRET_ID = System.getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
}
```

```
private final static String SECRET_KEY = System.getenv("TENCENTCLOUD_SECRET_KEY");
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" + "x-tc-action:" + action.toLowerCase() + "\n";
    String signedHeaders = "content-type;host;x-tc-action";

    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
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

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": [{"Values": [u"未命名"], "Name": "instance-name"}]}

# ***** 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\nx-tc-action:%s\n" % (ct, host, action.lower())
signed_headers = "content-type;host;x-tc-action"
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"
"os"
"strings"
"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() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
    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 := fmt.Sprintf("content-type:%s\nhost:%s\nx-tc-action:%s\n",
        "application/json; charset=utf-8", host, strings.ToLower(action))
    signedHeaders := "content-type;host;x-tc-action"
    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
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$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 = "";
$canonicalHeaders = implode("\n", [
"content-type:application/json; charset=utf-8",
"host:".$host,
"x-tc-action:".strtolower($action),
```

```
""
});
$signedHeaders = implode(";", [
    "content-type",
    "host",
    "x-tc-action",
]);
$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=". $signedHeaders.", 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
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

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}\nxc-action:#{action.downcase}\n"
signed_headers = 'content-type;host;x-tc-action'
# 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, 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"
+ "x-tc-action:" + action.ToLower() + "\n";
string signedHeaders = "content-type;host;x-tc-action";
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
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

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, 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
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

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 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"
+ "x-tc-action:" + action.toLowerCase() + "\\n"
const signedHeaders = "content-type;host;x-tc-action"

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 <algorithm>
#include <cstdlib>
#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
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
string SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
string SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");

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 lower = action;
std::transform(action.begin(), action.end(), lower.begin(), ::tolower);
string canonicalHeaders = string("content-type:application/json; charset=utf-8\n")
+ "host:" + host + "\n"
+ "x-tc-action:" + lower + "\n";
string signedHeaders = "content-type;host;x-tc-action";
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;
};
```

C

```
#include <ctype.h>
#include <string.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#include <stdint.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

void get_utc_date(int64_t timestamp, char* utc, int len)
{
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(utc, len, "%Y-%m-%d", &sttime);
}

void sha256_hex(const char* str, char* result)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str, strlen(str));
    SHA256_Final(hash, &sha256);
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        snprintf(buf, sizeof(buf), "%02x", hash[i]);
        strcat(result, buf);
    }
}

void hmac_sha256(const char* key, int key_len,
const char* input, int input_len,
unsigned char* output, unsigned int* output_len)
{
    HMAC_CTX *h;
```

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

HMAC_Init_ex(h, key, key_len, EVP_sha256(), NULL);
HMAC_Update(h, ( unsigned char* )input, input_len);
HMAC_Final(h, output, output_len);

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

}

void hex_encode(const char* input, int input_len, char* output)
{
    static const char* lut = "0123456789abcdef";

    char add_out[128] = {0};
    char temp[2] = {0};
    for (size_t i = 0; i < input_len; ++i)
    {
        const unsigned char c = input[i];
        temp[0] = lut[c >> 4];
        strcat(add_out, temp);
        temp[0] = lut[c & 15];
        strcat(add_out, temp);
    }
    strncpy(output, add_out, 128);
}

void lowercase(const char * src, char * dst)
{
    for (int i = 0; src[i]; i++)
    {
        dst[i] = tolower(src[i]);
    }
}

int main()
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    const char* SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    const char* SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");
    const char* service = "cvm";
    const char* host = "cvm.tencentcloudapi.com";
    const char* region = "ap-guangzhou";
    const char* action = "DescribeInstances";
```

```
const char* version = "2017-03-12";
int64_t timestamp = 1551113065;
char date[20] = {0};
get_utc_date(timestamp, date, sizeof(date));

// ***** 步骤 1: 拼接规范请求串 *****
const char* http_request_method = "POST";
const char* canonical_uri = "/";
const char* canonical_query_string = "";
char canonical_headers[100] = {"Content-type:application/json; charset=utf-8\nhost:"};
strcat(canonical_headers, host);
strcat(canonical_headers, "\nX-tc-action:");
char value[100] = {0};
lowercase(action, value);
strcat(canonical_headers, value);
strcat(canonical_headers, "\n");
const char* signed_headers = "content-type;host;x-tc-action";
const char* payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]\"}";
char hashed_request_payload[100] = {0};
sha256_hex(payload, hashed_request_payload);

char canonical_request[256] = {0};
sprintf(canonical_request, "%s\n%s\n%s\n%s\n%s\n%s", http_request_method,
canonical_uri, canonical_query_string, canonical_headers,
signed_headers, hashed_request_payload);
printf("%s\n", canonical_request);

// ***** 步骤 2: 拼接待签名字符串 *****
const char* algorithm = "TC3-HMAC-SHA256";
char request_timestamp[16] = {0};
sprintf(request_timestamp, "%d", timestamp);
char credential_scope[64] = {0};
strcat(credential_scope, date);
sprintf(credential_scope, "%s/%s/tc3_request", date, service);
char hashed_canonical_request[100] = {0};
sha256_hex(canonical_request, hashed_canonical_request);
char string_to_sign[256] = {0};
sprintf(string_to_sign, "%s\n%s\n%s\n%s", algorithm, request_timestamp,
credential_scope, hashed_canonical_request);
printf("%s\n", string_to_sign);

// ***** 步骤 3: 计算签名 *****
char k_key[64] = {0};
sprintf(k_key, "%s", "TC3", SECRET_KEY);
unsigned char k_date[64] = {0};
unsigned int output_len = 0;
hmac_sha256(k_key, strlen(k_key), date, strlen(date), k_date, &output_len);
unsigned char k_service[64] = {0};
hmac_sha256(k_date, output_len, service, strlen(service), k_service, &output_len);
unsigned char k_signing[64] = {0};
hmac_sha256(k_service, output_len, "tc3_request", strlen("tc3_request"), k_signing, &output_len);
unsigned char k_hmac_sha_sign[64] = {0};
hmac_sha256(k_signing, output_len, string_to_sign, strlen(string_to_sign), k_hmac_sha_sign, &output_len);

char signature[128] = {0};
```

```
hex_encode(k_hmac_sha_sign, output_len, signature);
printf("%s\n", signature);

// ***** 步骤 4: 拼接 Authorization *****
char authorization[512] = {0};
sprintf(authorization, "%s Credential=%s/%s, SignedHeaders=%s, Signature=%s",
algorithm, SECRET_ID, credential_scope, signed_headers, signature);
printf("%s\n", authorization);

char curlcmd[10240] = {0};
sprintf(curlcmd, "curl -X POST https://%s\n \
-H \"Authorization: %s\"\n \
-H \"Content-Type: application/json; charset=utf-8\"\n \
-H \"Host: %s\"\n \
-H \"X-TC-Action: %s\"\n \
-H \"X-TC-Timestamp: %s\"\n \
-H \"X-TC-Version: %s\"\n \
-H \"X-TC-Region: %s\"\n \
-d '%s'",
host, authorization, host, action, request_timestamp, version, region, payload);
printf("%s\n", curlcmd);
return 0;
}
```

其他语言

- Lua: [GitHub](#)
- Swift: [GitHub](#)
- Dart: [GitHub](#)
- Shell(Bash): [GitHub](#)

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：2025-06-30 11:19:46

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 8 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++](#)、[Ruby](#)。

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：

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 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: AKID*****
- SecretKey: *****

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	AKID*****
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou
InstanceIds.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',
  'InstanceIds.0' : 'ins-09dx96dg',
  'Limit' : 20,
  'Nonce' : 11886,
  'Offset' : 0,
  'Region' : 'ap-guangzhou',
  'SecretId' : 'AKID*****',
  '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&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****
&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&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-gu
angzhou&SecretId=AKID*****&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 = '*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&R
egion=ap-guangzhou&SecretId=AKID*****&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：

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

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 7RAM2xfNMO9EiVTNmPg06MRnCvQ=，最终得到的签名串请求参数（Signature）为：

7RAM2xfNMO9EiVTNmPg06MRnCvQ%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 编码前需要先以 UTF-8 进行编码。

注意：有些编程语言的网络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
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++](#)
- [Ruby](#)

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&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-`

guangzhou&SecretId=AKID*****&Signature=7RAM2xfNMO9EiVTNmPg06MRnCvQ%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，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
        // 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKID*****
        params.put("SecretId", System.getenv("TENCENTCLOUD_SECRET_ID")); // 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("InstanceIds.0", "ins-09dx96dg"); // 业务参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
params.put("Signature", sign(getStringToSign(params), System.getenv("TENCENTCLOUD_SECRET_KEY"), "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 os
import time

import requests

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

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',
        'InstanceIds.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"
    "os"
    "sort"
    "strconv"
)

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
    params := map[string]string{
        "Nonce": "11886",
        "Timestamp": strconv.Itoa(1465185768),
        "Region": "ap-guangzhou",
        "SecretId": secretId,
        "Version": "2017-03-12",
        "Action": "DescribeInstances",
        "InstanceIds.0": "ins-09dx96dg",
        "Limit": strconv.Itoa(20),
        "Offset": strconv.Itoa(0),
    }

    var buf bytes.Buffer
    buf.WriteString("GET")
    buf.WriteString("evm.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
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$params["Nonce"] = 11886;//rand();
$params["Timestamp"] = 1465185768;//time();
$params["Region"] = "ap-guangzhou";
$params["SecretId"] = $secretId;
$params["Version"] = "2017-03-12";
$params["Action"] = "DescribeInstances";
$params["InstanceIds.0"] = "ins-09dx96dg";
$params["Limit"] = 20;
$params["Offset"] = 0;

ksort($params);

$signStr = "GETcvm.tencentcloudapi.com/?";
foreach ( $params 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
// $params["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($params);
// 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'

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
```

```
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
  'Action' => 'DescribeInstances',
  'InstanceIds.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
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

    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("InstanceIds.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'] = encodeURIComponent(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
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

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['InstanceIds.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：2025-06-30 11:19:46

云 API 3.0 接口默认返回 JSON 数据，返回非 JSON 格式的接口会在文档中做出说明。返回 JSON 数据时最大限制为 50 MB，如果返回的数据超过最大限制，请求会失败并返回内部错误。请根据接口文档中给出的过滤功能（例如时间范围）或者分页功能，控制返回数据不要过大。

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 [腾讯云客服](#) 或 [提交工单](#)，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 [腾讯云客服](#) 或 [提交工单](#)，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。

参数类型

最近更新时间：2025-06-30 11:19:46

目前腾讯云 API 3.0 输入参数和输出参数支持如下几种数据格式：

- String: 字符串。
- Integer: 整型，上限为无符号64位整数。SDK 3.0 不同编程语言支持的类型有所差异，建议以所使用编程语言的最大整型定义，例如 Golang 的 `uint64`。
- Boolean: 布尔型。
- Float: 浮点型。
- Double: 双精度浮点型。
- Date: 字符串，日期格式。例如：2022-01-01。
- Timestamp: 字符串，时间格式。例如：2022-01-01 00:00:00。
- Timestamp ISO8601: ISO 8601 是由国际标准化组织（International Organization for Standardization，ISO）发布的关于日期和时间格式的国际标准，对应国标《GB/T 7408-2005数据元和交换格式信息交换日期和时间表示法》。建议以所使用编程语言的标准库进行格式解析。例如：2022-01-01T00:00:00+08:00。
- Binary: 二进制内容，需要以特定协议请求和解析。

