

AI Agent 安全网关 API 文档



腾讯云

【 版权声明 】

©2013–2026 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

模型管理相关接口

修改模型

查询模型列表

查询模型详情

删除模型

创建模型

模型服务管理相关接口

编辑模型服务

查询模型服务列表

查询模型服务详情

删除模型服务

创建模型服务

应用管理相关接口

修改app的model service关联

修改app

查询app列表

查询app model服务关联列表

查询app mcp关联列表

查询app详情

删除app的model service关联

删除app的mcp server

删除app

创建app的model service关联

创建app的mcp server关联

创建app

MCP 管理相关接口

修改mcp server

查询mcp server列表

查询mcp server详情

删除mcp server

创建mcp server

凭据管理相关接口

修改凭据

查询凭据列表

查询凭据详情

删除凭据

创建凭据

数据结构

错误码

API 文档

更新历史

最近更新时间：2026-04-29 01:11:11

第 5 次发布

发布时间：2026-04-29 01:10:59

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateModelService](#)
 - 新增入参：TargetSelect, FindHostKeyMethod, HostKeyHeaderName, FallbackStatus, FallbackModels
- [ModifyModelService](#)
 - 新增入参：TargetSelect, FindHostKeyMethod, HostKeyHeaderName, FallbackStatus, FallbackModels

修改数据结构：

- [DescribeModelServiceResponseVO](#)
 - 新增成员：TargetSelect, FindHostKeyMethod, HostKeyHeaderName, FallbackStatus, FallbackModels

第 4 次发布

发布时间：2026-04-21 01:10:59

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateModelService](#)
 - 新增入参：SensitiveDataCheckStatus, SensitiveDataCheckConfig
- [ModifyModelService](#)
 - 新增入参：SensitiveDataCheckStatus, SensitiveDataCheckConfig

新增数据结构：

- [SensitiveDataCheckConfigDTO](#)

修改数据结构：

- [DescribeModelServiceResponseVO](#)
 - 新增成员：SensitiveDataCheckStatus, SensitiveDataCheckConfig

第 3 次发布

发布时间：2026-03-31 01:10:44

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [CreateModelService](#)
 - 新增入参：PromptModerateStatus, PromptModerateConfig
- [ModifyModelService](#)
 - 新增入参：PromptModerateStatus, PromptModerateConfig

新增数据结构：

- [PromptModerateConfigDTO](#)

修改数据结构：

- [DescribeModelServiceResponseVO](#)
 - 新增成员：PromptModerateStatus, PromptModerateConfig

第 2 次发布

发布时间：2026-03-24 01:11:04

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateAgentCredential](#)
- [DeleteAgentCredential](#)
- [DescribeAgentCredential](#)
- [DescribeAgentCredentials](#)
- [ModifyAgentCredential](#)

第 1 次发布

发布时间：2026-03-23 17:43:24

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateAgentApp](#)
- [CreateAgentAppMcpServers](#)
- [CreateAgentAppModelServices](#)

- [CreateMcpServer](#)
- [CreateModel](#)
- [CreateModelService](#)
- [DeleteAgentApp](#)
- [DeleteAgentAppMcpServers](#)
- [DeleteAgentAppModelServices](#)
- [DeleteMcpServer](#)
- [DeleteModel](#)
- [DeleteModelService](#)
- [DescribeAgentApp](#)
- [DescribeAgentAppMcpServers](#)
- [DescribeAgentAppModelServices](#)
- [DescribeAgentApps](#)
- [DescribeMcpServer](#)
- [DescribeMcpServers](#)
- [DescribeModel](#)
- [DescribeModelService](#)
- [DescribeModelServices](#)
- [DescribeModels](#)
- [ModifyAgentApp](#)
- [ModifyAgentAppModelServices](#)
- [ModifyMcpServer](#)
- [ModifyModel](#)
- [ModifyModelService](#)

新增数据结构:

- [AgentAppMcpServerDTO](#)
- [AgentAppMcpServerVO](#)
- [AgentAppModelServiceDTO](#)
- [AgentAppSecretKeyVO](#)
- [AgentCredentialContentDTO](#)
- [AgentCredentialContentHeaderDTO](#)
- [BindMcpSecurityRuleDTO](#)
- [BindMcpSecurityRuleVO](#)
- [CreateAgentAppResp](#)
- [DescribeAgentAppMcpServersResp](#)
- [DescribeAgentAppResp](#)
- [DescribeAgentAppsResp](#)
- [DescribeAgentAppsSortDTO](#)
- [DescribeAgentCredentialResp](#)
- [DescribeMcpServerResponseVO](#)
- [DescribeMcpServersResponseVO](#)
- [DescribeModelResponseVO](#)

- [DescribeModelServiceResponseVO](#)
- [DescribeModelServicesResponseVO](#)
- [DescribeModelServicesSort](#)
- [DescribeModelsResponseVO](#)
- [DescribeModelsSort](#)
- [IDNameVO](#)
- [InvokeLimitConfigDTO](#)
- [IpConfig](#)
- [LimitWindowsDTO](#)
- [McpInputOutSchema](#)
- [McpSecurityRule](#)
- [McpSecurityRulesVO](#)
- [McpTool](#)
- [McpToolAnnotation](#)
- [McpUrlObj](#)
- [PluginConfigDTO](#)
- [PluginFormValueDTO](#)
- [ResultIDVO](#)
- [ResultIDsVO](#)
- [StartEndTime](#)
- [TargetHostDTO](#)
- [TargetModelDTO](#)
- [TmsConfigDTO](#)
- [TokenLimitConfigDTO](#)
- [ToolConfigDTO](#)
- [ToolConfigVO](#)

简介

最近更新时间：2026-03-23 17:44:03

概述

AI Agent 安全网关是面向大模型场景的智能化安全中枢，提供模型上下文协议（Model Context Protocol, MCP）资产管理、MCP 安全防护，MCP Server 快速封装三大核心能力，助力企业实现 AI 基础设施的安全可控与高效部署。

本章节介绍的 AI Agent 安全网关 API 接口均为 API 3.0 接口。

您可以调用 API 对 AI Agent 安全网关进行操作，例如模型 API 接入、MCP 配置、策略管理、实例管理等。

AI Agent 安全网关支持的所有接口信息，请参见 API 概览。

术语表

AI Agent 安全网关 API 接口的常见术语请参见下表：

术语	描述
QPS	每秒查询率，衡量 AI Agent 安全网关实例处理请求能力的核心性能指标
MCP	Model Context Protocol，模型上下文协议
应用	应用持有密钥，关联 MCP、模型 API 后可使用应用对应的认证类型访问 MCP 或模型 API
模型	大模型的访问配置信息，如模型的访问地址、API key 鉴权等信息
模型 API	可将大模型的 API 接口发布为网关上的模型 API，实现负载、限流、token 限制、内容安全等能力
API 分组	多个 API 可关联到一个 API 分组下，用于管理 API 匹配的域名，证书，鉴权的 token 信息
API	适用于 HTTP 接口代理，不适合发布为模型 API 的接口可发布为 API

使用限制

对于 API 接口的参数限制，请参考各接口文档中的参数说明。

API 快速入门

您可以使用 API Explorer 工具在线调用 API。

本文以创建应用为例，通过 API Explorer 工具调用 API 接口的步骤如下：

1. 进入 [API Explorer](#) 工具页面。更多 API Explorer 工具使用信息，请参见 [使用 API Explorer](#)。
2. 调用 CreateAgentApp 接口，新建应用。
3. apiKey、secretKey 认证类型应用会返回密钥信息，请复制并妥善保存响应结果中的密钥信息。

API 概览

最近更新时间：2026-03-26 01:10:36

MCP 管理相关接口

接口名称	接口功能	频率限制（次/秒）
CreateMcpServer	创建mcp server	20
DeleteMcpServer	删除mcp server	20
DescribeMcpServer	查询mcp server详情	20
DescribeMcpServers	查询mcp server列表	20
ModifyMcpServer	修改mcp server	20

应用管理相关接口

接口名称	接口功能	频率限制（次/秒）
CreateAgentApp	创建app	20
CreateAgentAppMcpServers	创建app的mcp server关联	20
CreateAgentAppModelServices	创建app的model service关联	20
DeleteAgentApp	删除app	20
DeleteAgentAppMcpServers	删除app的mcp server	20
DeleteAgentAppModelServices	删除app的model service关联	20
DescribeAgentApp	查询app详情	20
DescribeAgentAppMcpServers	查询app mcp关联列表	20
DescribeAgentAppModelServices	查询app model服务关联列表	20
DescribeAgentApps	查询app列表	20
ModifyAgentApp	修改app	20
ModifyAgentAppModelServices	修改app的model service关联	20

模型服务管理相关接口

接口名称	接口功能	频率限制（次/秒）
CreateModelService	创建模型服务	20

接口名称	接口功能	频率限制（次/秒）
DeleteModelService	删除模型服务	20
DescribeModelService	查询模型服务详情	20
DescribeModelServices	查询模型服务列表	20
ModifyModelService	编辑模型服务	20

模型管理相关接口

接口名称	接口功能	频率限制（次/秒）
CreateModel	创建模型	20
DeleteModel	删除模型	20
DescribeModel	查询模型详情	20
DescribeModels	查询模型列表	20
ModifyModel	修改模型	20

凭据管理相关接口

接口名称	接口功能	频率限制（次/秒）
CreateAgentCredential	创建凭据	20
DeleteAgentCredential	删除凭据	20
DescribeAgentCredential	查询凭据详情	20
DescribeAgentCredentials	查询凭据列表	20
ModifyAgentCredential	修改凭据	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：2026-03-23 17:44:03

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

接入地域	域名
欧洲地区（法兰克福）	apis.eu-frankfurt.tencentcloudapi.com

注意：由于**金融区**和非金融区是隔离不互通的，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）	apis.ap-shanghai-fsi.tencentcloudapi.com
华南地区（深圳金融）	apis.ap-shenzhen-fsi.tencentcloudapi.com

2. 通信协议

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

3. 请求方法

支持的 HTTP 请求方法：

- POST（推荐）
- GET

POST 请求支持的 Content-Type 类型：

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

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

4. 字符编码

均使用 UTF-8 编码。

公共参数

最近更新时间：2026-03-23 17:44:04

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输入公共参数 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 apis；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 。
Version	String	是	操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

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

HTTP GET 请求结构示例：

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

```
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=37ac2f4fde00b0ac9bd9eadeb459b1bbec224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzh
ou&Nonce=23823223&SecretId=AKID*****
```

地域列表

本产品所有接口 Region 字段的可选值如下表所示。如果接口不支持该表中的所有地域，则会在接口文档中单独说明。

地域	取值
华北地区 (北京)	ap-beijing
西南地区 (成都)	ap-chengdu
华南地区 (广州)	ap-guangzhou
华东地区 (南京)	ap-nanjing
华东地区 (上海)	ap-shanghai
亚太东南 (新加坡)	ap-singapore

