

应用型负载均衡

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

转发规则相关接口

修改转发规则属性

查询转发规则

删除转发规则

创建转发规则

目标组相关接口

从目标组内移除后端服务

修改目标组内后端服务信息

修改目标组

根据后端服务查询绑定的目标组

查询目标组列表

查询目标组内后端服务

删除目标组

创建目标组

向目标组内添加后端服务

监听器相关接口

修改监听器

监听器解绑扩展证书

查询监听器列表

查询监听器健康状态

查询监听器详细信息

查询监听器绑定证书列表

删除监听器

创建监听器

监听器绑定扩展证书

实例相关接口

更新负载均衡实例属性

修改负载均衡实例的网络类型

查询负载均衡列表

查询负载均衡实例详细信息

删除应用型负载均衡实例

创建应用型负载均衡实例

安全策略相关接口

修改安全策略

查询系统安全策略

查询安全策略关联关系

获取支持的安全策略加密套件设置

查询自定义安全策略

删除安全策略

创建自定义安全策略

其他相关接口

设置负载均衡实例的安全组

通知负载均衡解绑后端服务

设置负载均衡实例操作保护

创建负载均衡实例询价

查询可用区信息

查询配额

查询异步任务信息

共享带宽包相关接口

实例解绑共享带宽包

实例绑定共享带宽包

健康检查模板相关接口

修改健康检查模板

查询健康检查模板列表

删除健康检查模板

创建健康检查模板

数据结构

错误码

API 文档

更新历史

最近更新时间：2026-07-15 01:09:06

第 3 次发布

发布时间：2026-07-15 01:08:49

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ModifyHealthCheckTemplate](#)
 - 删除入参：Tags

第 2 次发布

发布时间：2026-07-09 01:09:48

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [AddTargetsToTargetGroup](#)
 - 修改入参：Targets
- [CreateRules](#)
 - 修改入参：Rules
- [DeleteHealthCheckTemplates](#)
 - 修改入参：HealthCheckTemplateIds
- [ModifyLoadBalancerModificationProtection](#)
 - 修改入参：ModificationProtectionEnabled
- [ModifyTargetsInTargetGroup](#)
 - 修改入参：Targets
- [RemoveTargetsFromTargetGroup](#)
 - 修改入参：Targets

第 1 次发布

发布时间：2026-07-07 10:57:21

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AddTargetsToTargetGroup](#)

- [AssociateBandwidthPackageWithLoadBalancer](#)
- [AssociateListenerAdditionalCertificates](#)
- [CreateHealthCheckTemplate](#)
- [CreateListener](#)
- [CreateLoadBalancer](#)
- [CreateRules](#)
- [CreateSecurityPolicy](#)
- [CreateTargetGroup](#)
- [DeleteHealthCheckTemplates](#)
- [DeleteListener](#)
- [DeleteLoadBalancers](#)
- [DeleteRules](#)
- [DeleteSecurityPolicy](#)
- [DeleteTargetGroups](#)
- [DescribeAsyncJobs](#)
- [DescribeHealthCheckTemplates](#)
- [DescribeListenerCertificates](#)
- [DescribeListenerDetail](#)
- [DescribeListenerHealthStatus](#)
- [DescribeListeners](#)
- [DescribeLoadBalancerDetail](#)
- [DescribeLoadBalancers](#)
- [DescribeQuota](#)
- [DescribeRules](#)
- [DescribeSecurityPolicies](#)
- [DescribeSecurityPolicyCapabilities](#)
- [DescribeSecurityPolicyRelations](#)
- [DescribeSystemSecurityPolicies](#)
- [DescribeTargetGroupTargets](#)
- [DescribeTargetGroups](#)
- [DescribeTargetGroupsByTarget](#)
- [DescribeZones](#)
- [DisassociateBandwidthPackageFromLoadBalancer](#)
- [DisassociateListenerAdditionalCertificates](#)
- [InquirePriceCreateLoadBalancer](#)
- [ModifyHealthCheckTemplate](#)
- [ModifyListenerAttributes](#)
- [ModifyLoadBalancerAddressType](#)
- [ModifyLoadBalancerAttributes](#)
- [ModifyLoadBalancerModificationProtection](#)
- [ModifyRulesAttributes](#)
- [ModifySecurityPolicyAttributes](#)
- [ModifyTargetGroupAttributes](#)
- [ModifyTargetsInTargetGroup](#)
- [NotifyUnbindTarget](#)

- [RemoveTargetsFromTargetGroup](#)
- [SetLoadBalancerSecurityGroups](#)