(新)地址池相关接口

删除地址池

最近更新时间：2025-07-22 01:51:58

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

删除地址池

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：DeleteAddressPool。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
PoolId	是	Integer	地址池id 示例值：5

3. 输出参数

参数名称	类型	描述
Msg	String	是否成功 示例值：success
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除地址池

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteAddressPool
<公共请求参数>

{
  "PoolId": 1
}
```

输出示例

```
{
  "Response": {
    "Msg": "success",
    "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.DeleteError	操作删除失败。
FailedOperation.DeletePoolFailed	删除地址池失败
FailedOperation.ExceededLimit	超出限额。
FailedOperation.PoolWasUsingError	地址池正在使用。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.PoolUnauthorized	未授权,不可操作该地址池
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation	操作不支持。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

获取地址所属地域

最近更新时间：2025-07-22 01:51:57

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

获取地址所属地域

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：DescribeAddressLocation。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Address.N	是	Array of String	地址数组 示例值：["1.2.2.1"]

3. 输出参数

参数名称	类型	描述
AddressLocation	Array of AddressLocation	所属地域
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取IP归属

```
{
  "Address": [
    "0.0.0.1"
  ]
}
```

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAddressLocation
<公共请求参数>

{
  "Address": [
    "0.0.0.1"
  ]
}
```

输出示例

```
{
  "Response": {
    "AddressLocation": [
      {

```

```
"Addr": "0.0.0.1",
"Location": "未知"
},
{
  "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
}
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InternalServerError	内部错误。
InvalidParameter	参数错误。
MissingParameter	缺少参数错误。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

地址池详情

最近更新时间：2025-07-22 01:51:57

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

地址池详情

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：DescribeAddressPoolDetail。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
PoolId	是	Integer	地址池id 示例值：5

3. 输出参数

参数名称	类型	描述
AddressPool	AddressPoolDetail	资源组详情描述
AddressSet	Array of Address	资源组中的资源列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 地址池详情

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAddressPoolDetail
<公共请求参数>

{
  "PoolId": 1
}
```

输出示例

```
{
  "Response": {
    "AddressPool": {
      "PoolId": 1,
      "PoolName": "测试地址池",
      "TrafficStrategy": "ALL",
      "MonitorId": 1,
      "CreatedOn": "2020-09-22T00:00:00+00:00",
      "UpdatedOn": "2020-09-22T00:00:00+00:00"
    },
    "AddressSet": [
      {
        "AddressId": 1,
        "Addr": "1.1.1.2",
        "Location": "上好电信",
        "Status": "OK",
        "IsEnable": "ENABLED",
        "Weight": 1,
        "CreatedOn": "2020-09-22T00:00:00+00:00",
        "UpdatedOn": "2020-09-22T00:00:00+00:00"
      }
    ],
    "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。

错误码	描述
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.PoolUnauthorized	未授权,不可操作该地址池
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

地址池列表

最近更新时间：2025-07-22 01:51:57

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

地址池列表

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：DescribeAddressPoolList。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Filters.N	否	Array of ResourceFilter	告警过滤条件：PoolName：地址池名称；MonitorId：监控器id 示例值：[{"Name": "MonitorName", "Value": ["监控器001"]}, {"Name": "MonitorId", "Value": ["123"]}]
Offset	否	Integer	页数 示例值：1
Limit	否	Integer	每页数 示例值：10

3. 输出参数

参数名称	类型	描述
AddressPoolSet	Array of AddressPool	资源组列表
TotalCount	Integer	总数 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 地址池列表

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeAddressPools
<公共请求参数>
```

```
{
  "Filters": [
    {
      "Name": "PoolName",
      "Value": [
        "gtm-dnspod"
      ],
      "Fuzzy": true
    }
  ],
  "Offset": 1,
  "Limit": 1
}
```

输出示例

```
{
  "Response": {
    "AddressPoolSet": [
      {
        "PoolId": 1,
        "PoolName": "gtm-dnspod",
        "TrafficStrategy": "WEIGHT",
        "MonitorId": 1,
        "Status": "WARN",
        "AddressNum": 0,
        "MonitorGroupNum": 0,
        "MonitorTaskNum": 0,
        "AddressSet": [
          {
            "AddressId": 1,
            "Addr": "1.1.1.2",
            "Location": "上海电信",
            "Status": "UNMONITORED",
            "IsEnable": "ENABLED",
            "Weight": 1,
            "CreatedOn": "2020-09-22T00:00:00+00:00",
            "UpdatedOn": "2020-09-22T00:00:00+00:00"
          }
        ],
        "CreatedOn": "2020-09-22T00:00:00+00:00",
        "UpdatedOn": "2020-09-22T00:00:00+00:00"
      }
    ],
    "TotalCount": 1,
    "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.QueryNotAllow	不支持的查询
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

修改地址池

最近更新时间：2025-07-22 01:51:56

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

修改地址池

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：ModifyAddressPool。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
PoolId	是	Integer	地址池id 示例值：5
PoolName	否	String	地址池名称，不允许重复 示例值：addressGroup
TrafficStrategy	否	String	流量策略：WEIGHT负载均衡，ALL解析全部 示例值：ALL
MonitorId	否	Integer	监控器id 示例值：1
AddressSet.N	否	Array of Address	地址列表 示例值：[{"AddressId":100,"Addr":"1.1.1.2","Location":"中国上海","IsEnable":"ENABLED","Weight":99}]

3. 输出参数

参数名称	类型	描述
Msg	String	是否修改成功 示例值：success
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 编辑地址池

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyAddressPool
```

<公共请求参数>

```
{
  "PoolId": 1,
  "PoolName": "testname",
  "TrafficStrategy": "all",
  "MonitorId": 1,
  "AddressSet": [
    {
      "AddressId": 1,
      "Addr": "1.1.1.2",
      "Location": "中国上海",
      "Status": "ok",
      "IsEnable": "ENABLED",
      "Weight": 1,
      "CreatedOn": "2020-09-22T00:00:00+00:00",
      "UpdatedOn": "2020-09-22T00:00:00+00:00"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "Msg": "success",
    "RequestId": "9b19115c-873gt2-4940-91339-98952a13f159"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.AddrExceededLimit	地址池中地址数量超出限额
FailedOperation.ExceededLimit	超出限额。
FailedOperation.ExceededMonitorNumLimit	探点数配额不足。
FailedOperation.ModifyError	修改错误。
FailedOperation.ModifyPoolFailed	修改地址池失败
FailedOperation.PoolExceededLimit	地址池数量超出限额
FailedOperation.TaskExceededLimit	探测任务数量超出限额
InternalError	内部错误。
InternalError.UndefinedError	错误未定义
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.PoolAddrDuplicated	地址池地址重复
InvalidParameterValue.PoolAddrEmpty	地址池地址不能为空
InvalidParameterValue.PoolAddrInvalid	地址池地址非法
InvalidParameterValue.PoolAddrMixed	地址池地址不能混用
InvalidParameterValue.PoolAddrWeightEmpty	地址池地址权重不能为空
MissingParameter	缺少参数错误。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.PoolAddressUnauthorized	未授权,不可操作该地址池地址
UnauthorizedOperation.PoolUnauthorized	未授权,不可操作该地址池
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.PoolNameDuplicate	地址池名称不能重复
UnsupportedOperation.ResourceNameDuplicated	资源名不能重复。
UnsupportedOperation.UserNotFound	用户未登录, 缺少Uin参数。

创建地址池

最近更新时间：2025-07-22 01:51:58

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

创建地址池

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：CreateAddressPool。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
PoolName	是	String	地址池名称，不允许重复 示例值：GroupName
TrafficStrategy	是	String	流量策略：WEIGHT负载均衡，ALL解析所有健康地址 示例值：ALL
AddressSet.N	是	Array of Address	地址列表 示例值：[{"Addr": "2.2.2.2"}]
MonitorId	否	Integer	监控器id 示例值：1

3. 输出参数

参数名称	类型	描述
AddressPoolId	Integer	地址池id 示例值：5
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建地址池

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateAddressPool
<公共请求参数>

{
```

```
"PoolName": "测试地址池",
  "TrafficStrategy": "ALL",
  "MonitorId": 1,
  "AddressSet": [
    {
      "AddressId": 1,
      "Addr": "1.1.1.2",
      "Location": "上海电信",
      "Status": "",
      "IsEnable": "ENABLED"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "AddressPoolId": 1,
    "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.AddrExceededLimit	地址池中地址数量超出限额
FailedOperation.CreateError	创建失败。

错误码	描述
FailedOperation.CreatePoolFailed	创建地址池失败
FailedOperation.ExceededLimit	超出限额。
FailedOperation.ModifyError	修改错误。
FailedOperation.PoolExceededLimit	地址池数量超出限额
FailedOperation.TaskExceededLimit	探测任务数量超出限额
InternalError	内部错误。
InternalError.DbErr	数据库读写错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.PoolAddrDuplicated	地址池地址重复
InvalidParameterValue.PoolAddrEmpty	地址池地址不能为空
InvalidParameterValue.PoolAddrInvalid	地址池地址非法
InvalidParameterValue.PoolAddrMixed	地址池地址不能混用
InvalidParameterValue.PoolAddrWeightEmpty	地址池地址权重不能为空
MissingParameter	缺少参数错误。
ResourceNotFound	资源不存在。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation	操作不支持。
UnsupportedOperation.PoolNameDuplicate	地址池名称不能重复
UnsupportedOperation.ResourceNameDuplicated	资源名不能重复。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

配额查询

最近更新时间：2025-07-21 01:27:54

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

配额查询

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：DescribeQuotas。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
AccessDomain	否	String	接入域名 示例值：gtm-access.com

3. 输出参数

参数名称	类型	描述
Quotas	Quota	配额id 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 配额

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeQuotas
<公共请求参数>

{}
```

输出示例

```
{
  "Response": {
    "Quotas": {
      "TaskQuota": 1,
```

```
"PoolQuota": 1,
"AddressQuota": 1,
"MonitorQuota": 1,
"MessageQuota": 1
},
"RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
}
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

(新)实例相关接口

创建实例

最近更新时间：2025-07-25 01:26:57

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com 。

创建实例接口，仅供免费实例使用

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：CreateInstance。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Domain	是	String	业务域名 示例值：mainDomain.com
AccessType	是	String	CUSTOM: 自定义接入域名 SYSTEM: 系统接入域名 示例值：CUSTOM
GlobalTtl	是	Integer	解析生效时间 示例值：600
PackageType	是	String	套餐类型 FREE: 免费版 STANDARD: 标准版 ULTIMATE: 旗舰版 示例值：FREE
InstanceName	否	String	实例名称 示例值：实例名称
AccessDomain	否	String	接入主域名 示例值：rootDomain.com
AccessSubDomain	否	String	接入子域名 示例值：subDomain
Remark	否	String	备注 示例值：备注信息
ResourceId	否	String	套餐资源id 示例值：ins-sdsdad12x