签名方法 v3

最近更新时间：2026-03-23 17:44:04

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：

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】创建一对密钥。

签名版本 v3 签名过程

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

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

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

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

假设用户的 SecretId 和 SecretKey 分别是：AKID***** 和

*****。用户想查看广州云服务器名为“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 中问号 (?) 后面的字符串内容, 例如: Limit=10&Offset=0。 注意: CanonicalQueryString 需要参考 RFC3986 进行 URLEncode 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=10b1a37a7301a02ca19a647ad722d5e43b4b3
cff309d421d85b46093f6ab6c4f
```

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/cv
m/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a64
7ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
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.DataMapper;

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)).to
LowerCase();
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}\n-x-tc-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=#{s
igned_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, MidpointRound
ing.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, 否则容易出错
DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}\"";

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

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

NodeJS

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

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

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

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

function main(){
  // 密钥参数
  // 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 OPENSSSL_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 OPENSSSL_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 OPENSSL_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* const 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, "\n\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\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%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：2026-03-23 17:44:05

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“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"，因此格式化后就为 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。

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

示例的拼接结果为：

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&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&Region=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() / 100
    0);
    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_KE
    Y"), "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("cvm.tencentcloudapi.com")
    buf.WriteString("/")
    buf.WriteString("?")

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

```
sort.Strings(keys)

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

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

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

PHP

```
<?php
// 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
// $param["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($param);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

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

# 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：2026-03-23 17:44:05

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：2026-03-23 17:44:05

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。

模型管理相关接口

修改模型

最近更新时间：2026-03-23 17:43:49

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

修改模型

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：ModifyModel。
Version	是	String	公共参数，本接口取值：2024-08-01。
Region	是	String	公共参数，详见产品支持的 地域列表 。
InstanceId	是	String	实例 示例值：ins-a7af1980
ID	是	String	模型ID 示例值：mod-fe2c8801
Name	否	String	模型名称 示例值：hello_eee
HttpProtocolType	否	String	协议类型：http/https 示例值：http
TargetPath	否	String	目标路径 示例值：/v122444
TargetHosts.N	否	Array of TargetHostDTO	目标服务器
CredentialID	否	String	凭据ID 示例值：agc-4440aa1d

参数名称	必选	类型	描述
CheckTargetCertsError	否	Boolean	https时, 是否检查证书合法 示例值: false
HttpProtocolVersion	否	String	http协议版本: 1.1/2.0 示例值: 1.1

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	结果集
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 编辑模型

输入示例

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

{
  "InstanceId": "ins-a7af1980",
  "ID": "mod-e6866e22",
  "Name": "hello"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "mod-e6866e22"
    },
    "RequestId": "d5f4707c-85e9-4fc3-b2d5-65f00afb1045"
  }
}
```

```
}  
  
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询模型列表

最近更新时间：2026-03-23 17:43:50

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询模型列表

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：DescribeModels。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 示例值：ins-a7af1980
Offset	否	Integer	分页参数 示例值：0
Limit	否	Integer	分页限制 示例值：10
IDs.N	否	Array of String	ID列表 示例值：["010101"]
NotIDs.N	否	Array of String	排除的ID列表 示例值：["0202020"]
CredentialID	否	String	凭据ID 示例值：0303030
Keyword	否	String	关键词 示例值：hehe
Sort	否	DescribeModelsSort	排序

3. 输出参数

参数名称	类型	描述
Data	DescribeModelsResponseVO	结果集
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 查询模型列表

输入示例

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

```
{
  "InstanceID": "ins-a7af1980",
  "Offset": 0,
  "Limit": 10,
  "IDs": [
    "010101"
  ],
  "NotIDs": [
    "0202020"
  ],
  "CredentialID": "0303030",
  "Keyword": "hehe",
  "Sort": {
    "CreateTime": 1,
    "LastUpdateTime": 1,
    "Name": -1
  }
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "Items": [],
      "Total": 0
    },
    "RequestId": "dc855e1e-41fd-4a12-9c19-4fdab33e96e3"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询模型详情

最近更新时间：2026-03-23 17:43:50

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询模型详情

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：DescribeModel。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 示例值：ins-a7af1980
ID	是	String	模型ID 示例值：mod-fe2c8801

3. 输出参数

参数名称	类型	描述
Data	DescribeModelResponseVO	结果集
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 查询模型详情

输入示例

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

<公共请求参数>

```
{
  "InstanceID": "ins-a7af1980",
  "ID": "mod-e6866e22"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "AppID": 1300273807,
      "CheckTargetCertsError": false,
      "CreateTime": "2026-02-03T07:35:59.803Z",
      "CredentialID": "",
      "CredentialName": "",
      "HttpProtocolType": "https",
      "HttpProtocolVersion": "1.1",
      "ID": "mod-e6866e22",
      "InstanceID": "ins-a7af1980",
      "LastUpdateTime": "2026-02-03T07:35:59.803Z",
      "ModelServiceCount": 0,
      "Name": "deepseek",
      "TargetHosts": [
        {
          "Host": "10.2.1.0",
          "Rank": 10
        }
      ],
      "TargetPath": "/v1",
      "Uin": "700001136234"
    },
    "RequestId": "3cd3dfe6-2aac-44f2-a886-03e4412eb618"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

删除模型

最近更新时间：2026-03-23 17:43:51

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

删除模型

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：DeleteModel。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 示例值：ins-a7af1980
IDs.N	是	Array of String	模型ID数组 示例值：["mod-edd55e4f"]

3. 输出参数

参数名称	类型	描述
Data	ResultIDsVO	结果集
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 删除模型

输入示例

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

{
  "InstanceID": "ins-a7af1980",
  "IDs": [
    "mod-edd55e4f"
  ]
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "IDs": [
        "mod-edd55e4f"
      ]
    },
    "RequestId": "fc5afc91-7bbc-4e9f-b165-969b30f5bf96"
  }
}
```

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: [CNB](#), [GitHub](#), [Gitee](#)

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

创建模型

最近更新时间：2026-03-23 17:43:51

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

创建模型

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：CreateModel。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 示例值：ins-a7af1980
Name	是	String	模型名称 示例值：hehehee
HttpProtocolType	是	String	协议类型：http/https 示例值：http
TargetPath	是	String	目标路径 示例值：/v12222
TargetHosts.N	是	Array of TargetHostDTO	目标服务器
CredentialID	否	String	凭据ID 示例值：agc-4440aa1d
CheckTargetCertsError	否	Boolean	https时，是否检查证书合法 示例值：true
HttpProtocolVersion	否	String	http协议版本：1.1/2.0 示例值：1.1

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	结果集
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 创建模型

输入示例

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

{
  "InstanceId": "ins-a7af1980",
  "Name": "deepseek",
  "HttpProtocolType": "https",
  "TargetPath": "/v1",
  "TargetHosts": [
    {
      "Host": "10.2.1.0",
      "Rank": 10
    }
  ],
  "CheckTargetCertsError": false,
  "HttpProtocolVersion": "1.1"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "Model-e6866e22"
    },
    "RequestId": "8e37dddc-3e52-46c6-89dd-f510ea3c36d3"
  }
}
```

```
}  
  
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

模型服务管理相关接口

编辑模型服务

最近更新时间：2026-04-29 01:11:06

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

修改模型服务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyModelService。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceID	是	String	实例 示例值：ins-****
ID	是	String	模型服务ID 示例值：mds-****
Name	否	String	模型服务名称 示例值：deepseek
Description	否	String	描述

参数名称	必选	类型	描述
			示例值: deepseek
TargetModels.N	否	Array of TargetModelDTO	模板模型列表
InvokeLimitConfigStatus	否	Boolean	是否开启限流 示例值: false
InvokeLimitConfig	否	InvokeLimitConfigDTO	限流配置
TokenLimitStatus	否	Boolean	是否开启token控制 示例值: false
TokenLimitConfig	否	TokenLimitConfigDTO	token控制
TmsStatus	否	Boolean	是否开启内容安全 示例值: false
TmsConfig	否	TmsConfigDTO	内容安全配置
IpWhiteStatus	否	Boolean	是否开启IP白名单 示例值: false
IpWhiteList.N	否	Array of String	IP白名单 示例值: ["1.1.1.1"]
IpBlackStatus	否	Boolean	是否开启IP黑名单 示例值: false
IpBlackList.N	否	Array of String	IP黑名单 示例值: ["1.1.1.1"]
PluginConfigs.N	否	Array of PluginConfigDTO	插件配置
Timeout	否	Integer	超时配置, 秒 示例值: 60
PromptModerateStatus	否	Boolean	是否开启提示词安全检测配置

参数名称	必选	类型	描述
			示例值: false
PromptModerateConfig	否	PromptModerateConfigDTO	提示词安全检测配置
SensitiveDataCheckStatus	否	Boolean	是否开启敏感数据检测 示例值: true
SensitiveDataCheckConfig	否	SensitiveDataCheckConfigDTO	敏感数据检测配置
TargetSelect	否	String	负载方式 枚举值: <ul style="list-style-type: none"> • random: 随机 • consistentHash: 会话保持 示例值: consistentHash
FindHostKeyMethod	否	String	会话判断方式 枚举值: <ul style="list-style-type: none"> • fromClientIP: 从客户端IP判断 • fromHeader: 从请求header判断 • autoDetect: 自动探测 示例值: autoDetect
HostKeyHeaderName	否	String	会话判断header名称 示例值: x-session-id
FallbackStatus	否	Boolean	是否开启备份模型 示例值: true
FallbackModels.N	否	Array of TargetModelDTO	备份模型

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	结果集

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 ModifyModelService

ModifyModelService

输入示例

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

{
  "InstanceID": "ins-*****",
  "ID": "mds-*****",
  "Name": "deepseek",
  "Description": "deepseek",
  "TargetModels": [
    {
      "ID": "mod-78bd8b5e",
      "Name": "deepseek",
      "Rank": 10
    }
  ],
  "InvokeLimitConfigStatus": false,
  "InvokeLimitConfig": {
    "Type": "tokenBucket",
    "TokenBucketMaxNum": 20000,
    "TokenBucketRate": 1,
    "FunnelMaxNum": 20000,
    "FunnelRate": 1,
    "SlidingWindowMaxNum": 20000,
    "SlidingWindowSize": 1,
    "TimeWindow": 20000,
    "TimeWindowInterval": 1
  },
  "TokenLimitStatus": false,
  "TokenLimitConfig": {
```

```
"LimitRequestBody": 10,
"LimitWindows": [
  {
    "Interval": 60,
    "Limit": 100
  }
],
"TmsStatus": false,
"TmsConfig": {
  "Scope": [],
  "Mode": "",
  "Action": "",
  "MergeCount": 0,
  "BizType": "",
  "InterceptMessage": ""
},
"IpWhiteStatus": false,
"IpWhiteList": [],
"IpBlackStatus": false,
"IpBlackList": [],
"PluginConfigs": [],
"Timeout": 60,
"PromptModerateStatus": false,
"PromptModerateConfig": {
  "Action": "watch",
  "InterceptMessage": ""
}
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "mds-*****"
    },
    "RequestId": "ebd06dc9-30ae-4f30-84eb-a1ed61d7a098"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询模型服务列表

最近更新时间：2026-03-23 17:43:53

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询模型服务列表

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：DescribeModelServices。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 示例值：ins-a7af1980
Offset	否	Integer	分页参数 示例值：0
Limit	否	Integer	分页限制 示例值：10
IDs.N	否	Array of String	ID列表 示例值：["mds-0101"]
NotIDs.N	否	Array of String	排除的ID列表 示例值：["mds-0101"]
Status	否	String	状态：normal, disabled 示例值：normal
Keyword	否	String	关键词 示例值：hehe_tes

参数名称	必选	类型	描述
ModelID	否	String	模型ID 示例值：mod-0101
Sort	否	DescribeModelServicesSort	排序