新增数据结构：

- [AccessLogConfig](#)
- [CertificateInfo](#)
- [DefaultAction](#)
- [DeletionProtectionConfig](#)
- [Filter](#)
- [FixedResponseInfo](#)
- [HTTPCookieInfo](#)
- [HTTPHeaderInfo](#)
- [HTTPQueryStringInfo](#)
- [HTTPRedirectInfo](#)
- [HTTPRewriteInfo](#)
- [HealthCheckConfig](#)
- [HealthCheckTemplate](#)
- [IPAddressInfo](#)
- [InsertHTTPHeaderInfo](#)
- [Job](#)
- [ListenerOutput](#)
- [LoadBalancer](#)
- [LoadBalancerAddress](#)
- [LoadBalancerBillingConfig](#)
- [LoadBalancerDetail](#)
- [LoadBalancerOperationLocksItem](#)
- [ModificationProtectionInfo](#)
- [PostPayPriceInfo](#)
- [Price](#)
- [QuotaInfo](#)
- [RelatedListener](#)
- [RemoveHTTPHeaderInfo](#)
- [RuleAction](#)
- [RuleCondition](#)
- [RuleHealthStatusInfo](#)
- [RuleInput](#)
- [RuleModify](#)
- [RuleOutput](#)
- [SecurityPolicyCapability](#)
- [SecurityPolicyInfo](#)
- [SecurityPolicyRelations](#)
- [StickySessionConfig](#)
- [TagInfo](#)
- [TargetGroupConfig](#)
- [TargetGroupHealthInfo](#)

- [TargetGroupOutput](#)
- [TargetGroupStickySession](#)
- [TargetGroupTuple](#)
- [TargetHealthStatusInfo](#)
- [TargetOutput](#)
- [TargetToAdd](#)
- [TargetToModify](#)
- [TargetToRemove](#)
- [XForwardedForConfig](#)
- [Zone](#)
- [ZoneMappingInfo](#)
- [ZoneMappingsItem](#)

简介

最近更新时间：2026-07-09 01:10:17

概述

应用型负载均衡（Application Load Balancer，ALB）是腾讯云提供的专注于 HTTP、HTTPS、QUIC 等应用层协议的负载均衡服务，具备强大的应用层流量处理与弹性伸缩能力。ALB 支持基于请求内容的高级路由，能够将来自客户端的访问请求按照配置的转发策略分发到后端服务器，并与容器、微服务等云原生服务深度集成，可作为云原生 Ingress 网关使用，适用于音视频、直播、游戏、微服务通信及大模型 AI 等业务场景。

您可以通过 [API Explorer](#) 在线调试本产品接口，也可以参考 [API 概览](#) 了解全部接口能力。ALB 支持通过控制台、SDK、Terraform 等多种方式进行创建与管理。

术语表

应用型负载均衡 API 接口的常见术语请参见下表：

术语	说明
实例 (LoadBalancer)	面向七层（应用层）提供负载均衡能力的资源实体，是承载监听、转发规则和后端服务的基础单元。
监听器 (Listener)	ALB 的最小业务单元，需配置协议（HTTP/HTTPS/QUIC）与端口，用于接收并处理客户端请求。
转发规则 (Rule)	定义请求的路由方式，支持基于域名、路径（URL）、HTTP 请求头、Cookie、请求方法、来源 IP 等条件进行匹配转发，并支持重定向、重写等高级操作。
目标组 (TargetGroup)	后端服务器的逻辑分组，包含一组处理实际请求的后端服务，支持 CVM、容器、弹性网卡等多种类型。
健康检查 (HealthCheck)	周期性探测后端服务器的可用性，自动隔离异常节点，保障业务连续性；支持自定义检查参数及健康检查模板。
VIP	流量分发的入口 IP，为 VPC 内的私网 IP 地址，是 ALB 对外提供服务的地址实体。
公网 / 私网 ALB	公网 ALB 通过弹性公网 IP（EIP）对外提供服务，私网 ALB 仅在 VPC 内部提供服务。
EIP（弹性公网 IP）	可独立申请、动态绑定到 ALB 实例的公网 IP 地址，仅公网 ALB 需要绑定；支持灵活解绑与迁移，实现公网访问能力。
域名	解析至 ALB 实例地址，公网与私网 ALB 均可使用；公网 ALB 解析到 EIP，私网 ALB 解析到 VIP； ALB 的域名不能直接访问 ，需通过 CNAME 将自有域名解析至 ALB 域名，实现业务无缝接入。
共享带宽包	提供地域级公网带宽的共享复用能力，多个 ALB 实例（或 EIP）可共用一个带宽包，按需分配带宽，有效节省公网带宽成本。

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和配额说明。其他限制如下：

限制项	说明
破坏性操作	删除（Delete）、销毁（Destroy）、隔离（Isolate）、解绑（Remove）等操作不可逆，请谨慎调用。
鉴权方式	租户侧接口通过 CAM 进行资源级鉴权，需保证调用账号对目标资源具备相应操作权限。

API 快速入门

以创建一个 ALB 实例为例，快速体验 ALB 的接口调用流程：

1. 进入 [API Explorer](#) 页面，选择「应用型负载均衡 ALB」产品。
2. 调用 [CreateLoadBalancer](#) 接口创建负载均衡实例，指定地域（Region）、网络类型（公网/私网）、VPC 等参数。
3. 调用 [CreateListener](#) 接口为实例创建监听器，配置协议与端口。
4. 调用 [CreateRule](#) 接口配置转发规则，绑定目标组以完成请求路由。
5. 前往 [应用型负载均衡控制台](#) 查看实例状态并进行后续配置管理。