3. 输出参数

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需

参数名称	类型	描述
		要提供该次请求的 RequestId。

4. 示例

示例1 创建免费版套餐

创建免费版套餐

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateInstance
<公共请求参数>

{
  "InstanceName": "example-test-instance",
  "Domain": "igtm.example.com",
  "AccessType": "custom",
  "AccessDomain": "example.com",
  "AccessSubDomain": "access.igtm",
  "GlobalTtl": 300,
  "PackageType": "Free",
  "Remark": "测试实例"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "ec040ddf-7c7d-4aef-9e0d-6f3105fa0a4c"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CreateError	创建失败。
FailedOperation.CreateInstanceFailed	创建实例失败
FailedOperation.QueryErr	查询失败
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.AccessTypeInvalid	实例接入类型不合法
InvalidParameter.GlobalAccessDomainConflict	全局接入域名冲突，当前域名被其他实例使用，请解决占用后重试
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.AccessDomainUnauthorized	接入域名未使用腾讯云 DNSPod 解析，请重新输入
UnsupportedOperation.DomainTtlInvalid	接入域名的套餐不允许此TTL值
UnsupportedOperation.InstanceNameDuplicate	实例名称不能重复
UnsupportedOperation.PackageUnauthorized	套餐无效，请选择正确套餐
UnsupportedOperation.TTLInvalid	TTL无效,请设置正确的TTL
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

实例详情

最近更新时间：2025-07-22 01:51:56

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

实例详情

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：DescribeInstanceDetail。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
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
Instance	InstanceDetail	实例详情
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 实例详情

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeInstanceDetail
<公共请求参数>

{}
```

输出示例

```
{
  "Response": {
    "Instance": {
      "InstanceId": "gtm-dsdd123xdo",
      "InstanceName": "测试实例",
      "Domain": "gtmtest.com",
      "AccessType": "CUSTOM",
```

```
"AccessSubDomain": "access-gtm",
"AccessDomain": "gtmtest.com",
"GlobalTtl": 600,
"PackageType": "ULTIMATE",
"WorkingStatus": "NORMAL",
"Status": "ENABLED",
"IsCnameConfigured": false,
"Remark": "备注信息",
"StrategyNum": 0,
"AddressPoolNum": 0,
"MonitorNum": 0,
"ResourceId": "ins-oxrvmmnl4yrd",
"NotifyEventSet": [],
"CreatedOn": "2024-07-19 15:40:19",
"UpdatedOn": "2024-07-19 15:40:19"
},
"RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
}
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
RequestLimitExceeded.InnerRequestLimitExceeded	当前请求过于频繁，请稍后重试

错误码	描述
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

获取实例列表

最近更新时间：2025-07-22 01:51:55

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

实例列表

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：DescribeInstanceList。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Offset	否	Integer	分页偏移量 示例值：0
Limit	否	Integer	分页数 示例值：20
Filters.N	否	Array of ResourceFilter	Name: "实例名称" 模糊查询Domain: "域名" 模糊查询MonitorId: "监控器 id" PoolId: "地址池id", AccessDomain接入主域名

3. 输出参数

参数名称	类型	描述
InstanceSet	Array of Instance	实例列表
TotalCount	Integer	列表总数 示例值：1
SystemAccessEnabled	Boolean	是否支持系统域名接入：true支持；false不支持 示例值：false
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 实例列表

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeInstanceList
<公共请求参数>
```

```
{
  "Offset": 1,
  "Limit": 1,
  "Filters": [
    {
      "Name": "Domain",
      "Value": [
        "gtmtest.com"
      ],
      "Fuzzy": true
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "InstanceSet": [
      {
        "InstanceId": "gtm-dsdd123xdo",
        "InstanceName": "gtm测试实例",
        "Domain": "gtmtest.com",
        "AccessType": "CUSTOM",
        "AccessDomain": "igtm-access",
        "AccessSubDomain": "gtmtest.com",
        "GlobalTtl": 600,
        "PackageType": "ULTIMATE",
        "WorkingStatus": "UNKNOWN",
        "Status": "ENABLED",
        "Remark": "测试实例",
        "StrategyNum": 0,
        "AddressPoolNum": 0,
        "MonitorNum": 0,
        "CreatedOn": "2024-07-19 15:40:19",
        "UpdatedOn": "2024-07-19 15:40:19",
        "ResourceId": "ins-oxrvmn14yrd"
      }
    ],
    "TotalCount": 1,
    "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CreateError	创建失败。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
RequestLimitExceeded.InnerRequestLimitExceeded	当前请求过于频繁，请稍后重试
ResourceNotFound	资源不存在。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

修改实例配置

最近更新时间：2025-06-30 11:19:39

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

修改实例配置

默认接口请求频率限制：10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：ModifyInstanceConfig。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
InstanceConfig	否	InstanceConfig	实例配置详情

3. 输出参数

参数名称	类型	描述
Instance	InstanceDetail	实例详情 示例值：实例详情
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改实例配置

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyInstanceConfig
<公共请求参数>

{
  "InstanceConfig": {
    "InstanceName": "测试实例 1",
    "Domain": "xxxx.xxxx.xxxxxxxx@exmaple.com",
    "AccessType": "CUSTOM",
    "AccessDomain": "exmaple-access.com",
    "AccessSubDomain": "xxxx.xxxx.xxxxxxxx.access",
    "Remark": "测试实例",
    "GlobalTtl": 300
  }
}
```

```
}  
}
```

输出示例

```
{  
  "Response": {  
    "Instance": {  
      "InstanceName": "测试实例 1",  
      "InstanceId": "gtm-xxxxxxxxxx",  
      "ResourceId": "",  
      "Domain": "xxxx.xxxx.xxxxxxx@exmaple.com",  
      "AccessType": "CUSTOM",  
      "AccessDomain": "exmaple-access.com",  
      "AccessSubDomain": "xxxx.xxxx.xxxxxxx.access",  
      "GlobalTtl": 300,  
      "PackageType": "FREE",  
      "WorkingStatus": "NORMAL",  
      "Status": "ENABLED",  
      "Remark": "测试实例",  
      "StrategyNum": 0,  
      "AddressPoolNum": 0,  
      "MonitorNum": 0,  
      "CreatedOn": "2023-12-20 06:43:16",  
      "UpdatedOn": "2023-12-26 20:00:03",  
      "NotifyEventSet": []  
    },  
    "RequestId": "be7d3d39-82cf-41e8-9b60-151e9745c422"  
  }  
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.ModifyInstanceFailed	修改实例失败
FailedOperation.QueryErr	查询失败
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.AccessTypeInvalid	实例接入类型不合法
InvalidParameter.GlobalAccessDomainConflict	全局接入域名冲突，当前域名被其他实例使用，请解决占用后重试
InvalidParameter.InstanceDisabled	实例不可用，请检查实例状态
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.AccessDomainUnauthorized	接入域名未使用腾讯云 DNSPod 解析，请重新输入
UnsupportedOperation.DomainTtlInvalid	接入域名的套餐不允许此TTL值
UnsupportedOperation.InstanceNameDuplicate	实例名称不能重复

(新)监控器相关接口

删除监控器

最近更新时间：2025-07-22 01:51:55

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

删除监控器

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：DeleteMonitor。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
MonitorId	是	Integer	监控器id 示例值：1

3. 输出参数

参数名称	类型	描述
Msg	String	成功返回 示例值：success
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除监控器

删除监控器

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteMonitor
<公共请求参数>

{
  "MonitorId": 1
}
```

输出示例

```
{
  "Response": {
    "Msg": "success",
    "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.DeleteError	操作删除失败。
FailedOperation.DeleteMonitorFailed	删除监控器失败
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

新增监控器

最近更新时间：2025-07-22 01:51:55

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

新增监控器

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：CreateMonitor。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
MonitorName	是	String	监控器名称 示例值：监控器1
CheckProtocol	是	String	探测协议，可选值 PING TCP HTTP HTTPS 示例值：PING
CheckInterval	是	Integer	检查间隔（秒），可选值有15 60 120 300 示例值：60
Timeout	是	Integer	超时时间，单位秒，可选值为2 3 5 10 示例值：5
FailTimes	是	Integer	重试次数，可选值为 0，1，2 示例值：1
FailRate	是	Integer	失败比例，取值为 20 30 40 50 60 70 80 100，默认值50 示例值：50
DetectorStyle	是	String	监控节点类型，可选值有AUTO INTERNAL OVERSEAS IPV6 ALL 示例值：AUTO
DetectorGroupIds.N	是	Array of Integer	探测器组id列表以,分隔 示例值：[5,6,7]
PingNum	否	Integer	PING 包数目，当CheckProtocol=ping时必填，可选值有20 50 100 示例值：20
TcpPort	否	Integer	检查端口，可选值在1-65535之间 示例值：80
Host	否	String	Host 设置，默认为业务域名 示例值：www.dnspod.com
Path	否	String	URL 路径，默认为 “/” 示例值：/index.html
ReturnCodeThreshold	否	Integer	返回错误码阈值，可选值为 400 和 500，默认值500 示例值：500
EnableRedirect	否	String	跟随 3XX 重定向，不开启为 DISABLED，开启为 ENABLED，默认不开启 示例值：DISABLED

参数名称	必选	类型	描述
EnableSni	否	String	启用 SNI，不开启为 DISABLED， 开启为 ENABLED，默认不开启 示例值：DISABLED
PacketLossRate	否	Integer	丢包率告警阈值，当CheckProtocol=ping时必须取值10 30 50 80 90 100 示例值：50
ContinuePeriod	否	Integer	持续周期数，可选值1-5 示例值：1

3. 输出参数

参数名称	类型	描述
MonitorId	Integer	监控器id 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 tcp监控

tcp监控

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateMonitor
<公共请求参数>

{
  "MonitorName": "监控tcp",
  "CheckProtocol": "TCP",
  "CheckInterval": 60,
  "Timeout": 2,
  "FailTimes": 0,
  "FailRate": 50,
  "DetectorStyle": "AUTO",
  "DetectorGroupIds": [
    1
  ],
  "TcpPort": 80
}
```

输出示例

```
{
  "Response": {
    "MonitorId": 92,
    "RequestId": "ea6eaf52-428b-4d1d-b7d7-1712cf508124"
  }
}
```

示例2 http监控

http监控

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateMonitor
<公共请求参数>

{
  "MonitorName": "监控http",
  "CheckProtocol": "HTTP",
  "CheckInterval": 60,
  "Timeout": 2,
  "FailTimes": 0,
  "FailRate": 50,
  "DetectorStyle": "AUTO",
  "DetectorGroupIds": [
    1
  ],
  "TcpPort": 80,
  "ReturnCodeThreshold": 400,
  "EnableRedirect": "DISABLED",
  "EnableSni": "DISABLED"
}
```

输出示例

```
{
  "Response": {
    "MonitorId": 93,
    "RequestId": "506ffb0f-a5d4-4b8e-8ce8-14e512ec731e"
  }
}
```