3. 输出参数

参数名称	类型	描述
Data	DescribeModelServicesResponseVO	结果集
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 查询模型服务

输入示例

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

{
  "InstanceID": "ins-a7af1980",
  "Offset": 0,
  "Limit": 10,
  "IDs": [
    "mds-0101"
  ],
  "NotIDs": [
    "mds-0101"
  ],
  "Status": "normal",
  "Keyword": "hehe_tes",
  "ModelID": "mod-0101",
  "Sort": {
    "CreateTime": 1,
```

```
"LastUpdateTime": 1,
"Name": 1,
"Status": -1
}
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "Items": [],
      "Total": 0
    },
    "RequestId": "fedea4c1-a673-4ac2-824c-73a2c9ef5d4f"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询模型服务详情

最近更新时间：2026-03-23 17:43:53

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询模型服务详情

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：DescribeModelService。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 示例值：ins-a7af1980
ID	是	String	模型服务ID 示例值：mds-ea8d96c6

3. 输出参数

参数名称	类型	描述
Data	DescribeModelServiceResponseVO	结果集
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 查询服务详情

输入示例

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

```
{
  "InstanceID": "ins-a7af1980",
  "ID": "mds-ea8d96c6"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "AppID": 1300273807,
      "CreateTime": "2026-02-03T11:22:25.627Z",
      "Description": "deepseek",
      "ID": "mds-ea8d96c6",
      "InstanceID": "ins-a7af1980",
      "InvokeLimitConfig": {
        "FunnelMaxNum": 0,
        "FunnelRate": 0,
        "SlidingWindowMaxNum": 0,
        "SlidingWindowSize": 0,
        "TimeWindow": 0,
        "TimeWindowInterval": 0,
        "TokenBucketMaxNum": 0,
        "TokenBucketRate": 0,
        "Type": ""
      },
      "InvokeLimitConfigStatus": false,
      "IpBlackList": [],
      "IpBlackStatus": false,
      "IpWhiteList": [],
      "IpWhiteStatus": false,
      "LastUpdateTime": "2026-02-03T11:23:42.212Z",
      "ModelNames": [
```

```
"deepseek-r1"
],
"Name": "deepseek-r101",
"PathMatchType": "prefix",
"PluginConfigs": [],
"PubPath": "/v1",
"RelateAgentAppNum": 0,
"Status": "normal",
"Timeout": 10,
"TmsConfig": {
  "Action": "",
  "BizType": "",
  "InterceptMessage": "",
  "MergeCount": 0,
  "Mode": "",
  "Scope": null
},
"TmsStatus": false,
"TokenLimitConfig": {
  "LimitRequestBody": 0,
  "LimitWindows": []
},
"TokenLimitStatus": false,
"Uin": "700001136234"
},
"RequestId": "d404956c-e692-4909-89b9-a92c2066c7bd"
}
```

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: [CNB](#), [GitHub](#), [Gitee](#)

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

删除模型服务

最近更新时间：2026-03-23 17:43:54

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

删除模型服务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteModelService。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 示例值：ins-a7af1980
ID	是	String	模型服务ID 示例值：mds-e9b168e4

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	结果集
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 删除模型服务

输入示例

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

{
  "InstanceID": "ins-a7af1980",
  "ID": "mds-e9b168e4"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "mds-e9b168e4"
    },
    "RequestId": "459519b6-c90e-4b27-bf5f-2e900a888045"
  }
}
```

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

创建模型服务

最近更新时间：2026-04-29 01:11:07

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

创建模型服务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateModelService。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例 示例值：ins-****
Name	是	String	模型服务名称 示例值：testmodel
Description	是	String	描述 示例值：testmodel
PubPath	是	String	访问路径 示例值：/model

参数名称	必选	类型	描述
TargetModels.N	是	Array of TargetModelDTO	模型ID列表
PathMatchType	否	String	路径匹配类型: prefix 前缀匹配(不送默认); absolute 绝对匹配; regex正则匹配; 示例值: prefix
InvokeLimitConfigStatus	否	Boolean	是否开启限流 示例值: false
InvokeLimitConfig	否	InvokeLimitConfigDTO	限流配置
TokenLimitStatus	否	Boolean	是否开启token控制 示例值: false
TokenLimitConfig	否	TokenLimitConfigDTO	token控制
TmsStatus	否	Boolean	是否开启内容安全 示例值: false
TmsConfig	否	TmsConfigDTO	内容安全配置
IpWhiteStatus	否	Boolean	是否开启IP白名单 示例值: false
IpWhiteList.N	否	Array of String	IP白名单 示例值: ["1.1.1.1"]
IpBlackList.N	否	Array of String	IP黑名单 示例值: ["1.1.1.1"]
PluginConfigs.N	否	Array of PluginConfigDTO	插件配置
Timeout	否	Integer	超时配置, 秒 示例值: 60
PromptModerateStatus	否	Boolean	是否开启提示词安全检测

参数名称	必选	类型	描述
			示例值: true
PromptModerateConfig	否	PromptModerateConfigDTO	提示词安全检测配置
SensitiveDataCheckStatus	否	Boolean	是否开启敏感数据检测 示例值: true
SensitiveDataCheckConfig	否	SensitiveDataCheckConfigDTO	敏感数据检测配置
TargetSelect	否	String	负载方式 枚举值: <ul style="list-style-type: none"> • random: 随机 • consistentHash: 会话保持 示例值: consistentHash
FindHostKeyMethod	否	String	会话判断方式 枚举值: <ul style="list-style-type: none"> • fromClientIP: 客户端IP • fromHeader: 通过header值 • autoDetect: 自动探测 示例值: autoDetect
HostKeyHeaderName	否	String	会话判定方式为fromHeader时会话的header名称 示例值: x-session-id
FallbackStatus	否	Boolean	是否启用Fallback模型 示例值: false
FallbackModels.N	否	Array of TargetModelDTO	Fallback模型配置

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	结果集

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 CreateModelService

CreateModelService

输入示例

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

{
  "InstanceID": "ins-*****",
  "Name": "testmodel",
  "Description": "testmodel",
  "PubPath": "/model",
  "TargetModels": [
    {
      "ID": "ins-*****",
      "Name": "deepseek",
      "Rank": 10
    }
  ],
  "PathMatchType": "prefix",
  "InvokeLimitConfigStatus": false,
  "InvokeLimitConfig": {
    "Type": "tokenBucket",
    "TokenBucketMaxNum": 20000,
    "TokenBucketRate": 1,
    "FunnelMaxNum": 20000,
    "FunnelRate": 1,
    "SlidingWindowMaxNum": 20000,
    "SlidingWindowSize": 1,
    "TimeWindow": 20000,
    "TimeWindowInterval": 1,
    "Timeout": 60
  },
}
```

```
"TokenLimitStatus": false,
"TokenLimitConfig": {
  "LimitRequestBody": 10,
  "LimitWindows": [
    {
      "Interval": 60,
      "Limit": 100
    }
  ],
},
"TmsStatus": false,
"TmsConfig": {
  "Scope": [],
  "Mode": "",
  "Action": "",
  "MergeCount": 0,
  "BizType": "",
  "InterceptMessage": ""
},
"IpWhiteStatus": false,
"IpWhiteList": [],
"IpBlackList": [],
"PluginConfigs": [],
"Timeout": 60,
"PromptModerateStatus": true,
"PromptModerateConfig": {
  "Action": "intercept",
  "InterceptMessage": "intercept"
},
"TargetSelect": "consistentHash",
"FindHostKeyMethod": "autoDetect",
"HostKeyHeaderName": "x-session-id",
"FallbackStatus": false
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "mds-572bf562"
    },
  },
  "RequestId": "3766ec52-78e1-478a-a83d-686981c94a92"
}
```

```
}  
  
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

应用管理相关接口

修改app的model service关联

最近更新时间：2026-03-23 17:43:55

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

编辑app的model service关联

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：ModifyAgentAppModelServices。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：ins-a7af1980
ID	是	String	应用ID 示例值：aga-41d42939
ModelServices.N	是	Array of AgentAppModelServiceDTO	关联的model service

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	app id
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 修改应用关联模型服务

输入示例

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

{
  "InstanceID": "ins-a7af1980",
  "ID": "aga-41d42939",
  "ModelServices": [
    {
      "ID": "mds-ea8d96c6"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "aga-41d42939"
    },
    "RequestId": "075b6412-222d-4994-9d93-c04d922c6b6d"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

修改app

最近更新时间：2026-03-23 17:43:55

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

修改app

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：ModifyAgentApp。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：ins-xxxxxxxx
ID	是	String	应用ID 示例值：aga-xxxxxxxx
Name	否	String	名称 示例值：testname
OAuth2ResourceServerID	否	String	OAuth2资源服务器ID 示例值：ors-xxxxxxxx
Description	否	String	描述 示例值：description 01

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	app id

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 修改app

修改app

输入示例

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

{
  "ID": "aga-71702335",
  "InstanceID": "ins-e6fbc9b9",
  "Name": "测试app",
  "Description": "测试desp"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "aga-71702335"
    },
    "RequestId": "ccb7929f-710a-4e1a-befb-192ec3f28da0"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询app列表

最近更新时间：2026-03-23 17:43:56

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询app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：DescribeAgentApps。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Limit	是	Integer	分页大小 示例值：10
Offset	否	Integer	分页偏移量 示例值：0
IDs.N	否	Array of String	服务ID数组 示例值：["aga-xxxxxxxx"]
InstanceID	否	String	实例ID 示例值：ins-xxxxxxxx
NotIDs.N	否	Array of String	notID列表 示例值：["aga-xxxxxxxx"]
Status	否	String	状态：normal正常状态，disabled下线状态 示例值：normal
Keyword	否	String	关键词 示例值：key

参数名称	必选	类型	描述
AuthType	否	String	认证类型: apiKey, secretKey 示例值: apiKey
Sort	否	DescribeAgentAppsSortDTO	排序
AgentCredentialID	否	String	凭据ID 示例值: agc-xxxxxxx

3. 输出参数

参数名称	类型	描述
Data	DescribeAgentAppsResp	app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 查询app列表

查询app列表

输入示例

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

{
  "Offset": 0,
  "Limit": 10,
  "InstanceID": "ins-e6fbc9b9"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "Items": [
```

```
{
  "ApiKeys": [
    "sk-*****"
  ],
  "AppID": 1300273807,
  "AuthType": "apiKey",
  "CreateTime": "2025-07-14T06:09:49.354Z",
  "Description": "desp",
  "ID": "aga-71702335",
  "InstanceID": "ins-e6fbc9b9",
  "Name": "测试",
  "SecretKeys": [],
  "Status": "normal",
  "Uin": "700001136234",
  "UpdateTime": "2025-07-14T06:09:49.354Z"
},
{
  "Total": 1
},
{
  "RequestId": "73c48cae-cbf1-4377-b230-f1f42c89ff4c"
}
}
```

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

- Tencent Cloud SDK 3.0 for Ruby: [CNB](#), [GitHub](#), [Gitee](#)

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询app model服务关联列表

最近更新时间：2026-03-23 17:43:56

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询app modelService关联列表

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：DescribeAgentAppModelServices。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

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）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询应用关联的模型服务

输入示例

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

<公共请求参数>

```
{}
```

输出示例

```
{
  "Response": {
    "RequestId": "44bd5c9a-9da8-401e-a462-d0e6b44c5271"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询app mcp关联列表

最近更新时间：2026-03-23 17:43:57

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询app mcpServer关联列表

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：DescribeAgentAppMcpServers。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Limit	是	Integer	分页大小 示例值：10
Offset	否	Integer	分页偏移量 示例值：0
InstanceId	否	String	实例ID 示例值：ins-a7af1980
AgentAppIDs.N	否	Array of String	应用ID 示例值：["aga-05b85106"]
McpServerIDs.N	否	Array of String	关联的mcp 示例值：["mcp-2fc41d77"]
AgentCredentialIDs.N	否	Array of String	凭据ID 示例值：["agc-xxxxxxxxx"]
Keyword	否	String	关键词 示例值：keyword

3. 输出参数

参数名称	类型	描述
Data	DescribeAgentAppMcpServersResp	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 查询app mcp关联列表

输入示例

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

{
  "Limit": 10,
  "Offset": 0,
  "InstanceID": "ins-a7af1980",
  "AgentAppIDs": [
    "aga-05b85106"
  ],
  "McpServerIDs": [
    "mcp-2fc41d77"
  ]
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "Items": [
        {
          "AgentCredentialID": "",
          "AgentCredentialVO": {
```

```
"AppID": 0,
"Content": {
  "Headers": [],
  "STSService": "",
  "STSSystem": ""
},
"CreateTime": "0001-01-01T00:00:00Z",
"ID": "",
"InstanceID": "",
"LastUpdateTime": "0001-01-01T00:00:00Z",
"Name": "",
"RelateAgentAppNum": 0,
"RelateMcpServerNum": 0,
"Status": "",
"Type": "",
"Uin": ""
},
"ID": "aga-05b85106",
"McpServerVO": {
  "AppID": 1300273807,
  "Catalogs": null,
  "CategoryIDs": [],
  "CheckTargetCertsError": false,
  "CreateTime": "2025-08-07T09:55:58.696Z",
  "CustomHttpHost": "",
  "Description": "testmcp",
  "HttpHostType": "targetHost",
  "HttpProtocolType": "http",
  "ID": "mcp-2fc41d77",
  "InstanceID": "ins-a7af1980",
  "InvokeLimitConfig": {
    "FunnelMaxNum": 20000,
    "FunnelRate": 1,
    "SlidingWindowMaxNum": 20000,
    "SlidingWindowSize": 1,
    "TimeWindow": 20000,
    "TimeWindowInterval": 1,
    "TokenBucketMaxNum": 20000,
    "TokenBucketRate": 1,
    "Type": "tokenBucket"
  },
  "InvokeLimitConfigStatus": false,
  "IpBlackConfig": {
```

```
"Comment": "",
"EffectTimes": [],
"EffectType": "always",
"Ips": []
},
"IpBlackStatus": false,
"IpWhiteConfig": {
  "Comment": "",
  "EffectTimes": [],
  "EffectType": "always",
  "Ips": []
},
"IpWhiteStatus": false,
"LabelIDs": [],
"Labels": null,
"LastUpdateTime": "2025-12-04T06:07:53.241Z",
"McpSecurityRulesVO": [],
"McpVersion": "2024-11-05",
"Mode": "wrap",
"Name": "testmcp",
"PluginConfigs": [],
"RelateAgentAppNum": 0,
"Status": "normal",
"TargetHosts": [],
"TargetPath": "/",
"TargetSelect": "random",
"Timeout": 60,
"ToolConfigs": [],
"ToolMessage": "",
"ToolNum": 0,
"Tools": [],
"Uin": "700001136234",
"Url": "",
"UrlObj": {
  "SSEUrl": "http://10.0.0.4/_llmsgw_/sse/aga-05b85106/mcp-2fc41d77",
  "StreamAbleUrl": "http://10.0.0.4/_llmsgw_/mcp/aga-05b85106/mcp-2fc41d77"
},
"WrapPaasID": "app-be27914b",
"WrapServices": [
  "svr-7e7f07e8"
]
},
"NeedAuth": false,
```

```
"RelateTime": "2025-12-02T03:59:02.13Z",
"SSEResourceIdentifier": ""
},
],
"Total": 1
},
"RequestId": "4958f16e-4ab8-4429-8ad5-167f6b25799b"
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询app详情

最近更新时间：2026-03-23 17:43:57

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询app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：DescribeAgentApp。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：ins-xxxxxxxx
ID	是	String	app id 示例值：aga-xxxxxxxx

3. 输出参数

参数名称	类型	描述
Data	DescribeAgentAppResp	app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 查询app详情

查询app详情

输入示例

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

```
{
  "ID": "aga-71702335",
  "InstanceID": "ins-e6fbc9b9"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ApiKeys": [
        "sk-d3b6f59b27B792B7dd6aeb26554D0DAC"
      ],
      "AppID": 1300273807,
      "AuthType": "apiKey",
      "CreateTime": "2025-07-14T06:09:49.354Z",
      "Description": "测试desp",
      "ID": "aga-71702335",
      "InstanceID": "ins-e6fbc9b9",
      "Name": "测试app",
      "SecretKeys": [],
      "Status": "normal",
      "Uin": "700001136234",
      "UpdateTime": "2025-07-14T06:13:55.995Z"
    },
    "RequestId": "7020c298-312c-4cc6-9f77-66d64c22936e"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

删除app的model service关联

最近更新时间：2026-03-23 17:43:58

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

删除app的model service关联

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：DeleteAgentAppModelServices。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：ins-a7af1980
ID	是	String	应用ID 示例值：aga-41d42939
ModelServiceIds.N	是	Array of String	关联的model service id 示例值：["mds-ea8d96c6"]

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	app id
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 删除应用关联模型服务

输入示例

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

{
  "InstanceID": "ins-a7af1980",
  "ID": "aga-41d42939",
  "ModelServiceIDs": [
    "mds-ea8d96c6"
  ]
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "aga-41d42939"
    },
    "RequestId": "02dd3ade-7813-4f83-9024-5948f207e02c"
  }
}
```

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: [CNB](#), [GitHub](#), [Gitee](#)

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

删除app的mcp server

最近更新时间：2026-03-23 17:43:58

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

删除app的mcp server

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：DeleteAgentAppMcpServers。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：ins-xxxxxxxx
ID	是	String	应用ID 示例值：aga-xxxxxxxx
McpServerIds.N	是	Array of String	mcp server id数组 示例值：["mcp-xxxxxxxx"]

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	app id
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app的mcp server

删除app的mcp server

输入示例

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

<公共请求参数>

```
{
  "InstanceID": "ins-e6fbc9b9",
  "ID": "aga-71702335",
  "McpServerIDs": [
    "mcp-6a63b40b"
  ]
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "aga-71702335"
    },
    "RequestId": "35433e7a-0634-4b3b-ab95-2ea7de191c38"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

删除app

最近更新时间：2026-03-23 17:43:59

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

删除app

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：DeleteAgentApp。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：ins-xxxxxxxx
ID	是	String	应用ID 示例值：aga-xxxxxxxx

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	app id
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app

删除app

输入示例

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

{
  "InstanceID": "ins-9c4a1db3",
  "ID": "aga-2b2e0214"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "aga-2b2e0214"
    },
    "RequestId": "aaa7c5ac-d5fc-4f34-81a0-aeb0af415e28"
  }
}
```

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

创建app的model service关联

最近更新时间：2026-03-23 17:43:59

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

创建app的model service关联

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： CreateAgentAppModelServices。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：ins-b1dacafc
ID	是	String	应用ID 示例值：aga-3f83397c
ModelServices.N	是	Array of AgentAppModelServiceDTO	关联的model service

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	app id
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 创建应用关联模型服务

输入示例

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

{
  "InstanceID": "ins-a7af1980",
  "ID": "aga-41d42939",
  "ModelServices": [
    {
      "ID": "mds-ea8d96c6",
      "InvokeLimitConfigStatus": false
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "aga-41d42939"
    },
    "RequestId": "6a58841b-8aca-43bb-8894-8b9e65619267"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

创建app的mcp server关联

最近更新时间：2026-03-23 17:43:59

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

创建app的mcp server关联

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： CreateAgentAppMcpServers。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：ins-xxxxxxxx
ID	是	String	应用ID 示例值：aga-xxxxxxxx
McpServers.N	是	Array of AgentAppMcpServerDTO	关联的mcp server

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	app id
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app的mcp server关联

创建app的mcp server关联

输入示例

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

<公共请求参数>

```
{
  "ID": "aga-71702335",
  "InstanceID": "ins-e6fbc9b9",
  "McpServers": [
    {
      "ID": "mcp-6a63b40b",
      "NeedAuth": false
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "6a63b40b"
    },
    "RequestId": "c613fd7b-5f4b-41b9-80f0-8c18ab8cccdf"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

创建app

最近更新时间：2026-03-23 17:44:00

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

创建app

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：CreateAgentApp。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceID	是	String	实例ID 示例值：ins-xxxxxxxxx
Name	是	String	名称 示例值：name01
AuthType	是	String	认证类型 示例值：apiKey
OAuth2ResourceServerID	否	String	OAuth2资源服务器ID 示例值：ors-be1312ba
Description	否	String	描述 示例值：description 01

3. 输出参数

参数名称	类型	描述
Data	CreateAgentAppResp	app id

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 创建app

创建app

输入示例

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

{
  "InstanceId": "ins-e6fbc9b9",
  "Name": "hello1",
  "AuthType": "apiKey"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "aga-2b2e0214"
    },
    "RequestId": "aaa7c5ac-d5fc-4f34-81a0-aeb0af415e28"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

MCP 管理相关接口

修改mcp server

最近更新时间：2026-03-26 01:10:45

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

修改mcp server

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：ModifyMcpServer。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ID	是	String	mcp server ID 示例值：mcp-xxxxxxxx
Mode	是	String	模式：proxy代理模式；wrap封装模式； 示例值：proxy
McpVersion	是	String	版本号：2024-11-05 2025-03-26 示例值：2024-11-05
InstanceId	是	String	实例ID 示例值：ins-xxxxxxxx
Name	是	String	名称 示例值：testname
Description	否	String	描述 示例值：testdesc
WrapServices.N	否	Array of String	封装服务列表 示例值：["svr-xxxxxxxx"]

参数名称	必选	类型	描述
TargetSelect	否	String	负载方式, robin random consistentHash 示例值: random
TargetHosts.N	否	Array of TargetHostDTO	目标服务器
HttpProtocolType	否	String	后端协议: http https 示例值: http
CheckTargetCertsError	否	Boolean	证书检查 示例值: false
TargetPath	否	String	目标路径 示例值: /sse
InvokeLimitConfigStatus	否	Boolean	流量控制开启状态 示例值: false
InvokeLimitConfig	否	InvokeLimitConfigDTO	流量控制配置
IpWhiteStatus	否	Boolean	IP白名单开启状态 示例值: false
IpWhiteConfig	否	IpConfig	IP白名单配置
IpBlackStatus	否	Boolean	IP黑名单开启状态 示例值: false
IpBlackConfig	否	IpConfig	IP黑名单配置
TargetHostType	否	Integer	目标Host类型 0 默认 1 vpc 示例值: 1
CustomHttpHost	否	String	自定义host 示例值: test.com
HttpHostType	否	String	Http 请求host类型: useRequestHost 保持源请求; host targetHost 修正为源站host; customHost 自定义host 示例值: useRequestHost
Timeout	否	Integer	请求的超时时间 示例值: 10
McpSecurityRules.N	否	Array of McpSecurityRule	安全规则集
ToolConfigs.N	否	Array of ToolConfigDTO	工具集配置 (openapi可能会用到)
WrapPaasID	否	String	封装的API分组ID 示例值: app-010101