示例3 新增监控器

新增监控器

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateMonitor
<公共请求参数>

{
  "MonitorName": "监控器1",
  "DetectorGroupIds": [
    5,
    6,
    7
  ],
  "CheckInterval": 60,
  "PacketLossRate": 20,
```

```
"CheckProtocol": "HTTP",
"FailRate": 20,
"TcpPort": 80,
"EnableSni": "DISABLED",
"PingNum": 10,
"ReturnCodeThreshold": 500,
"Host": "test.com",
"Timeout": 50,
"Path": "/index.html",
"FailTimes": 2,
"DetectorStyle": "AUTO"
}
```

输出示例

```
{
  "Response": {
    "MonitorId": 1,
    "RequestId": "ea6eaf52-428b-4d1d-b7d7-1712cf508124"
  }
}
```

示例4 监控器ping协议

监控器ping协议

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateMonitor
<公共请求参数>

{
  "MonitorName": "监控ping",
  "CheckProtocol": "PING",
  "CheckInterval": 60,
  "Timeout": 2,
  "FailTimes": 0,
  "FailRate": 50,
  "DetectorStyle": "AUTO",
  "DetectorGroupIds": [
    1
  ],
  "PingNum": 20,
  "PacketLossRate": 10
}
```

输出示例

```
{
  "Response": {
    "MonitorId": 91,
    "RequestId": "93ba54c1-a533-42bb-aab5-458aef4b1666"
  }
}
```

```
}  
  
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CreateError	创建失败。
FailedOperation.CreateMonitorFailed	创建监控器失败
FailedOperation.ExceededLimit	超出限额。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

修改监控器

最近更新时间：2025-07-22 01:51:54

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

修改监控器

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：ModifyMonitor。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
MonitorId	是	Integer	监控器id 示例值：1
MonitorName	是	String	监控器名称 示例值：监控器1
CheckProtocol	是	String	检查协议，可选值 PING TCP HTTP HTTPS 示例值：PING
CheckInterval	是	Integer	检查间隔（秒），可选值 15 60 120 300 示例值：60
Timeout	是	Integer	超时时间，单位:秒，可选值 2 3 5 10 示例值：5
FailTimes	是	Integer	重试次数，可选值为 0，1，2 示例值：2
FailRate	是	Integer	失败比例，可选值为 20 30 40 50 60 70 80 100，默认值为50 示例值：50
DetectorStyle	是	String	监控节点类型，可选值有AUTO INTERNAL OVERSEAS IPV6 ALL 示例值：AUTO
DetectorGroupIds.N	是	Array of Integer	探测器组id列表 示例值：[5,6,7]
PingNum	否	Integer	PING 包数目，当CheckProtocol=ping时必须填，可选值 20 50 100 示例值：10
TcpPort	否	Integer	检查端口，可选值为1-65535之间的整数 示例值：80
Host	否	String	Host 设置，默认为业务域名 示例值：www.dnspod.com
Path	否	String	URL 路径，默认为 “/” 示例值：/index.html
ReturnCodeThreshold	否	Integer	返回错误码阈值，可选值为 400 和 500，默认值500 示例值：500

参数名称	必选	类型	描述
EnableRedirect	否	String	跟随 3XX 重定向，不开启为 DISABLED，开启为 ENABLED，默认不开启 示例值：DISABLED
EnableSni	否	String	启用 SNI，不开启为 DISABLED，开启为 ENABLED，默认不开启 示例值：DISABLED
PacketLossRate	否	Integer	丢包率告警阈值，当CheckProtocol=ping时必须填，取值在10 30 50 80 90 100 示例值：50
ContinuePeriod	否	Integer	持续周期数，可选值1-5 示例值：1

3. 输出参数

参数名称	类型	描述
Msg	String	success 为修改成功 示例值：success
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改监控器

修改监控器

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyMonitor
<公共请求参数>

{
  "MonitorId": 1,
  "MonitorName": "测试监控器",
  "CheckProtocol": "PING",
  "CheckInterval": 1,
  "Timeout": 1,
  "FailTimes": 1,
  "FailRate": 1,
  "DetectorStyle": "AUTO",
  "DetectorGroupIds": [
    1
  ],
  "PingNum": 1,
  "TcpPort": 1,
  "Host": "url",
  "Path": "/path",
  "ReturnCodeThreshold": 1,
  "EnableRedirect": "DISABLED",
  "EnableSni": "DISABLED",
  "PacketLossRate": 1
}
```

输出示例

```
{
  "Response": {
    "Msg": "success",
    "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.ExceededLimit	超出限额。
FailedOperation.ExceededMonitorNumLimit	探点数配额不足。
FailedOperation.ModifyError	修改错误。
FailedOperation.ModifyMonitorFailed	修改监控器失败
InternalError	内部错误。
InternalError.ParameterError	参数格式错误
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.MonitorNameExist	监控器名已存在

错误码	描述
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

获取所有监控器

最近更新时间：2025-07-22 01:51:54

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

获取所有监控器

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：DescribeMonitors。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Offset	否	Integer	分页，偏移值 示例值：0
Limit	否	Integer	分页，当前分页记录数 示例值：10
IsDetectNum	否	Integer	是否查探测次数0否1是 示例值：0

3. 输出参数

参数名称	类型	描述
MonitorDataSet	Array of MonitorDetail	监控器列表
TotalCount	Integer	数量 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取所有监控器

获取所有监控器

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeMonitors
<公共请求参数>

{
  "Offset": 0,
  "Limit": 1
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "MonitorDataSet": [
      {
        "MonitorId": 1,
        "MonitorName": "测试监控器",
        "Uin": "10012894645",
        "DetectorGroupIds": [
          1
        ],
        "CheckProtocol": "HTTP",
        "CheckInterval": 1,
        "PingNum": 1,
        "TcpPort": 1,
        "Host": "igtmtest.com",
        "Path": "/path",
        "ReturnCodeThreshold": 1,
        "EnableRedirect": "DISABLED",
        "EnableSni": "DISABLED",
        "PacketLossRate": 50,
        "Timeout": 15,
        "FailTimes": 4,
        "FailRate": 100,
        "CreatedOn": "2025-02-25 20:24:49",
        "UpdatedOn": "2025-02-25 20:24:49",
        "DetectorStyle": "INTERNAL",
        "DetectNum": 0,
        "ContinuePeriod": 1
      }
    ],
    "RequestId": "398db591-8793-4a0f-aaa3-3a0131494184"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation	操作不支持。
UnsupportedOperation.QueryNotAllow	不支持的查询

查询监控器详情

最近更新时间：2025-07-21 01:27:53

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

查询监控器详情接口

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：DescribeMonitorDetail。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
MonitorId	是	Integer	监控器id 示例值：1

3. 输出参数

参数名称	类型	描述
MonitorDetail	MonitorDetail	探测规则
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询监控器详情

查询监控器详情

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeHealthyCheckRule
<公共请求参数>

{
  "MonitorId": 1
}
```

输出示例

```
{
  "Response": {
    "MonitorDetail": {
      "MonitorId": 1,
      "MonitorName": "测试监控器",
      "Uin": "10012894645",
      "DetectorGroupIds": [
        1
      ],
      "CheckProtocol": "HTTP",
      "CheckInterval": 300,
      "PingNum": 20,
      "TcpPort": 80,
      "Host": "gtmtest.com",
      "Path": "/path",
      "ReturnCodeThreshold": 500,
      "EnableRedirect": "DISABLED",
      "EnableSni": "DISABLED",
      "PacketLossRate": 50,
      "Timeout": 15,
      "FailTimes": 1,
      "FailRate": 100,
      "CreatedOn": "2023-01-1 11:11:11",
      "UpdatedOn": "2023-01-1 11:11:11",
      "DetectorStyle": "INTERNAL",
      "DetectNum": 0,
      "ContinuePeriod": 1
    },
    "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

(新)探测节点相关接口

获取探测节点列表

最近更新时间：2025-06-30 11:19:38

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com 。

获取探测节点列表接口

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：DescribeDetectors。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
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
DetectorGroupSet	Array of DetectorGroup	探测器组列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取探测节点列表

获取探测节点列表

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeDetectors
<公共请求参数>

{}
```

输出示例

```
{
  "Response": {
    "RequestId": "xxx",
```

```
"DetectorGroupSet": [
{
"Gid": 30,
"InternetFamily": "ipv4",
"GroupType": "bgp",
"GroupName": "广州",
"PackageSet": [
"FREE",
"STANDARD",
"ULTIMATE"
]
}
]
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。

错误码	描述
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

(新)策略相关接口

新建策略

最近更新时间：2025-07-22 01:51:54

- - - - -

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

新建策略接口

默认接口请求频率限制：10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：CreateStrategy。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
InstanceId	是	String	实例id 示例值：gtm-Ul65AOsdfDvxVD
StrategyName	是	String	策略名称，不允许重复 示例值：policy_name
Source.N	是	Array of Source	解析线路 示例值：[{"Name": "默认", "GroupLineId": 1}]
MainAddressPoolSet.N	是	Array of MainAddressPool	主力地址池集合，最多允许配置四级 示例值：[{"AddressPools": [{"PoolId": 1, "Weight": 50}], "MinAliveNodeNum": 1, "TrafficStrtegy": "weight"}, {"AddressPools": [{"PoolId": 3, "Weight": 30}], "MinAliveNodeNum": 1, "TrafficStrtegy": "weight"}, {"AddressPools": [{"PoolId": 3}], "MinAliveNodeNum": 1, "TrafficStrtegy": "all"}]
FallbackAddressPoolSet.N	是	Array of MainAddressPool	兜底地址池集合，只允许配置一级，且地址池数量为1 示例值：[{"AddressPools": [{"PoolId": 1}]}]
KeepDomainRecords	否	String	是否开启策略强制保留默认线路 disabled, enabled，默认不开启且只有一个策略能开启 示例值：enabled
SwitchPoolType	否	String	策略调度模式：AUTO默认切换；STOP仅暂停不切换 示例值：AUTO

3. 输出参数

参数名称	类型	描述
StrategyId	Integer	新增策略id 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建策略

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: CreateStrategy
```

<公共请求参数>

```
{
  "InstanceId": "gtm-sdsderd12e",
  "StrategyName": "testname",
  "Source": [
    {
      "DnsLineId": 1,
      "Name": "line1"
    }
  ],
  "MainAddressPoolSet": [
    {
      "MainAddressPoolId": 1,
      "AddressPools": [
        {
          "PoolId": 1,
          "Weight": 1
        }
      ],
      "MinSurviveNum": 1,
      "TrafficStrategy": "all"
    }
  ],
  "FallbackAddressPoolSet": [
    {
      "MainAddressPoolId": 1,
      "AddressPools": [
        {
          "PoolId": 1,
          "Weight": 1
        }
      ],
      "MinSurviveNum": 1,
      "TrafficStrategy": "all"
    }
  ],
  "KeepDomainRecords": "DISABLED"
}
```

输出示例

```
{
  "Response": {
    "StrategyId": 0,
    "RequestId": "abcasd232dsad1dsadasd"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CreateError	创建失败。
FailedOperation.CreateStrategyFailed	创建策略失败
FailedOperation.ExceededLimit	超出限额。
FailedOperation.ModifyError	修改错误。
InternalServerError	内部错误。
InternalServerError.DbErr	数据库读写错误。
InvalidParameter	参数错误。
InvalidParameter.InstanceDisabled	实例不可用，请检查实例状态
InvalidParameter.TrafficStrategyInvalid	流量策略不合法
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TrafficStrategyInvalid	流量策略不合法
InvalidParameterValue.TrafficStrategyWeightEmpty	流量策略为负载均衡且主力集合内有多个地址池时，权重不能为空
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
ResourceInsufficient.NoInstance	没有实例,请先购买实例
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.InstanceUnauthorized	未授权,不可操作该实例
UnauthorizedOperation.MainBindPoolUnauthorized	未授权,主力集合中包含未授权地址池
UnauthorizedOperation.MainPoolUnauthorized	未授权,不可操作该主力地址池

错误码	描述
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnknownParameter	未知参数错误。
UnsupportedOperation.FallBackPoolInvalid	兜底地址池格式不合法
UnsupportedOperation.MainBindPoolDuplicate	主力地址池重复
UnsupportedOperation.MainBindPoolEmpty	主力地址池为空
UnsupportedOperation.MainBindPoolLevelExceed	主力地址集合数量超过限制
UnsupportedOperation.MainBindPoolMixed	主力地址集合内地址类型不能混用
UnsupportedOperation.MainBindPoolNumInvalid	主力地址集合数量不合法
UnsupportedOperation.MainPoolDuplicate	主力地址集合重复
UnsupportedOperation.MainPoolEmpty	主力集合不能为空
UnsupportedOperation.MainPoolSurviveNumInvalid	主力地址池阈值不合法
UnsupportedOperation.ResourceNameDuplicated	资源名不能重复。
UnsupportedOperation.ResourcePoolSourceNotExist	主备资源组中资源不存在。
UnsupportedOperation.ResourceSourceDuplicated	存在重复的解析来源。
UnsupportedOperation.StrategyNameDuplicate	策略名称不能重复
UnsupportedOperation.StrategySourceDuplicate	策略线路重复
UnsupportedOperation.StrategySourceInUsed	策略线路已使用
UnsupportedOperation.StrategySourceUnauthorized	策略线路不在实例套餐中
UnsupportedOperation.StrategySwitchTypeInvalid	策略切换类型不合法
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

删除策略

最近更新时间：2025-07-22 01:51:53

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

删除策略接口

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：DeleteStrategy。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
StrategyId	是	Integer	策略id 示例值：5
InstanceId	是	String	实例id 示例值：gtm-Ul65AOsdfDvxVD

3. 输出参数

参数名称	类型	描述
Msg	String	是否成功 示例值：success
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除策略

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DeleteStrategy
<公共请求参数>

{
  "StrategyId": 1,
  "InstanceId": "gtm-ddgasdwe"
}
```

输出示例

```
{
  "Response": {
    "Msg": "success",
    "RequestId": "9b19115c-8732-4940-9199-98952a13dsf15"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CreateError	创建失败。
FailedOperation.DeleteError	操作删除失败。
FailedOperation.DeleteStrategyFailed	删除策略失败
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.InstanceDisabled	实例不可用，请检查实例状态
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceInsufficient.NoInstance	没有实例,请先购买实例
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.InstanceUnauthorized	未授权,不可操作该实例
UnauthorizedOperation.StrategyUnauthorized	未授权,不可操作该策略

错误码	描述
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

策略详情

最近更新时间：2025-07-22 01:51:53

-

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

策略详情

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：DescribeStrategyDetail。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
InstanceId	是	String	实例id 示例值：gtm-dsdd123xdo
StrategyId	是	Integer	策略 id 示例值：1

3. 输出参数

参数名称	类型	描述
StrategyDetail	StrategyDetail	策略详情
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 策略详情

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeStrategyDetail
<公共请求参数>

{
  "InstanceId": "gtm-dsdd123xdo",
  "StrategyId": 1
}
```

输出示例

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

```
"InstanceId": "gtm-dsdd123xdo",
"StrategyId": 1,
"Name": "测试策略",
"Source": [
{
"DnsLineId": 1,
"Name": "默认"
}
],
"MainAddressPoolSet": [
{
"MainAddressPoolId": 1,
"AddressPools": [
{
"PoolId": 1,
"Weight": 1
}
],
"MinSurviveNum": 1,
"TrafficStrategy": "WEIGHT"
}
],
"FallbackAddressPoolSet": [
{
"MainAddressPoolId": 1,
"AddressPools": [
{
"PoolId": 1,
"Weight": 1
}
],
"MinSurviveNum": 1,
"TrafficStrategy": "WEIGHT"
}
],
"KeepDomainRecords": "DISABLED",
"CreatedOn": "2020-09-22T00:00:00+00:00",
"UpdatedOn": "2020-09-22T00:00:00+00:00"
},
"RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
}
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.StrategyUnauthorized	未授权,不可操作该策略
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

策略列表

最近更新时间：2025-07-22 01:51:53

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com 。

策略列表接口

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：DescribeStrategyList。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
InstanceId	是	String	实例 id 示例值：gtm-Ul65AOsdfDvxVD
Offset	否	Integer	偏移量 示例值：10
Limit	否	Integer	每页条数 示例值：10
Filters.N	否	Array of ResourceFilter	策略过滤条件：StrategyName：策略名称 示例值：[{"Name": "StrategyName", "Value": ["策略1"]}]

3. 输出参数

参数名称	类型	描述
StrategySet	Array of Strategy	策略列表
TotalCount	Integer	总数 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 策略列表

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeStrategies
<公共请求参数>
```

```
{
  "InstanceId": "gtm-dsdd123xdo",
  "Offset": 1,
  "Limit": 1,
  "Filters": [
    {
      "Name": "StrategyName",
      "Value": [
        "test-strategy"
      ],
      "Fuzzy": true
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "StrategySet": [
      {
        "InstanceId": "gtm-dsdd123xdo",
        "StrategyId": 1,
        "Name": "test-strategy",
        "Source": [
          {
            "DnsLineId": 1,
            "Name": "默认"
          }
        ],
        "Status": "WARN",
        "ActivePoolType": "MAIN",
        "ActiveTrafficStrategy": "WEIGHT",
        "MonitorNum": 1,
        "KeepDomainRecords": "DISABLED",
        "CreatedOn": "2020-09-22T00:00:00+00:00",
        "UpdatedOn": "2020-09-22T00:00:00+00:00"
      }
    ],
    "TotalCount": 1,
    "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound	资源不存在。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.QueryNotAllow	不支持的查询
UnsupportedOperation.ResourceNameDuplicated	资源名不能重复。
UnsupportedOperation.ResourceSourceDuplicated	存在重复的解析来源。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

查询分组线路列表

最近更新时间：2025-07-21 01:27:52

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

查询分组线路列表接口

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：DescribeDnsLineList。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
InstanceId	是	String	实例id 示例值：gtm-Ul65AOsdfDvxVD

3. 输出参数

参数名称	类型	描述
DnsLineSet	Array of GroupLine	地址池列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 线路列表

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeDnsLines
<公共请求参数>

{
  "InstanceId": "gtm-dsdd123xdo"
}
```

输出示例

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

```
"DnsLineSet": [
  {
    "DnsLineId": 1,
    "Parent": 1,
    "LineName": "默认",
    "LineId": "1",
    "Useful": true,
    "SubGroup": 1,
    "LinePackage": 1,
    "Weight": 1
  }
],
"RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0"
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。

错误码	描述
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

修改策略

最近更新时间：2025-07-22 01:51:52

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

修改策略接口

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：ModifyStrategy。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
InstanceId	是	String	实例id 示例值：gtm-Ul65AOsdfDvxVD
StrategyId	是	Integer	策略id 示例值：3
Source.N	是	Array of Source	解析线路，需要全量传参 示例值：境外:欧洲,美国,东南亚
MainAddressPoolSet.N	是	Array of MainAddressPool	主力地址池集合，需要全量传参 示例值：[{"AddressPools": [{"PoolId":1,"Weight":50}], "MinAliveNodeNum":1,"TrafficStrtegy":"weight"}, {"AddressPools": [{"PoolId":3,"Weight":30}], "MinAliveNodeNum":1,"TrafficStrtegy":"weight"}, {"AddressPools":[{"PoolId":3}], "MinAliveNodeNum":1,"TrafficStrtegy":"all"}]
FallbackAddressPoolSet.N	是	Array of MainAddressPool	兜底地址池集合，需要全量传参 示例值：[{"AddressPools":[{"PoolId":1}]}]
StrategyName	否	String	策略名称，不允许重复 示例值：name
IsEnabled	否	String	策略开启状态：ENABLED开启；DISABLED关闭 示例值：ENABLED
KeepDomainRecords	否	String	是否开启策略强制保留默认线路 disabled, enabled，默认不开启且只有一个策略能开启 示例值：enabled
SwitchPoolType	否	String	调度模式：AUTO默认；STOP仅暂停不切换 示例值：AUTO

3. 输出参数

参数名称	类型	描述
Msg	String	是否成功 示例值：success
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需提供该次请求的 RequestId。

4. 示例

示例1 修改策略

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: ModifyStrategy
<公共请求参数>

{
  "InstanceId": "gtm-sdsdrdasd",
  "StrategyId": 1,
  "StrategyName": "test_strategy",
  "Source": [
    {
      "DnsLineId": 1,
      "Name": "默认"
    }
  ],
  "MainAddressPoolSet": [
    {
      "MainAddressPoolId": 1,
      "AddressPools": [
        {
          "PoolId": 1,
          "Weight": 1
        }
      ],
      "MinSurviveNum": 1,
      "TrafficStrategy": "all"
    }
  ],
  "FallbackAddressPoolSet": [
    {
      "MainAddressPoolId": 1,
      "AddressPools": [
        {
          "PoolId": 1,
          "Weight": 1
        }
      ],
      "MinSurviveNum": 1,
      "TrafficStrategy": "all"
    }
  ],
  "IsEnabled": "ENABLED",
  "KeepDomainRecords": "DISABLED"
}
```