参数名称	必选	类型	描述
PluginConfigs.N	否	Array of PluginConfigDTO	插件配置

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	mcp server ID
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mcp server

修改mcp server

输入示例

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

{
  "InstanceID": "ins-9c4a1db3",
  "ID": "mcp-aa12c447",
  "Mode": "proxy",
  "McpVersion": "2024-11-05",
  "Name": "testmcp",
  "Description": "",
  "WrapServices": [],
  "TargetSelect": "random",
  "TargetHosts": [
    {
      "Host": "127.0.0.1:3000",
      "Rank": 10
    }
  ],
  "HttpProtocolType": "http",
  "CheckTargetCertsError": false,
```

```
"TargetPath": "/sse",
"InvokeLimitConfigStatus": false,
"InvokeLimitConfig": {
  "Type": "tokenBucket",
  "TokenBucketMaxNum": 20000,
  "TokenBucketRate": 1,
  "FunnelMaxNum": 20000,
  "FunnelRate": 1,
  "SlidingWindowMaxNum": 20000,
  "SlidingWindowSize": 1,
  "TimeWindow": 20000,
  "TimeWindowInterval": 1
},
"IpWhiteStatus": false,
"IpWhiteConfig": {
  "Ips": [],
  "EffectType": "always",
  "EffectTimes": [],
  "Comment": ""
},
"IpBlackStatus": false,
"IpBlackConfig": {
  "Ips": [],
  "EffectType": "always",
  "EffectTimes": [],
  "Comment": ""
},
"CustomHttpHost": "",
"HttpHostType": "targetHost",
"Timeout": 60,
"McpSecurityRules": [
  {
    "ID": "mr-2bdbbf32",
    "Act": "watch",
    "Status": "open"
  }
]
}
```

输出示例

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

```
"Data": {
  "ID": "mcp-aa12c447"
},
"RequestId": "388289da-01b2-4be6-8ba9-7b40028ac05b"
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询mcp server列表

最近更新时间：2026-03-23 17:44:01

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询mcp server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：DescribeMcpServers。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Limit	是	Integer	分页大小 示例值：10
Offset	否	Integer	分页偏移量 示例值：0
InstanceId	否	String	实例ID 示例值：ins-xxxxxxxx
Statuses.N	否	Array of String	状态数组：normal正常状态，disabled下线状态 示例值：["normal"]
Keyword	否	String	关键词 示例值：filter
IDs.N	否	Array of String	服务ID数组 示例值：["mcp-xxxxxxxx"]
Modes.N	否	Array of String	模式：proxy代理模式；wrap封装模式； 示例值：["wrap"]

参数名称	必选	类型	描述
McpSecurityRuleID	否	String	绑定的安全规则ID 示例值：mc-xxxxxxxx
McpSecurityRuleAct	否	String	绑定安全规则的处置动作（填写时McpSecurityRuleID得必填） 示例值：filter
PluginID	否	String	已绑定插件id 示例值：plg-xxxxxxxx

3. 输出参数

参数名称	类型	描述
Data	DescribeMcpServersResponseVO	mcp server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 查询mcp server列表

查询mcp server列表

输入示例

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

{
  "InstanceID": "ins-9c4a1db3",
  "Offset": 0,
  "Limit": 10,
  "Statuses": [],
  "Modes": [],
  "Keyword": "test"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "Total": 1,
      "Items": [
        {
          "AppID": 1300273807,
          "Uin": "700001136234",
          "InstanceID": "ins-9c4a1db3",
          "Name": "test",
          "Description": "",
          "LabelIDs": [],
          "CategoryIDs": [],
          "Mode": "proxy",
          "McpVersion": "2024-11-05",
          "WrapServices": [],
          "TargetSelect": "random",
          "TargetHosts": [
            {
              "Host": "127.0.0.1:3002",
              "Rank": 10
            }
          ],
          "HttpProtocolType": "http",
          "CheckTargetCertsError": false,
          "TargetPath": "/sse",
          "InvokeLimitConfigStatus": false,
          "InvokeLimitConfig": {
            "Type": "tokenBucket",
            "TokenBucketMaxNum": 20000,
            "TokenBucketRate": 1,
            "FunnelMaxNum": 20000,
            "FunnelRate": 1,
            "SlidingWindowMaxNum": 20000,
            "SlidingWindowSize": 1,
            "TimeWindow": 20000,
            "TimeWindowInterval": 1
          },
          "IpWhiteStatus": false,
          "IpWhiteConfig": {
            "Ips": [],
            "EffectType": "always",
            "EffectTimes": [],
```

```
"Comment": ""
},
"IpBlackStatus": false,
"IpBlackConfig": {
  "Ips": [],
  "EffectType": "always",
  "EffectTimes": [],
  "Comment": ""
},
"CustomHttpHost": "",
"HttpHostType": "targetHost",
"Timeout": 60,
"ID": "mcp-8d5c4195",
"Status": "disabled",
"Url": "http://test.com/_llmsgw_/mcp?server=mcp-8d5c4195&paasid=app-83a7a4aa",
"App": {
  "ID": "app-83a7a4aa",
  "Name": "test"
},
"Catalogs": [],
"Labels": [],
"ToolNum": -1,
"CreateTime": "2025-05-20T08:24:18.702Z",
"LastUpdateTime": "2025-05-20T08:42:05.916Z",
"McpSecurityRulesVO": [
  {
    "ID": "mr-2bdbbf32",
    "Type": "sensitive_data_check",
    "BodyType": "",
    "IconType": "code",
    "Name": "验证码",
    "Description": "识别mcp服务调用过程中, 响应参数可能存在的验证码",
    "RiskLevel": "middle",
    "VersionNumber": "20250509_v1",
    "Status": "open",
    "Act": "watch",
    "SupportActs": [
      "watch",
      "desensitization",
      "filter"
    ]
  }
]
```

```
}  
]  
},  
"RequestId": "3fcd8e59-0c84-434e-9f07-9091f1de997d"  
}  
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询mcp server详情

最近更新时间：2026-03-23 17:44:01

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询mcp server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：DescribeMcpServer。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：ins-xxxxxxxx
ID	是	String	mcp server ID 示例值：mcp-xxxxxxxx

3. 输出参数

参数名称	类型	描述
Data	DescribeMcpServerResponseVO	mcp server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 查询mcp server详情

查询mcp server详情

输入示例

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

{
  "InstanceID": "ins-9c4a1db3",
  "ID": "mcp-8d5c4195"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "AppID": 1300273807,
      "Uin": "700001136234",
      "InstanceID": "ins-9c4a1db3",
      "Name": "test",
      "Description": "",
      "LabelIDs": [],
      "CategoryIDs": [],
      "Mode": "proxy",
      "McpVersion": "2024-11-05",
      "WrapServices": [],
      "TargetSelect": "random",
      "TargetHosts": [
        {
          "Host": "127.0.0.1:3002",
          "Rank": 10
        }
      ],
      "HttpProtocolType": "http",
      "CheckTargetCertsError": false,
      "TargetPath": "/sse",
      "InvokeLimitConfigStatus": false,
      "InvokeLimitConfig": {
        "Type": "tokenBucket",
```

```
"TokenBucketMaxNum": 20000,
"TokenBucketRate": 1,
"FunnelMaxNum": 20000,
"FunnelRate": 1,
"SlidingWindowMaxNum": 20000,
"SlidingWindowSize": 1,
"TimeWindow": 20000,
"TimeWindowInterval": 1
},
"IpWhiteStatus": false,
"IpWhiteConfig": {
  "Ips": [],
  "EffectType": "always",
  "EffectTimes": [],
  "Comment": ""
},
"IpBlackStatus": false,
"IpBlackConfig": {
  "Ips": [],
  "EffectType": "always",
  "EffectTimes": [],
  "Comment": ""
},
"CustomHttpHost": "",
"HttpHostType": "targetHost",
"Timeout": 60,
"ID": "mcp-8d5c4195",
"Status": "disabled",
"Url": "http://test.com/_llmsgw_/mcp?server=mcp-8d5c4195&paasid=app-83a7a4aa",
"App": {
  "ID": "app-83a7a4aa",
  "Name": "test"
},
"Catalogs": [],
"Labels": [],
"ToolNum": -1,
"CreateTime": "2025-05-20T08:24:18.702Z",
"LastUpdateTime": "2025-05-20T08:42:05.916Z",
"McpSecurityRulesVO": [
  {
    "ID": "mr-2bdbbf32",
    "Type": "sensitive_data_check",
    "BodyType": "",
```

```
"IconType": "code",
"Name": "验证码",
"Description": "识别mcp服务调用过程中, 响应参数可能存在的验证码",
"RiskLevel": "middle",
"VersionNumber": "20250509_v1",
"Status": "open",
"Act": "watch",
"SupportActs": [
  "watch",
  "desensitization",
  "filter"
]
},
"RequestId": "7ecfb435-bc50-49b8-b206-4278f2b08db1"
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

删除mcp server

最近更新时间：2026-03-23 17:44:02

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

删除mcp server

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：DeleteMcpServer。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
InstanceId	是	String	实例ID 示例值：ins-xxxxxxxx
ID	是	String	mcp server ID 示例值：mcp-xxxxxxxx

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	mcp server ID
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mcp server