输出示例

```
{
  "Response": {
    "Msg": "success",
    "RequestId": "abdsadasddc12312jhjyg"
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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.CanNotDeleteDefaultLineError	必须存在一条默认解析线路。
FailedOperation.ExceededLimit	超出限额。
FailedOperation.ModifyError	修改错误。
FailedOperation.ModifyStrategyFailed	修改策略失败
FailedOperation.PoolWasUsingError	地址池正在使用。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.InstanceDisabled	实例不可用，请检查实例状态
InvalidParameter.TrafficStrategyInvalid	流量策略不合法
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TrafficStrategyWeightEmpty	流量策略为负载均衡且主力集合内有多个地址池时，权重不能为空
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
ResourceInsufficient.NoInstance	没有实例,请先购买实例
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.InstanceUnauthorized	未授权,不可操作该实例
UnauthorizedOperation.MainBindPoolUnauthorized	未授权,主力集合中包含未授权地址池
UnauthorizedOperation.MainPoolUnauthorized	未授权,不可操作该主力地址池
UnauthorizedOperation.StrategyUnauthorized	未授权,不可操作该策略

错误码	描述
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation	操作不支持。
UnsupportedOperation.FallBackPoolInvalid	兜底地址池格式不合法
UnsupportedOperation.MainBindPoolDuplicate	主力地址池重复
UnsupportedOperation.MainBindPoolEmpty	主力地址池为空
UnsupportedOperation.MainBindPoolLevelExceed	主力地址集合数量超过限制
UnsupportedOperation.MainBindPoolMixed	主力地址集合内地址类型不能混用
UnsupportedOperation.MainBindPoolNumInvalid	主力地址集合数量不合法
UnsupportedOperation.MainPoolDuplicate	主力地址集合重复
UnsupportedOperation.MainPoolEmpty	主力集合不能为空
UnsupportedOperation.MainPoolSurviveNumInvalid	主力地址池阈值不合法
UnsupportedOperation.ResourceNameDuplicated	资源名不能重复。
UnsupportedOperation.ResourcePoolSourceNotExist	主备资源组中资源不存在。
UnsupportedOperation.ResourceSourceDuplicated	存在重复的解析来源。
UnsupportedOperation.StrategyNameDuplicate	策略名称不能重复
UnsupportedOperation.StrategySourceDuplicate	策略线路重复
UnsupportedOperation.StrategySourceInUsed	策略线路已使用
UnsupportedOperation.StrategySourceUnauthorized	策略线路不在实例套餐中
UnsupportedOperation.StrategySwitchTypeInvalid	策略切换类型不合法
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

(新)套餐管理相关接口

获取实例套餐列表

最近更新时间：2025-07-11 01:29:30

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

实例套餐列表

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：DescribeInstancePackageList。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Limit	否	Integer	每页条数 示例值：10
Filters.N	否	Array of ResourceFilter	InstanceId实例Id，InstanceName实例名称，ResourceId套餐Id，PackageType套餐类型 示例值：[{"Name": "InstanceName", "Value": ["未命名"]}]
IsUsed	否	Integer	是否使用：0未使用1已使用 示例值：0

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	总数 示例值：1
InstanceSet	Array of InstancePackage	实例套餐列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 实例套餐列表

实例套餐列表

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeInstancePackageList
```

<公共请求参数>

{}

输出示例

```
{
  "Response": {
    "InstanceSet": [
      {
        "AutoRenewFlag": 1,
        "CostItemList": [],
        "CreateTime": "2024-03-27 14:08:03",
        "CurrentDeadline": "2024-04-27 14:08:03",
        "InstanceId": "",
        "InstanceName": "",
        "IsExpire": 0,
        "MinCheckInterval": 60,
        "MinGlobalTtl": 60,
        "PackageType": "STANDARD",
        "Remark": "",
        "ResourceId": "gtm-pdnssdbxdst",
        "ScheduleStrategy": [
          "LOCATION",
          "DELAY"
        ],
        "Status": "ENABLED",
        "TrafficStrategy": [
          "ALL",
          "WEIGHT"
        ]
      },
      {
        "AutoRenewFlag": 0,
        "CostItemList": [
          {
            "CostName": "sv_igtm_standard_instance",
            "CostValue": 1
          }
        ],
        "CreateTime": "2024-03-27 09:55:06",
        "CurrentDeadline": "2025-04-27 09:55:07",
        "InstanceId": "gtm-axjpoifqwev",
        "InstanceName": "gjz-测试",
        "IsExpire": 0,
        "MinCheckInterval": 60,
        "MinGlobalTtl": 60,
        "PackageType": "STANDARD",
        "Remark": "",
        "ResourceId": "gtm-rznpntnbzov",
        "ScheduleStrategy": [
          "LOCATION",
          "DELAY"
        ],
        "Status": "ENABLED",
        "TrafficStrategy": [
```

```
"ALL",
"WEIGHT"
]
}
],
"RequestId": "d90cae16-82b5-481c-bc1b-3e11ee829d50",
"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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

探测任务套餐详情

最近更新时间：2025-07-21 01:27:53

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com 。

探测任务包详情

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：DescribeDetectPackageDetail。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
ResourceId	是	String	资源id 示例值：task-dsdd123xdo

3. 输出参数

参数名称	类型	描述
ResourceId	String	资源id 示例值：task-dsdd123xdo
ResourceType	String	资源类型 TASK 探测任务 示例值：TASK
Quota	Integer	额度 示例值：1
CurrentDeadline	String	过期时间 示例值：2023-12-20 12:00:00
CreateTime	String	创建时间 示例值：2023-11-20 12:00:00
IsExpire	Integer	是否过期 示例值：0
Status	String	状态 ENABLED: 正常 ISOLATED: 隔离 DESTROYED: 销毁 REFUNDED: 已退款 示例值：ENABLED
AutoRenewFlag	Integer	是否自动续费：0否1是 示例值：1
Remark	String	备注 示例值：这是备注
CostItemList	Array of CostItem	计费项
UsedNum	Integer	使用数量 示例值：0

参数名称	类型	描述
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 探测任务包详情

探测任务包详情

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeDetectPackageDetail
<公共请求参数>

{
  "ResourceId": "task-dsdd123xdo"
}
```

输出示例

```
{
  "Response": {
    "AutoRenewFlag": 1,
    "CostItemList": [
      {
        "CostName": "sv_igtm_monitor_task_mt",
        "CostValue": 1
      }
    ],
    "CreateTime": "2024-03-13 15:41:10",
    "CurrentDeadline": "2024-04-13 15:41:11",
    "IsExpire": 0,
    "Quota": 50,
    "Remark": "",
    "RequestId": "8f0325a8-4dd6-4fcb-8f6b-c45e587e51b0",
    "ResourceId": "task-hzxwkl1fznla",
    "ResourceType": "TASK_NUM",
    "Status": "ENABLED",
    "UsedNum": 17
  }
}
```

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: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

获取探测任务套餐列表

最近更新时间：2025-07-11 01:29:30

1. 接口描述

接口请求域名：igtm.tencentcloudapi.com。

探测任务套餐列表

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：DescribeDetectTaskPackageList。
Version	是	String	公共参数 ，本接口取值：2023-10-24。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Limit	否	Integer	每页条数 示例值：10
Filters.N	否	Array of ResourceFilter	探测任务过滤条件：ResourceId 探测任务的资源id，PeriodStart 最小过期时间,PeriodEnd 最大过期时间 示例值：[{"Name": "ResourceId", "Value": ["资源id"]}]

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	总数 示例值：1
TaskPackageSet	Array of DetectTaskPackage	探测任务套餐列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 探测任务套餐列表

探测任务套餐列表

输入示例

```
POST / HTTP/1.1
Host: igtm.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeDetectTaskPackageList
<公共请求参数>

{}
```