删除mcp server

输入示例

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

{
  "InstanceID": "ins-9c4a1db3",
  "ID": "mcp-aa12c447"
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "mcp-aa12c447"
    },
    "RequestId": "1eed7b01-8b36-4b06-97c2-242e461856db"
  }
}
```

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

创建mcp server

最近更新时间：2026-03-23 17:44:02

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

创建mcp server

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：CreateMcpServer。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Mode	是	String	模式：proxy代理模式；wrap封装模式； 示例值：proxy
McpVersion	是	String	版本号：2024-11-05 2025-03-26 示例值：2024-11-05
InstanceId	是	String	实例ID 示例值：ins-xxxxxxxxx
Name	是	String	名称 示例值：testname
Description	是	String	描述 示例值：testdesc
WrapServices.N	否	Array of String	封装服务列表 示例值：["svr-xxxxxxxxx"]
TargetSelect	否	String	负载方式，robin random consistentHash 示例值：random

参数名称	必选	类型	描述
TargetHosts.N	否	Array of TargetHostDTO	目标服务器
HttpProtocolType	否	String	后端协议: http https 示例值: http
CheckTargetCertsError	否	Boolean	证书检查 示例值: false
TargetPath	否	String	目标路径 示例值: /sse
InvokeLimitConfigStatus	否	Boolean	流量控制开启状态 示例值: false
InvokeLimitConfig	否	InvokeLimitConfigDTO	流量控制配置
IpWhiteStatus	否	Boolean	IP白名单开启状态 示例值: false
IpWhiteConfig	否	IpConfig	IP白名单配置
IpBlackStatus	否	Boolean	IP黑名单开启状态 示例值: false
IpBlackConfig	否	IpConfig	IP黑名单配置
CustomHttpHost	否	String	自定义host 示例值: test.com
HttpHostType	否	String	Http 请求host类型: useRequestHost 保持源请求; host targetHost 修正为源站host; customHost 自定义host 示例值: useRequestHost
Timeout	否	Integer	请求的超时时间 示例值: 10
McpSecurityRules.N	否	Array of McpSecurityRule	安全规则集
ToolConfigs.N	否	Array of ToolConfigDTO	工具集配置 (openapi时或许用的是)
WrapPaasID	否	String	封装的API分组ID 示例值: paasid
PluginConfigs.N	否	Array of PluginConfigDTO	插件配置

3. 输出参数

参数名称	类型	描述
Data	ResultIDVO	mcp server ID
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mcp server

创建mcp server

输入示例

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

{
  "InstanceId": "ins-9c4a1db3",
  "Mode": "proxy",
  "McpVersion": "2024-11-05",
  "Name": "testmcp",
  "Description": "",
  "WrapServices": [],
  "TargetSelect": "random",
  "TargetHosts": [
    {
      "Host": "127.0.0.1:3002",
      "Rank": 10
    }
  ],
  "HttpProtocolType": "http",
  "CheckTargetCertsError": false,
  "TargetPath": "/sse",
  "InvokeLimitConfigStatus": false,
  "InvokeLimitConfig": {
    "Type": "tokenBucket",
    "TokenBucketMaxNum": 20000,
    "TokenBucketRate": 1,
    "FunnelMaxNum": 20000,
    "FunnelRate": 1,
```

```
"SlidingWindowMaxNum": 20000,
"SlidingWindowSize": 1,
"TimeWindow": 20000,
"TimeWindowInterval": 1
},
"IpWhiteStatus": false,
"IpWhiteConfig": {
  "Ips": [],
  "EffectType": "always",
  "EffectTimes": [],
  "Comment": ""
},
"IpBlackStatus": false,
"IpBlackConfig": {
  "Ips": [],
  "EffectType": "always",
  "EffectTimes": [],
  "Comment": ""
},
"CustomHttpHost": "",
"HttpHostType": "targetHost",
"Timeout": 60,
"McpSecurityRules": [
  {
    "ID": "mr-2bdbbf32",
    "Act": "watch",
    "Status": "open"
  }
]
}
```

输出示例

```
{
  "Response": {
    "Data": {
      "ID": "mcp-aa12c447"
    },
    "RequestId": "388289da-01b2-4be6-8ba9-7b40028ac05b"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

凭据管理相关接口

修改凭据

最近更新时间：2026-03-24 01:11:24

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

修改Credential

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：ModifyAgentCredential。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

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）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 修改Credential

修改Credential

输入示例

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

```
{}
```

输出示例

```
{
  "Response": {
    "RequestId": "0cd7d9e9-5eb8-456d-b50d-747c5b236ef8"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询凭据列表

最近更新时间：2026-03-24 01:11:24

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询Credential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：DescribeAgentCredentials。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

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）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询Credential列表

查询Credential列表

输入示例

```
POST / HTTP/1.1
Host: apis.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: DescribeAgentCredentials
```

<公共请求参数>

```
{}
```

输出示例

```
{
  "Response": {
    "RequestId": "77fc4cfd-ca44-4986-9f71-9175b3aaaaca"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

查询凭据详情

最近更新时间：2026-03-24 01:11:24

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

查询Credential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：DescribeAgentCredential。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

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）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询Credential详情

查询Credential详情

输入示例

```
POST / HTTP/1.1
Host: apis.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: DescribeAgentCredential
```

<公共请求参数>

```
{}
```

输出示例

```
{
  "Response": {
    "RequestId": "fb5e8042-5259-4fe3-b07d-cdec73a5554"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

删除凭据

最近更新时间：2026-03-24 01:11:24

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

删除Credential

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：DeleteAgentCredential。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

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）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 删除Credential

删除Credential

输入示例

```
POST / HTTP/1.1
Host: apis.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: DisableAgentCredential
```

```
<公共请求参数>
```

```
{}
```

输出示例

```
{
  "Response": {
    "RequestId": "3ff5f982-8e21-42e9-834a-e6f4e50bb83d"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

创建凭据

最近更新时间：2026-03-24 01:11:25

1. 接口描述

接口请求域名：apis.tencentcloudapi.com。

创建Credential

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：CreateAgentCredential。
Version	是	String	公共参数 ，本接口取值：2024-08-01。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。

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）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 创建Credential

创建Credential

输入示例

```
POST / HTTP/1.1
Host: apis.tencentcloudapi.com
Content-Type: application/json
```