输出示例

```
{
  "Response": {
    "RequestId": "b83ab7db-3a32-42ec-875a-aaff131836f1",
    "TaskPackageSet": [
      {
        "AutoRenewFlag": 1,
        "CostItemList": [
          {
            "CostName": "sv_igtm_monitor_task_mt",
            "CostValue": 1
          }
        ],
        "CreateTime": "2024-03-13 15:41:10",
        "CurrentDeadline": "2024-04-13 15:41:11",
        "IsExpire": 0,
        "Quota": 50,
        "Remark": "",
        "ResourceId": "task-hzxwlklfznla",
        "ResourceType": "TASK_NUM",
        "Status": "ENABLED"
      },
      {
        "AutoRenewFlag": 0,
        "CostItemList": [
          {
            "CostName": "sv_igtm_monitor_task_mt",
            "CostValue": 1
          }
        ],
        "CreateTime": "2024-03-13 14:48:20",
        "CurrentDeadline": "2024-03-12 14:48:21",
        "IsExpire": 1,
        "Quota": 1,
        "Remark": "",
        "ResourceId": "task-uukzocrmqior",
        "ResourceType": "TASK_NUM",
        "Status": "ENABLED"
      }
    ],
    "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: [GitHub](#), [Gitee](#)

- Tencent Cloud SDK 3.0 for Java: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for PHP: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Go: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Node.js: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for .NET: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for C++: [GitHub](#), [Gitee](#)
- Tencent Cloud SDK 3.0 for Ruby: [GitHub](#), [Gitee](#)

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。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。

数据结构

最近更新时间：2025-07-21 01:27:54

Address

添加地址池地址

被如下接口引用：CreateAddressPool, DescribeAddressPoolDetail, DescribeAddressPoolList, ModifyAddressPool。

名称	类型	必选	描述
Addr	String	是	地址值：只支持ipv4、ipv6和域名格式； 不支持回环地址、保留地址、内网地址与腾讯保留网段 示例值：20.1.24.242
IsEnable	String	是	是否启用:DISABLED不启用；ENABLED启用 示例值：ENABLED
AddressId	Integer	否	地址id 示例值：1
Location	String	否	地址名称 示例值：德国慕尼黑
Status	String	否	OK正常，DOWN故障，WARN风险，UNKNOWN探测中，UNMONITORED未知 示例值：探测状态
Weight	Integer	否	权重，流量策略为WEIGHT时，必填；范围1-100 注意：此字段可能返回 null，表示取不到有效值。 示例值：5
CreatedOn	Timestamp ISO8601	否	创建时间 示例值：2021-08-17T17:32:23+08:00
UpdatedOn	Timestamp ISO8601	否	修改时间 示例值：2021-08-17T17:32:23+08:00

AddressLocation

地址所属地域

被如下接口引用：DescribeAddressLocation。

名称	类型	描述
Addr	String	ip地址 示例值：1.1.1.1.1
Location	String	所属地域 示例值：局域网-未知-未知

AddressPool

地址池

被如下接口引用：DescribeAddressPoolList。

名称	类型	必选	描述
PoolId	Integer	否	地址池 id 示例值：5
PoolName	String	否	地址池名 示例值：测试地址池2
AddrType	String	否	地址池地址类型：IPV4、IPV6、DOMAIN 示例值：IPV4

名称	类型	必选	描述
TrafficStrategy	String	否	流量策略: WEIGHT负载均衡, ALL解析全部 示例值: ALL
MonitorId	Integer	否	监控器id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
Status	String	否	OK正常, DOWN故障, WARN风险, UNKNOWN未知 示例值: OK
AddressNum	Integer	否	地址数 示例值: 1
MonitorGroupNum	Integer	否	探点数 示例值: 1
MonitorTaskNum	Integer	否	探测任务数 示例值: 1
InstanceInfo	Array of InstanceInfo	否	实例相关信息 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: [{"InstanceId":"gtm-dsdd123xdo","InstanceName":"测试实例"}]
AddressSet	Array of Address	否	地址池地址信息 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: [{"AddressId":32,"Addr":"2.2.2.2","Location":"","Status":"UNKNOWN","IsEnable":"ENABLED","V07-09 21:03:16","UpdatedOn":"2025-07-09 21:03:16"}]
CreatedOn	Timestamp ISO8601	否	创建时间 示例值: 2021-08-17T17:32:23+08:00
UpdatedOn	Timestamp ISO8601	否	更新时间 示例值: 2021-08-17T17:32:23+08:00

AddressPoolDetail

地址池详情

被如下接口引用: DescribeAddressPoolDetail。

名称	类型	必选	描述
PoolId	Integer	否	地址池 id 示例值: 5
PoolName	String	否	地址池名 示例值: 测试地址池2
AddrType	String	否	地址池地址类型: IPV4、IPV6、DOMAIN 示例值: IPV4
TrafficStrategy	String	否	流量策略: WEIGHT负载均衡, ALL解析全部 示例值: ALL
MonitorId	Integer	否	监控器id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
CreatedOn	Timestamp ISO8601	否	创建时间 示例值: 2021-08-17T17:32:23+08:00
UpdatedOn	Timestamp ISO8601	否	更新时间 示例值: 2021-08-17T17:32:23+08:00

CostItem

计费项

被如下接口引用：DescribeDetectPackageDetail, DescribeDetectTaskPackageList, DescribeInstancePackageList。

名称	类型	描述
CostName	String	计费项名称 示例值：SV-xxx
CostValue	Integer	计费项值 示例值：1

DetectTaskPackage

探测任务套餐

被如下接口引用：DescribeDetectTaskPackageList。

名称	类型	描述
ResourceId	String	资源id 示例值：task-aabbb
ResourceType	String	资源类型 TASK 探测任务 示例值：TASK
Quota	Integer	额度 示例值：1
CurrentDeadline	String	套餐过期时间 示例值：2023-12-20 12:00:00
CreateTime	String	套餐创建时间 示例值：2023-12-20 12:00:00
IsExpire	Integer	是否过期0否1是 示例值：0
Status	String	状态 ENABLED: 正常 ISOLATED: 隔离 DESTROYED: 销毁 REFUNDED: 已退款 示例值：ENABLED
AutoRenewFlag	Integer	是否自动续费0不1是 示例值：1
Remark	String	备注 示例值：这是备注
CostItemList	Array of CostItem	计费项
Group	Integer	探测任务类型：100系统设定；200计费；300管理系统；110D监控迁移的免费任务；120容灾切换任务 注意：此字段可能返回 null，表示取不到有效值。 示例值：200

DetectorGroup

探测组

被如下接口引用：DescribeDetectors。

名称	类型	必选	描述
Gid	Integer	是	线路组id GroupLineId 示例值：8

名称	类型	必选	描述
GroupType	String	否	bgp, international, isp 示例值：bgp
GroupName	String	否	组名 示例值：深圳市
InternetFamily	String	否	ipv4, ipv6 示例值：ipv4
PackageSet	Array of String	否	支持的套餐类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：["FREE", "STANDARD", "ULTIMATE"]

GroupLine

线路列表

被如下接口引用：DescribeDnsLineList。

名称	类型	必选	描述
DnsLineId	Integer	是	分组线路id 示例值：1
Parent	Integer	是	父节点 0为根节点 示例值：0
LineName	String	是	线路名 示例值：境外地区
LineId	String	是	10=9 DNSPod 线路 id 示例值：线路id
Useful	Boolean	是	是否已使用过 示例值：0
SubGroup	Integer	是	0为未使用 示例值：子分组
LinePackage	Integer	否	权限标识 示例值：1
Weight	Integer	否	1 示例值：权重

Instance

返回实例

被如下接口引用：DescribeInstanceList。

名称	类型	必选	描述
InstanceId	String	否	实例id 示例值：gtm-xxx
InstanceName	String	否	实例名 示例值：instanceName
ResourceId	String	否	资源 id 示例值：gtmins-xxxxxx
Domain	String	否	业务域名 示例值：igtm-a101.com
AccessType	String	否	Cname域名接入方式 CUSTOM: 自定义接入域名

名称	类型	必选	描述
			SYSTEM: 系统接入域名 示例值: CUSTOM
AccessDomain	String	否	接入域名 示例值: gtmtest.com
AccessSubDomain	String	否	接入子域名 示例值: gtm-access
GlobalTtl	Integer	否	全局记录过期时间 示例值: 600
PackageType	String	否	套餐类型 FREE: 免费版 STANDARD: 标准版 ULTIMATE: 旗舰版 示例值: FREE
WorkingStatus	String	否	实例运行状态 NORMAL: 健康 FAULTY: 有风险 DOWN: 宕机 UNKNOWN: 未知 示例值: NORMAL
Status	String	否	实例状态, ENABLED: 正常, DISABLED: 禁用 示例值: ENABLED
IsCnameConfigured	Boolean	否	是否cname接入: true已接入; false未接入 示例值: true
Remark	String	否	备注 示例值: 备注
StrategyNum	Integer	否	策略数量 示例值: 7
AddressPoolNum	Integer	否	绑定地址池个数 示例值: 2
MonitorNum	Integer	否	绑定监控器数量 示例值: 2
PoolId	Integer	否	地址池id 示例值: 1
PoolName	String	否	地址池名称 示例值: 地址池1
CreatedOn	String	否	实例创建时间 示例值: 2023-12-20 12:00:00 03:11:46
UpdatedOn	String	否	实例更新时间 示例值: 2023-12-20 12:00:00 03:11:46

InstanceConfig

实例配置详情

被如下接口引用: ModifyInstanceConfig。

名称	类型	必选	描述
InstanceName	String	是	实例名称 示例值: 实例名称
Domain	String	是	业务域名 示例值: maingtm.com

名称	类型	必选	描述
AccessType	String	是	CUSTOM: 自定义接入域名, SYSTEM: 系统接入域名 示例值: CUSTOM
Remark	String	是	备注 示例值: 备注信息
GlobalTtl	Integer	是	全局记录过期时间 示例值: 600
AccessDomain	String	否	接入主域名 示例值: gtmtest.com
AccessSubDomain	String	否	接入子域名 示例值: gtm-access

InstanceDetail

返回实例

被如下接口引用: DescribeInstanceDetail, ModifyInstanceConfig。

名称	类型	必选	描述
InstanceId	String	否	实例id 示例值: gtminstance1
InstanceName	String	否	实例名 示例值: instanceName
Domain	String	否	业务域名 示例值: maintest.com
AccessType	String	否	Cname域名接入方式 CUSTOM: 自定义接入域名 SYSTEM: 系统接入域名 示例值: CUSTOM
AccessSubDomain	String	否	接入子域名 示例值: gtm-access
AccessDomain	String	否	接入域名 示例值: gtmtest.com
GlobalTtl	Integer	否	解析生效时间 示例值: 600
PackageType	String	否	套餐类型 FREE: 免费版 STANDARD: 标准版 ULTIMATE: 旗舰版 示例值: FREE
WorkingStatus	String	否	实例运行状态 NORMAL: 健康 FAULTY: 有风险 DOWN: 宕机 UNKNOWN: 未知 示例值: NORMAL
Status	String	否	实例状态, ENABLED: 正常; DISABLED: 禁用 示例值: ENABLED
IsCnameConfigured	Boolean	否	cname是否接入: true已接入; false未接入 示例值: true
Remark	String	否	备注 示例值: 这是备注

名称	类型	必选	描述
StrategyNum	Integer	否	策略数量 示例值：2
AddressPoolNum	Integer	否	绑定地址池个数 示例值：2
MonitorNum	Integer	否	绑定监控器数量 示例值：2
ResourceId	String	否	实例绑定套餐资源id 示例值：ins-oxrvmn14yrd
NotifyEventSet	Array of String	否	订阅事件列表 示例值：[{"owner_uin":"10012894645"}]
CreatedOn	String	否	实例创建时间 示例值：2023-12-20 12:00:00 03:11:46
UpdatedOn	String	否	实例更新时间 示例值：2023-12-20 12:00:00 03:11:46

InstanceInfo

实例相关信息

被如下接口引用：DescribeAddressPoolList。

名称	类型	必选	描述
InstanceId	String	否	实例id 示例值：gtm-xxx
InstanceName	String	否	实例名称 示例值：实例1

InstancePackage

实例套餐

被如下接口引用：DescribeInstancePackageList。

名称	类型	描述
ResourceId	String	实例套餐资源id 注意：此字段可能返回 null，表示取不到有效值。 示例值：ins-aaaa
InstanceId	String	实例id 示例值：gtminstance1
InstanceName	String	实例名 示例值：instanceName
PackageType	String	套餐类型 FREE：免费版 STANDARD：标准版 ULTIMATE：旗舰版 示例值：FREE
CurrentDeadline	String	套餐过期时间 示例值：2023-12-20 12:00:00
CreateTime	String	套餐创建时间 示例值：2023-12-20 12:00:00
IsExpire	Integer	是否过期0否1是 示例值：0

名称	类型	描述
Status	String	实例状态 ENABLED: 正常 DISABLED: 禁用 示例值: ENABLED
AutoRenewFlag	Integer	是否自动续费0不1是 示例值: 1
Remark	String	备注 示例值: 这是备注
CostItemList	Array of CostItem	计费项
MinCheckInterval	Integer	最小检查间隔时间s 示例值: 60
MinGlobalTtl	Integer	最小TTL s 示例值: 600