```
X-TC-Action: CreateAgentCredential
```

```
<公共请求参数>
```

```
{}
```

输出示例

```
{
  "Response": {
    "RequestId": "70f5a87e-91d0-4b6a-8720-efd5f3ccec9e"
  }
}
```

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

数据结构

最近更新时间：2026-04-29 01:11:10

AgentAppMcpServerDTO

关联的mcp服务配置

被如下接口引用：CreateAgentAppMcpServers。

名称	类型	必选	描述
ID	String	是	mcp server id 示例值：mcp-xxxxxxxx
NeedAuth	Boolean	否	是否需要鉴权 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
AgentCredentialID	String	否	凭据代填的ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：agc-xxxxxxxx
SSEResourceIdentifier	String	否	应用为OAuth2认证时，sse模式请求mcp时的资源标识 注意：此字段可能返回 null，表示取不到有效值。 示例值：http://127.0.0.1/_llmsgw_/sse/aga-05b85106/mcp-2fc41d77
StreamableResourceIdentifier	String	否	应用为OAuth2认证时，streamable模式请求mcp时的资源标识 注意：此字段可能返回 null，表示取不到有效值。 示例值：http://127.0.0.1/_llmsgw_/mcp/aga-05b85106/mcp-2fc41d77

AgentAppMcpServerVO

应用关联的mcp响应

被如下接口引用：DescribeAgentAppMcpServers。

名称	类型	描述
ID	String	绑定ID 示例值：aga-41d42939
NeedAuth	Boolean	需要认证 示例值：true
AgentCredentialID	String	凭据ID 示例值：agc-4440aa1d
AgentCredentialVO	DescribeAgentCredentialResp	凭据详情

名称	类型	描述
McpServerVO	DescribeMcpServerResponseVO	mcp详情
RelateTime	String	关联时间 示例值：2025-12-28T00:55:38.218Z
SSEResourceIdentifier	String	应用为OAuth2认证时，sse模式请求mcp时的资源标识 注意：此字段可能返回 null，表示取不到有效值。 示例值： http://10.0.0.4/_llmsgw_/sse/aga-41d42939/mcp-2fc41d77
StreamableResourceIdentifier	String	应用为OAuth2认证时，streamable模式请求mcp时的资源标识 注意：此字段可能返回 null，表示取不到有效值。 示例值： http://10.0.0.4/_llmsgw_/aga-41d42939/mcp-2fc41d77
AgentAppID	String	应用ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：aga-41d42939
McpServerID	String	mcp ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：mcp-41d42939

AgentAppModelServiceDTO

应用绑定模型服务入参

被如下接口引用：CreateAgentAppModelServices, ModifyAgentAppModelServices。

名称	类型	必选	描述
ID	String	否	模型服务ID 示例值：mds-36d0b658
InvokeLimitConfigStatus	Boolean	否	是否开启流量控制 示例值：true
InvokeLimitConfig	InvokeLimitConfigDTO	否	流量控制
TokenLimitStatus	Boolean	否	是否开启token控制 示例值：true

名称	类型	必选	描述
TokenLimitConfig	TokenLimitConfigDTO	否	token控制

AgentAppSecretKeyVO

secretKey的出参

被如下接口引用：DescribeAgentApp。

名称	类型	描述
SecretID	String	secret id 示例值：AK290f08b25DF12029ceb2e47519BFC50C
SecretKey	String	secret key 示例值：****

AgentCredentialContentDTO

凭证内容

被如下接口引用：DescribeAgentAppMcpServers。

名称	类型	必选	描述
STSSystem	String	否	如果认证类型为sts时，该项必填 示例值：tencent
STSService	String	否	如果认证类型为sts时，该项必填 示例值：cos
Headers	Array of AgentCredentialContentHeaderDTO	否	如果认证类型为reqKey时，该项必填

AgentCredentialContentHeaderDTO

凭证内容头

被如下接口引用：DescribeAgentAppMcpServers。

名称	类型	必选	描述
Key	String	否	凭据header key 示例值：apiKey
Value	String	否	凭据header value 示例值：**

BindMcpSecurityRuleDTO

BindMcpSecurityRuleDTO，替换之前的McpSecurityRule

被如下接口引用：CreateMcpServer, ModifyMcpServer。

名称	类型	必选	描述
ID	String	否	规则ID 示例值：mr-xxxxxxxx
Act	String	否	处置动作 示例值：watch

BindMcpSecurityRuleVO

BindMcpSecurityRuleVO，重新定义了之前的McpSecurityRulesVO

被如下接口引用：DescribeMcpServer。

名称	类型	必选	描述
ID	String	否	规则ID 示例值：msr-0101
Type	String	否	规则类型 示例值：data
Name	String	否	规则名称 示例值：data
Description	String	否	规则描述 示例值：data
RiskLevel	String	否	风险等级 示例值：low
VersionNumber	String	否	版本号 示例值：2026010101
Act	String	否	当前选择的处置动作 示例值：watch
SupportActs	Array of String	否	支持的处置动作 示例值：["watch"]
BodyType	String	否	内容类型 示例值：data
IconType	String	否	icon类型 示例值：data

CreateAgentAppResp

创建Agent应用的返回值，根据创建的AuthType，返回ApiKey或者SecretKey

被如下接口引用：CreateAgentApp。

名称	类型	描述
ID	String	app id 示例值：aga-xxxxxxxx
ApiKey	String	如果authType为apiKey时，返回该字段 示例值：sk-01010100101010
SecretKey	String	如果authType为secretKey时，返回该字段 示例值：AK01010100101010
SecretID	String	如果authType为secretKey时，返回该字段 示例值：010101001010100101010

DescribeAgentAppMcpServersResp

查询App mcpServer绑定列表响应

被如下接口引用：DescribeAgentAppMcpServers。

名称	类型	描述
Total	Integer	条目总数 示例值：4
Items	Array of AgentAppMcpServerVO	具体条目

DescribeAgentAppResp

查询app详情响应

被如下接口引用：DescribeAgentApp, DescribeAgentApps。

名称	类型	描述
AppID	Integer	租户appID 示例值：1300273807
Uin	String	租户ID 示例值：700001136234
InstanceID	String	实例ID 示例值：ins-a7af1980
ID	String	应用ID 示例值：aga-41d42939
Name	String	名称 示例值：测试agentApp

名称	类型	描述
Description	String	描述 示例值: description 01
Status	String	状态 示例值: normal
CreateTime	String	创建时间 示例值: 2025-12-28T00:55:30.53Z
UpdateTime	String	修改时间 示例值: 2025-12-28T00:55:30.53Z
AuthType	String	认证类型 示例值: apiKey
ApiKeys	Array of String	apiKey列表, 脱敏 示例值: ["SK**DF"]
SecretKeys	Array of AgentAppSecretKeyVO	secretKey列表, 脱敏
OAuth2ResourceServerID	String	OAuth2 Resource Server ID 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ors-be1312ba
McpServersNum	Integer	绑定mcpServer数量 示例值: 1
ModelServicesNum	Integer	绑定的模型服务数量 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0

DescribeAgentAppsResp

查询App列表响应

被如下接口引用: DescribeAgentApps。

名称	类型	描述
Total	Integer	条目总数 示例值: 10
Items	Array of DescribeAgentAppResp	具体条目

DescribeAgentAppsSortDTO

agentApp查询排序

被如下接口引用：DescribeAgentApps。

名称	类型	必选	描述
CreateTime	Integer	否	创建时间 示例值：1
UpdateTime	Integer	否	修改时间 示例值：1
Name	Integer	否	名称 示例值：1
Status	Integer	否	状态 示例值：1

DescribeAgentCredentialResp

凭据详情响应

被如下接口引用：DescribeAgentAppMcpServers。

名称	类型	描述
AppID	Integer	租户应用ID 示例值：1300273807
Uin	String	租户ID 示例值：700001136234
InstanceID	String	实例ID 示例值：ins-a7af1980
ID	String	凭据ID 示例值：agc-4440aa1d
Name	String	凭据名称 示例值：pingju
Status	String	状态 示例值：normal
RelateAgentAppNum	Integer	关联应用数 示例值：0
RelateMcpServerNum	Integer	关联mcp数 示例值：0
RelateModelNum	Integer	关联模型数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Content	AgentCredentialContentDTO	凭据内容

名称	类型	描述
CreateTime	String	创建时间 示例值: 2025-08-07T10:06:37.246Z
LastUpdateTime	String	修改时间 示例值: 2025-08-07T10:06:37.246Z
Type	String	类型 示例值: reqKey

DescribeMcpServerResponseVO

DescribeMcpServerResponseVO

被如下接口引用: DescribeAgentAppMcpServers, DescribeMcpServer, DescribeMcpServers。

名称	类型	描述
InstanceId	String	实例ID 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ins-a7af1980
Name	String	名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: testmcp
Description	String	描述 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: testmcp
LabelIds	Array of String	标签ID数组 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ["01010"]
CategoryIds	Array of String	目录ID数组 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ["01010"]
TargetSelect	String	负载方式, robin random consistentHash 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: random
TargetHosts	Array of TargetHostDTO	目标服务器 注意: 此字段可能返回 null, 表示取不到有效值。
HttpProtocolType	String	后端协议: http https 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: http
CheckTargetCertsError	Boolean	证书检查 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: false

名称	类型	描述
TargetPath	String	目标路径 注意：此字段可能返回 null，表示取不到有效值。 示例值：/
InvokeLimitConfigStatus	Boolean	流量控制状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
InvokeLimitConfig	InvokeLimitConfigDTO	流量控制配置 注意：此字段可能返回 null，表示取不到有效值。
IpWhiteStatus	Boolean	IP白名单开启状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：false
IpWhiteConfig	IpConfig	IP白名单配置 注意：此字段可能返回 null，表示取不到有效值。
IpBlackStatus	Boolean	IP黑名单开启状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
IpBlackConfig	IpConfig	IP黑名单配置 注意：此字段可能返回 null，表示取不到有效值。
ID	String	mcp server ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：mcp-2fc41d77
Status	String	状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：normal
Url	String	预览地址 注意：此字段可能返回 null，表示取不到有效值。 示例值：http://1.1.1.1/
App	IDNameVO	应用 注意：此字段可能返回 null，表示取不到有效值。
Catalogs	Array of IDNameVO	目录 注意：此字段可能返回 null，表示取不到有效值。
Labels	Array of IDNameVO	标签 注意：此字段可能返回 null，表示取不到有效值。
CreateTime	String	创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2025-08-07T09:55:58.696Z
LastUpdateTime	String	最后修改时间 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
		示例值: 2025-12-04T06:07:53.241Z
AppID	Integer	用户appID 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1300273807
Uin	String	用户ID 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 700001136234
CustomHttpHost	String	自定义host 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: default
HttpHostType	String	Http 请求host类型 useRequestHost 保持源请求 host targetHost 修正为源站host customHost 自定义host 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: useRequestHost
Timeout	Integer	请求的超时时间 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 60
Mode	String	mcp server模式 示例值: wrap
McpVersion	String	mcp version 示例值: 2024-11-05
WrapServices	Array of String	封装模式下绑定的服务ID列表 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ["svr-7e7f07e8"]
ToolNum	Integer	工具数量 示例值: 0
McpSecurityRulesVO	Array of McpSecurityRulesVO	安全规则集响应
ToolConfigs	Array of ToolConfigVO	真实工具级别配置, 实时拉取了tool/list做渲染的, 如果 tool/list不通, 就拉不到。 注意: 此字段可能返回 null, 表示取不到有效值。
UrlObj	McpUrlObj	访问URL
ToolMessage	String	后端mcp服务是否正常 示例值: error
Tools	Array of McpTool	后端mcp服务的工具列表
WrapPaasID	String	封装的API分组ID 示例值: app-be27914b

名称	类型	描述
RelateAgentAppNum	Integer	关联的agentApp数量 示例值: 0
PluginConfigs	Array of PluginConfigDTO	插件配置 注意: 此字段可能返回 null, 表示取不到有效值。

DescribeMcpServersResponseVO

ServicesVO

被如下接口引用: DescribeMcpServers。

名称	类型	描述
Total	Integer	条目 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 10
Items	Array of DescribeMcpServerResponseVO	数组 注意: 此字段可能返回 null, 表示取不到有效值。

DescribeModelResponseVO

查询模型详情的响应

被如下接口引用: DescribeModel, DescribeModels。

名称	类型	描述
AppID	Integer	腾讯云AppID 示例值: 1300273807
Uin	String	腾讯云Uin 示例值: 700001136234
InstanceID	String	实例ID 示例值: ins-a7af1980
ID	String	模型ID 示例值: mod-85b343e0
Name	String	模型名称 示例值: {55.52493325438*55.52493325438}
CredentialID	String	凭据ID 示例值: agc-4440aa1d
CredentialName	String	凭据名称 示例值: pingju

名称	类型	描述
HttpProtocolType	String	http协议类型 示例值: http
CheckTargetCertsError	Boolean	https时, 是否校验目标证书 示例值: true
HttpProtocolVersion	String	http协议版本: 1.1/2.0 示例值: 1.1
TargetPath	String	目标路径 示例值: /v122222
TargetHosts	Array of TargetHostDTO	目标器列表
ModelServiceCount	Integer	被模型服务使用的个数 示例值: 1
CreateTime	String	创建时间 示例值: 2026-02-10T18:09:29.011Z
LastUpdateTime	String	最后修改时间 示例值: 2026-02-10T18:09:32.104Z

DescribeModelServiceResponseVO

查询模型服务详情的响应

被如下接口引用: DescribeModelService, DescribeModelServices。

名称	类型	描述
AppID	Integer	腾讯云AppID 示例值: 1300273807
Uin	String	腾讯云Uin 示例值: 70000****
InstanceID	String	实例ID 示例值: ins-****
ID	String	模型ID 示例值: mds-****
Name	String	模型名称

名称	类型	描述
		示例值: heheh
Description	String	描述 示例值: heheh
PubPath	String	访问路径 示例值: /modeltest
PathMatchType	String	路径匹配方式: absolute, prefix, regex 示例值: prefix
TargetModels	Array of TargetModelDTO	目标模型列表 注意: 此字段可能返回 null, 表示取不到有效值。
ModelNames	Array of String	模板模型的名称列表 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ["deepseek"]
InvokeLimitConfigStatus	Boolean	是否开启限流 示例值: true
InvokeLimitConfig	InvokeLimitConfigDTO	限流配置
CreateTime	String	创建时间 示例值: 2026-02-09T07:18:30.357Z
LastUpdateTime	String	最后修改时间 示例值: 2026-02-09T07:18:30.357Z