TrafficStrategy	Array of String	流量策略类型: ALL返回全部, WEIGHT权重 示例值: ALL
ScheduleStrategy	Array of String	策略类型: LOCATION按地理位置调度, DELAY按延迟调度 示例值: LOCATION

MainAddressPool

主力地址池

被如下接口引用: CreateStrategy, DescribeStrategyDetail, ModifyStrategy。

名称	类型	必选	描述
AddressPools	Array of MainPoolWeight	是	集合中的地址池id与权重, 数组 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: [{"PoolId":314,"Weight":1}]
MainAddressPoolId	Integer	否	地址池集合id 示例值: 1
MinSurviveNum	Integer	否	切换阈值, 不能大于主力集合内地址总数 示例值: 1
TrafficStrategy	String	否	切换策略:ALL解析所有地址; WEIGHT: 负载均衡。当为ALL时, 解析地址的权重值为1; 当为WEIGHT时; 权重为地址池权重*地址权重 示例值: ALL

MainPoolWeight

主力地址池id与权重

被如下接口引用: CreateStrategy, ModifyStrategy。

名称	类型	必选	描述
PoolId	Integer	是	地址池id 示例值: 1
Weight	Integer	否	权重 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 50

MonitorDetail

监控器详情

被如下接口引用: DescribeMonitorDetail, DescribeMonitors。

名称	类型	必选	描述
MonitorId	Integer	否	探测规则id 示例值：1
MonitorName	String	否	监控器名称 示例值：监控器1
Uin	String	否	所属用户 示例值：10012894645
DetectorGroupIds	Array of Integer	否	监控节点id组 示例值：[2,5,8]
CheckProtocol	String	否	探测协议 PING TCP HTTP HTTPS 示例值：PING
CheckInterval	Integer	否	探测周期 示例值：60
PingNum	Integer	否	发包数量 示例值：50
TcpPort	Integer	否	tcp端口 示例值：80
Host	String	否	探测 host 示例值：www.baidu.com
Path	String	否	探测路径 示例值：/path
ReturnCodeThreshold	Integer	否	返回值阈值 示例值：500
EnableRedirect	String	否	是否开启3xx重定向跟随 ENABLED DISABLED 示例值：DISABLED
EnableSni	String	否	是否启用 sni ENABLED DISABLED 示例值：DISABLED
PacketLossRate	Integer	否	丢包率上限 示例值：30
Timeout	Integer	否	探测超时 示例值：200
FailTimes	Integer	否	失败次数 示例值：50
FailRate	Integer	否	失败率上限100 示例值：50
CreatedOn	String	否	创建时间 示例值：2021-08-17T17:32:23+08:00
UpdatedOn	String	否	更新时间 示例值：2021-08-17T17:32:23+08:00
DetectorStyle	String	否	监控节点类型 AUTO INTERNAL OVERSEAS IPV6 ALL 示例值：AUTO
DetectNum	Integer	否	探测次数 示例值：0
ContinuePeriod	Integer	否	持续周期数 示例值：1

Quota

配额

被如下接口引用：DescribeQuotas。

名称	类型	必选	描述
TaskQuota	Integer	否	探测任务配额 示例值：2
PoolQuota	Integer	否	地址池配额 示例值：20
AddressQuota	Integer	否	地址配额 示例值：25
MonitorQuota	Integer	否	探点资源数 示例值：1
MessageQuota	Integer	否	消息资源数 示例值：1
UsedTaskQuota	Integer	否	已使用探测任务数 示例值：1
UsedFreeInstanceNum	Integer	否	已使用体验实例数 示例值：1
UsedBillInstanceNum	Integer	否	已使用付费实例 示例值：1
FreePackageNum	Integer	否	体验套餐总数 示例值：1
UsedBillPackageNum	Integer	否	已使用付费套餐数 示例值：1
BillPackageNum	Integer	否	付费套餐总数 示例值：1

ResourceFilter

查询时过滤条件。

被如下接口引用：DescribeAddressPoolList, DescribeDetectTaskPackageList, DescribeInstanceList, DescribeInstancePackageList, DescribeStrategyList。

名称	类型	必选	描述
Name	String	是	过滤字段名，支持的列表如下： - type：主资源类型，CDN。 - instanceId：IGTM实例ID。此为必传参数，未传将导致接口查询失败。 示例值：InstanceId
Value	Array of String	是	过滤字段值。 示例值：gtm-xxxxxxxxx
Fuzzy	Boolean	否	是否启用模糊查询，仅支持过滤字段名为domain。 模糊查询时，Value长度最大为1，否则Value长度最大为5。（预留字段，暂未使用） 示例值：Ture

Source

解析线路

被如下接口引用：CreateStrategy, DescribeStrategyDetail, DescribeStrategyList, ModifyStrategy。

名称	类型	必选	描述
DnsLineId	Integer	是	解析请求来源线路id 示例值：1
Name	String	否	解析请求来源线路名 示例值：境外

Strategy

地址池

被如下接口引用：DescribeStrategyList。

名称	类型	必选	描述
InstanceId	String	是	实例id 示例值：gtminstance1
Name	String	是	策略名 示例值：第一条策略
Source	Array of Source	是	地址来源 注意：此字段可能返回 null，表示取不到有效值。 示例值：测试地址池2
StrategyId	Integer	否	策略id 示例值：1
Status	String	否	健康状态：ok健康、warn风险、down故障 示例值：warn
ActivateMainPoolId	Integer	否	生效的主力池id，null则为未知 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
ActivateLevel	Integer	否	当前生效地址池所在级数，为0则代表兜底生效，null则为未知 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
ActivePoolType	String	否	当前生效地址池集合类型：main主力；fallback兜底 示例值：main
ActiveTrafficStrategy	String	否	当前生效地址池流量策略：all解析所有；weight负载均衡 示例值：all
MonitorNum	Integer	否	监控器数量 示例值：1
IsEnabled	String	否	是否开启：ENABLED开启；DISABLED关闭 示例值：ENABLED
KeepDomainRecords	String	否	是否保留线路：enabled保留，disabled不保留，只保留默认线路 示例值：enabled
SwitchPoolType	String	否	调度模式：AUTO默认；PAUSE仅暂停不切换 示例值：AUTO
CreatedOn	Timestamp ISO8601	否	创建时间 示例值：2021-08-17T17:32:23+08:00
UpdatedOn	Timestamp ISO8601	否	更新时间 示例值：2021-08-17T17:32:23+08:00

StrategyDetail

策略详情

被如下接口引用：DescribeStrategyDetail。

名称	类型	必选	描述
InstanceId	String	是	实例id 示例值: gtminstance1
StrategyId	Integer	是	策略id 示例值: 1
Name	String	是	策略名 示例值: 第一条策略
Source	Array of Source	是	线路 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: [{"Name": "默认", "GroupLineId": 1}]
MainAddressPoolSet	Array of MainAddressPool	是	主力地址池集合 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: [{"MainAddressPoolId": 5, "AddressPools": [{"PoolId": 314, "Weight": null}], "MinSurviveNum": 1, "TrafficStrategy": "WEIGHT"}]
FallbackAddressPoolSet	Array of MainAddressPool	是	兜底地址池id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: [{"AddressPools": [{"PoolId": 1}]}]
KeepDomainRecords	String	否	是否保留线路: enabled保留, disabled不保留, 只保留默认线路 示例值: enabled
ActivateMainPoolId	Integer	否	生效主力地址池id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
CreatedOn	Timestamp ISO8601	否	创建时间 示例值: 2021-08-17T17:32:23+08:00
UpdatedOn	Timestamp ISO8601	否	更新时间 示例值: 2021-08-17T17:32:23+08:00
SwitchPoolType	String	否	调度模式: AUTO默认; PAUSE仅暂停不切换 示例值: AUTO

错误码

最近更新时间：2025-06-30 11:19:46

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 和服务器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。
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。

错误码	说明
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.AddrExceededLimit	地址池中地址数量超出限额
FailedOperation.CanNotDeleteDefaultLineError	必须存在一条默认解析线路。
FailedOperation.CreateError	创建失败。
FailedOperation.CreateInstanceFailed	创建实例失败
FailedOperation.CreateMonitorFailed	创建监控器失败
FailedOperation.CreatePoolFailed	创建地址池失败
FailedOperation.CreateStrategyFailed	创建策略失败
FailedOperation.DeleteError	操作删除失败。
FailedOperation.DeleteMonitorFailed	删除监控器失败
FailedOperation.DeletePoolFailed	删除地址池失败
FailedOperation.DeleteStrategyFailed	删除策略失败
FailedOperation.ExceededLimit	超出限额。
FailedOperation.ExceededMonitorNumLimit	探点数配额不足。
FailedOperation.ModifyError	修改错误。
FailedOperation.ModifyInstanceFailed	修改实例失败
FailedOperation.ModifyMonitorFailed	修改监控器失败

错误码	说明
FailedOperation.ModifyPoolFailed	修改地址池失败
FailedOperation.ModifyStrategyFailed	修改策略失败
FailedOperation.PoolExceededLimit	地址池数量超出限额
FailedOperation.PoolWasUsingError	地址池正在使用。
FailedOperation.QueryErr	查询失败
FailedOperation.TaskExceededLimit	探测任务数量超出限额
InternalError.DbErr	数据库读写错误。
InternalError.ParameterError	参数格式错误
InternalError.UndefinedError	错误未定义
InvalidParameter.AccessTypeInvalid	实例接入类型不合法
InvalidParameter.GlobalAccessDomainConflict	全局接入域名冲突，当前域名被其他实例使用，请解决占用后重试
InvalidParameter.InstanceDisabled	实例不可用，请检查实例状态
InvalidParameter.TrafficStrategyInvalid	流量策略不合法
InvalidParameterValue.PoolAddrDuplicated	地址池地址重复
InvalidParameterValue.PoolAddrEmpty	地址池地址不能为空
InvalidParameterValue.PoolAddrInvalid	地址池地址非法
InvalidParameterValue.PoolAddrMixed	地址池地址不能混用
InvalidParameterValue.PoolAddrWeightEmpty	地址池地址权重不能为空
InvalidParameterValue.TrafficStrategyInvalid	流量策略不合法
InvalidParameterValue.TrafficStrategyWeightEmpty	流量策略为负载均衡且主力集合内有多个地址池时，权重不能为空
RequestLimitExceeded.InnerRequestLimitExceeded	当前请求过于频繁，请稍后重试
ResourceInsufficient.NoInstance	没有实例，请先购买实例
ResourceNotFound.SearchInfo	查询错误，未查询到相关资源。
UnauthorizedOperation.InstanceUnauthorized	未授权,不可操作该实例
UnauthorizedOperation.MainBindPoolUnauthorized	未授权,主力集合中包含未授权地址池
UnauthorizedOperation.MainPoolUnauthorized	未授权,不可操作该主力地址池
UnauthorizedOperation.PoolAddressUnauthorized	未授权,不可操作该地址池地址
UnauthorizedOperation.PoolUnauthorized	未授权,不可操作该地址池
UnauthorizedOperation.StrategyUnauthorized	未授权,不可操作该策略
UnauthorizedOperation.UnauthorizedError	资源未授权。
UnsupportedOperation.AccessDomainUnauthorized	接入域名未使用腾讯云 DNSPod 解析，请重新输入
UnsupportedOperation.DomainTtlInvalid	接入域名的套餐不允许此TTL值
UnsupportedOperation.FallBackPoolInvalid	兜底地址池格式不合法
UnsupportedOperation.InstanceNameDuplicate	实例名称不能重复
UnsupportedOperation.MainBindPoolDuplicate	主力地址池重复
UnsupportedOperation.MainBindPoolEmpty	主力地址池为空
UnsupportedOperation.MainBindPoolLevelExceed	主力地址集合数量超过限制

错误码	说明
UnsupportedOperation.MainBindPoolMixed	主力地址集合内地址类型不能混用
UnsupportedOperation.MainBindPoolNumInvalid	主力地址集合数量不合法
UnsupportedOperation.MainPoolDuplicate	主力地址集合重复
UnsupportedOperation.MainPoolEmpty	主力集合不能为空
UnsupportedOperation.MainPoolSurviveNumInvalid	主力地址池阈值不合法
UnsupportedOperation.MonitorNameExist	监控器名已存在
UnsupportedOperation.PackageUnauthorized	套餐无效，请选择正确套餐
UnsupportedOperation.PoolNameDuplicate	地址池名称不能重复
UnsupportedOperation.QueryNotAllowed	不支持的查询
UnsupportedOperation.ResourceNameDuplicated	资源名不能重复。
UnsupportedOperation.ResourcePoolSourceNotExist	主备资源组中资源不存在。
UnsupportedOperation.ResourceSourceDuplicated	存在重复的解析来源。
UnsupportedOperation.StrategyNameDuplicate	策略名称不能重复
UnsupportedOperation.StrategySourceDuplicate	策略线路重复
UnsupportedOperation.StrategySourceInUsed	策略线路已使用
UnsupportedOperation.StrategySourceUnauthorized	策略线路不在实例套餐中
UnsupportedOperation.StrategySwitchTypeInvalid	策略切换类型不合法
UnsupportedOperation.TTLInvalid	TTL无效,请设置正确的TTL
UnsupportedOperation.UserNotFound	用户未登录，缺少Uin参数。