TokenLimitStatus	Boolean	是否开启token控制 示例值: true
TokenLimitConfig	TokenLimitConfigDTO	token控制
TmsStatus	Boolean	是否开启tms配置

名称	类型	描述
		示例值: true
TmsConfig	TmsConfigDTO	tms配置
IpWhiteStatus	Boolean	是否开启IP白名单 示例值: true
IpWhiteList	Array of String	IP白名单列表 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ["1.1.1.1"]
IpBlackStatus	Boolean	是否开启IP黑名单 示例值: true
IpBlackList	Array of String	IP黑名单列表 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ["1.1.1.1"]
PluginConfigs	Array of PluginConfigDTO	插件配置 注意: 此字段可能返回 null, 表示取不到有效值。
Timeout	Integer	超时配置, 单位秒 示例值: 60
Status	String	状态: normal, disabled 示例值: disabled
RelateAgentAppNum	Integer	关联应用数 示例值: 1
Url	String	请求路径 示例值: http://10.0.0.6/modeltest
PromptModerateStatus	Boolean	是否开启提示词安全检测

名称	类型	描述
		注意：此字段可能返回 null，表示取不到有效值。 示例值：true
PromptModerateConfig	PromptModerateConfigDTO	提示词安全检测配置 注意：此字段可能返回 null，表示取不到有效值。
SensitiveDataCheckStatus	Boolean	是否开启敏感数据检测 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
SensitiveDataCheckConfig	SensitiveDataCheckConfigDTO	敏感数据检测配置 注意：此字段可能返回 null，表示取不到有效值。
TargetSelect	String	负载方式 枚举值： <ul style="list-style-type: none"> • random：随机 • consistentHash：会话保持 注意：此字段可能返回 null，表示取不到有效值。 示例值：consistentHash
FindHostKeyMethod	String	会话判断方式 枚举值： <ul style="list-style-type: none"> • fromClientIP：从客户端IP判断 • fromHeader：从请求header判断 • autoDetect：自动探测 注意：此字段可能返回 null，表示取不到有效值。 示例值：autoDetect
HostKeyHeaderName	String	会话判断header名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：x-session-id

名称	类型	描述
FallbackStatus	Boolean	是否开启备份模型 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
FallbackModels	Array of TargetModelDTO	备份模型 注意：此字段可能返回 null，表示取不到有效值。

DescribeModelServicesResponseVO

查询模型列表的响应

被如下接口引用：DescribeModelServices。

名称	类型	描述
Total	Integer	条目 示例值：2
Items	Array of DescribeModelServiceResponseVO	列表

DescribeModelServicesSort

查询模型服务列表的排序

被如下接口引用：DescribeModelServices。

名称	类型	必选	描述
CreateTime	Integer	否	创建时间 示例值：1
LastUpdateTime	Integer	否	最后修改时间 示例值：1
Name	Integer	否	模型名称 示例值：1
Status	Integer	否	状态 示例值：-1

DescribeModelsResponseVO

查询模型列表的响应

被如下接口引用: DescribeModels。

名称	类型	描述
Total	Integer	条目 示例值: 301
Items	Array of DescribeModelResponseVO	列表

DescribeModelsSort

DescribeModelsSort

被如下接口引用: DescribeModels。

名称	类型	必选	描述
CreateTime	Integer	否	创建时间 示例值: 1
LastUpdateTime	Integer	否	最后修改时间 示例值: 1
Name	Integer	否	模型名称 示例值: -1

IDNameVO

IDNameVO

被如下接口引用: DescribeMcpServer。

名称	类型	必选	描述
ID	String	否	业务ID 注意: 此字段可能返回 null, 表示取不到有效值。
Name	String	否	名称 注意: 此字段可能返回 null, 表示取不到有效值。

InvokeLimitConfigDTO

InvokeLimitConfigDTO

被如下接口引用: CreateAgentAppModelServices, CreateMcpServer, CreateModelService, DescribeMcpServer, DescribeModelService, ModifyAgentAppModelServices, ModifyMcpServer, ModifyModelService。

名称	类型	必选	描述
Type	String	否	Type类型 注意：此字段可能返回 null，表示取不到有效值。
TokenBucketMaxNum	Integer	否	令牌桶最大容量 注意：此字段可能返回 null，表示取不到有效值。
TokenBucketRate	Integer	否	令牌生成速率 注意：此字段可能返回 null，表示取不到有效值。
FunnelMaxNum	Integer	否	漏斗容量 注意：此字段可能返回 null，表示取不到有效值。
FunnelRate	Integer	否	漏嘴流速 注意：此字段可能返回 null，表示取不到有效值。
SlidingWindowMaxNum	Integer	否	滑动窗口最大请求数 注意：此字段可能返回 null，表示取不到有效值。
SlidingWindowSize	Integer	否	滑动窗口长度 注意：此字段可能返回 null，表示取不到有效值。
TimeWindow	Integer	否	时间窗口最大请求数 注意：此字段可能返回 null，表示取不到有效值。
TimeWindowInterval	Integer	否	时间窗口长度 注意：此字段可能返回 null，表示取不到有效值。
Timeout	Integer	否	请求的超时时间 注意：此字段可能返回 null，表示取不到有效值。

IpConfig

ip黑白名单配置

被如下接口引用：CreateMcpServer, DescribeMcpServer, ModifyMcpServer。

名称	类型	必选	描述
Ips	Array of String	否	ip数组 注意：此字段可能返回 null，表示取不到有效值。 示例值：["1.1.1.1"]
EffectType	String	否	生效类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：always
EffectTimes	Array of StartEndTime	否	生效时间 注意：此字段可能返回 null，表示取不到有效值。
Comment	String	否	备注 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	必选	描述
			示例值: test_e

LimitWindowsDTO

限流窗口配置

被如下接口引用: CreateModelService, ModifyModelService。

名称	类型	必选	描述
Interval	Integer	否	时间窗口, 分钟 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 10
Limit	Integer	否	累计上限, k 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 100

McpInputOutSchema

出入参说明

被如下接口引用: DescribeMcpServer。

名称	类型	必选	描述
Type	String	否	类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: string
Properties	String	否	属性 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: test_ee
Required	Array of String	否	必填字段 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ["string"]
Description	String	否	描述 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: test_ee

McpSecurityRule

mcp安全规则

被如下接口引用: CreateMcpServer, ModifyMcpServer。

名称	类型	必选	描述
ID	String	否	规则ID 示例值: mr-xxxxxxxx
Status	String	否	状态 示例值: open
Act	String	否	处置动作 示例值: watch

McpSecurityRulesVO

绑定的安全规则列表

被如下接口引用: DescribeMcpServer。

名称	类型	描述
ID	String	规则ID 示例值: msr-0101
Type	String	规则类型 示例值: data
Name	String	规则名称 示例值: data
Description	String	规则描述 示例值: data
RiskLevel	String	风险等级 示例值: low
VersionNumber	String	版本号 示例值: 2026010101
Status	String	状态开关 示例值: normal
Act	String	当前选择的处置动作 示例值: watch
SupportActs	Array of String	支持的处置动作 示例值: ["watch"]
BodyType	String	内容类型 示例值: data
IconType	String	icon类型 示例值: data

McpTool

mcp工具

被如下接口引用：DescribeMcpServer。

名称	类型	必选	描述
Name	String	否	名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：getUser
Description	String	否	描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：getUser
InputSchema	McpInputOutSchema	否	入参参数 注意：此字段可能返回 null，表示取不到有效值。
Annotations	McpToolAnnotation	否	注释 注意：此字段可能返回 null，表示取不到有效值。
OutputSchema	McpInputOutSchema	否	出参参数 注意：此字段可能返回 null，表示取不到有效值。

McpToolAnnotation

mcp工具注解

被如下接口引用：DescribeMcpServer。

名称	类型	必选	描述
Title	String	否	title for the tool 注意：此字段可能返回 null，表示取不到有效值。 示例值：getUser
ReadOnlyHint	Boolean	否	If true, the tool does not modify its environment 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
DestructiveHint	Boolean	否	If true, the tool may perform destructive updates 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
IdempotentHint	Boolean	否	If true, repeated calls with same args have no additional effect 注意：此字段可能返回 null，表示取不到有效值。 示例值：true
OpenWorldHint	Boolean	否	If true, tool interacts with external entities 注意：此字段可能返回 null，表示取不到有效值。 示例值：false

McpUrlObj

Mcp的访问URL

被如下接口引用：DescribeMcpServer。

名称	类型	描述
SSEUrl	String	sse访问URL 示例值：http://10.0.0.4/_llmsgw_/sse/aga-41d42939/mcp-2fc41d77
StreamAbleUrl	String	streamable访问URL 示例值：http://10.0.0.4/_llmsgw_/mcp/aga-2f812567/mcp-2fc41d77

PluginConfigDTO

PluginConfigDTO

被如下接口引用：CreateMcpServer, CreateModelService, DescribeMcpServer, DescribeModelService, ModifyMcpServer, ModifyModelService。

名称	类型	必选	描述
Status	Boolean	否	状态 注意：此字段可能返回 null，表示取不到有效值。
PluginName	String	否	名称 注意：此字段可能返回 null，表示取不到有效值。
Description	String	否	描述 注意：此字段可能返回 null，表示取不到有效值。
PluginID	String	否	ID 注意：此字段可能返回 null，表示取不到有效值。
Level	Integer	否	优先级 注意：此字段可能返回 null，表示取不到有效值。
PluginFormValues	Array of PluginFormValueDTO	否	表单配置 注意：此字段可能返回 null，表示取不到有效值。

PluginFormValueDTO

PluginFormValueDTO

被如下接口引用：CreateMcpServer, CreateModelService, ModifyMcpServer, ModifyModelService。

名称	类型	必选	描述
Field	String	否	字段 注意：此字段可能返回 null，表示取不到有效值。
Value	String	否	值 注意：此字段可能返回 null，表示取不到有效值。

PromptModerateConfigDTO

提示词安全配置

被如下接口引用：CreateModelService, DescribeModelService, ModifyModelService。

名称	类型	必选	描述
Action	String	否	执行动作 注意：此字段可能返回 null，表示取不到有效值。 示例值：watch
InterceptMessage	String	否	响应拦截内容 注意：此字段可能返回 null，表示取不到有效值。 示例值：intercept

ResultIDVO

ResultIDVO

被如下接口引用：CreateAgentAppMcpServers, CreateAgentAppModelServices, CreateMcpServer, CreateModel, CreateModelService, DeleteAgentApp, DeleteAgentAppMcpServers, DeleteAgentAppModelServices, DeleteMcpServer, DeleteModelService, ModifyAgentApp, ModifyAgentAppModelServices, ModifyMcpServer, ModifyModel, ModifyModelService。

名称	类型	描述
ID	String	对象ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：mds-36d0b658

ResultIDsVO

结果ID数组

被如下接口引用：DeleteModel。

名称	类型	描述
IDs	Array of String	结果ID数组 注意：此字段可能返回 null，表示取不到有效值。 示例值：["mod-edd55e4f"]

SensitiveDataCheckConfigDTO

敏感数据检测配置

被如下接口引用：CreateModelService, DescribeModelService, ModifyModelService。

名称	类型	必选	描述
Action	String	否	执行动作 注意：此字段可能返回 null，表示取不到有效值。 示例值：intercept
InterceptMessage	String	否	响应拦截内容 注意：此字段可能返回 null，表示取不到有效值。 示例值：intercept
CheckItems	Array of String	否	检测项 枚举值： <ul style="list-style-type: none"> • birthday: 生日 • email: 邮箱 • identity_number: 身份证 • phone_number: 电话号码 • secret: 密钥 • password: 密码 • private_key: 私钥 注意：此字段可能返回 null，表示取不到有效值。 示例值：["password"]

StartEndTime

开始结束时间结构体

被如下接口引用：CreateMcpServer, DescribeMcpServer, ModifyMcpServer。

名称	类型	必选	描述
StartTime	String	否	开始时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：202512121212
EndTime	String	否	结束时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：202601010101

TargetHostDTO

TargetHostDTO

被如下接口引用：CreateMcpServer, CreateModel, DescribeMcpServer, DescribeModel, ModifyMcpServer, ModifyModel。

名称	类型	必选	描述
Host	String	否	服务器 注意：此字段可能返回 null，表示取不到有效值。 示例值：10.1.2.3
Rank	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：10

TargetModelDTO

模板模型

被如下接口引用：CreateModelService, DescribeModelService, ModifyModelService。

名称	类型	必选	描述
ID	String	否	模型ID 示例值：mod-0926d298
Name	String	否	匹配名称 示例值：deepseek
Rank	Integer	否	权重 示例值：10

TmsConfigDTO

内容安全配置

被如下接口引用：CreateModelService, DescribeModelService, ModifyModelService。

名称	类型	必选	描述
Scope	Array of String	否	检测范围,请求/响应 注意：此字段可能返回 null，表示取不到有效值。 示例值：["request"]
Mode	String	否	检测形式 注意：此字段可能返回 null，表示取不到有效值。 示例值：sync
Action	String	否	执行动作 注意：此字段可能返回 null，表示取不到有效值。 示例值：watch
MergeCount	Integer	否	合并请求检测event数，联动Mode字段sync 注意：此字段可能返回 null，表示取不到有效值。 示例值：10

名称	类型	必选	描述
BizType	String	否	风控策略 注意：此字段可能返回 null，表示取不到有效值。 示例值：tms-01010
InterceptMessage	String	否	响应拦截内容 注意：此字段可能返回 null，表示取不到有效值。 示例值：failed

TokenLimitConfigDTO

Token限流配置

被如下接口引用：CreateAgentAppModelServices, CreateModelService, DescribeModelService, ModifyAgentAppModelServices, ModifyModelService。

名称	类型	必选	描述
LimitRequestBody	Integer	否	单次请求上限，k 注意：此字段可能返回 null，表示取不到有效值。 示例值：10000
LimitWindows	Array of LimitWindowsDTO	否	累次token总量消耗上限 注意：此字段可能返回 null，表示取不到有效值。

ToolConfigDTO

工具级别配置

被如下接口引用：CreateMcpServer, ModifyMcpServer。

名称	类型	必选	描述
ToolName	String	否	工具名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：tool-test
InvokeLimitConfigStatus	Boolean	否	是否开启限流配置 示例值：true
InvokeLimitConfig	InvokeLimitConfigDTO	否	限流配置
McpSecurityRules	Array of BindMcpSecurityRuleDTO	否	绑定安全规则

ToolConfigVO

工具配置响应

被如下接口引用：DescribeMcpServer。

名称	类型	描述
ToolName	String	工具名称 示例值：toolName
InvokeLimitConfigStatus	Boolean	是否开启限流配置 示例值：false
InvokeLimitConfig	InvokeLimitConfigDTO	限流配置
McpSecurityRules	Array of BindMcpSecurityRuleVO	绑定安全规则（响应）

错误码

最近更新时间：2026-03-23 17:44:05

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。

错误码	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。
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。

错误码	说明
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。