API 概览

最近更新时间：2026-07-07 10:58:13

共享带宽包相关接口

接口名称	接口功能	频率限制（次/秒）
AssociateBandwidthPackageWithLoadBalancer	实例绑定共享带宽包	20
DisassociateBandwidthPackageFromLoadBalancer	实例解绑共享带宽包	20

监听器相关接口

接口名称	接口功能	频率限制（次/秒）
AssociateListenerAdditionalCertificates	监听器绑定扩展证书	20
CreateListener	创建监听器	20
DeleteListener	删除监听器	20
DescribeListenerCertificates	查询监听器绑定证书列表	20
DescribeListenerDetail	查询监听器详细信息	20
DescribeListenerHealthStatus	查询监听器健康状态	20
DescribeListeners	查询监听器列表	20
DisassociateListenerAdditionalCertificates	监听器解绑扩展证书	20
ModifyListenerAttributes	修改监听器	20

目标组相关接口

接口名称	接口功能	频率限制（次/秒）
AddTargetsToTargetGroup	向目标组内添加后端服务	20
CreateTargetGroup	创建目标组	20
DeleteTargetGroups	删除目标组	20
DescribeTargetGroupTargets	查询目标组内后端服务	20
DescribeTargetGroups	查询目标组列表	20
DescribeTargetGroupsByTarget	根据后端服务查询绑定的目标组	20
ModifyTargetGroupAttributes	修改目标组	20
ModifyTargetsInTargetGroup	修改目标组内后端服务信息	20

接口名称	接口功能	频率限制（次/秒）
RemoveTargetsFromTargetGroup	从目标组内移除后端服务	20

其他相关接口

接口名称	接口功能	频率限制（次/秒）
SetLoadBalancerSecurityGroups	设置负载均衡实例的安全组	20
DescribeAsyncJobs	查询异步任务信息	20
DescribeQuota	查询配额	20
DescribeZones	查询可用区信息	20
InquirePriceCreateLoadBalancer	创建负载均衡实例询价	20
ModifyLoadBalancerModificationProtection	设置负载均衡实例操作保护	20
NotifyUnbindTarget	通知负载均衡解绑后端服务	20

健康检查模板相关接口

接口名称	接口功能	频率限制（次/秒）
CreateHealthCheckTemplate	创建健康检查模板	20
DeleteHealthCheckTemplates	删除健康检查模板	20
DescribeHealthCheckTemplates	查询健康检查模板列表	20
ModifyHealthCheckTemplate	修改健康检查模板	20

安全策略相关接口

接口名称	接口功能	频率限制（次/秒）
CreateSecurityPolicy	创建自定义安全策略	20
DeleteSecurityPolicy	删除安全策略	20
DescribeSecurityPolicies	查询自定义安全策略	20
DescribeSecurityPolicyCapabilities	获取支持的安全策略加密套件设置	20
DescribeSecurityPolicyRelations	查询安全策略关联关系	20
DescribeSystemSecurityPolicies	查询系统安全策略	20
ModifySecurityPolicyAttributes	修改安全策略	20

实例相关接口

接口名称	接口功能	频率限制（次/秒）
CreateLoadBalancer	创建应用型负载均衡实例	20
DeleteLoadBalancers	删除应用型负载均衡实例	20
DescribeLoadBalancerDetail	查询负载均衡实例详细信息	20
DescribeLoadBalancers	查询负载均衡列表	20
ModifyLoadBalancerAddressType	修改负载均衡实例的网络类型	20
ModifyLoadBalancerAttributes	更新负载均衡实例属性	20

转发规则相关接口

接口名称	接口功能	频率限制（次/秒）
CreateRules	创建转发规则	20
DeleteRules	删除转发规则	20
DescribeRules	查询转发规则	20
ModifyRulesAttributes	修改转发规则属性	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：2026-07-07 10:58:11

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

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）	alb.ap-shanghai-fsi.tencentcloudapi.com
华南地区（深圳金融）	alb.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：2026-07-10 01:09:29

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。取值参考接口文档输入参数章节关于公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	HTTP 请求头：X-TC-Timestamp。当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。 注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
Version	String	是	HTTP 请求头：X-TC-Version。操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKID***/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKID*** 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为具体产品名，通常为域名前缀。例如，域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 alb；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=37ac2f4fde00b0ac9bd9eadeb459b1bbec224158d66e7ae5fcadb70b2d181d02&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-guangzhou&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
华南地区（广州）	ap-guangzhou
华东地区（上海）	ap-shanghai

签名方法 v3

最近更新时间：2026-07-07 10:58:12

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=10b1a37a7301a02ca19a647ad722d5e43b4b3cfff309d421d85b46093f6ab6c4f" \
-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。

字段名称	解释
CanonicalURI	URI 参数，API 3.0 固定为正斜杠 (/)。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串""，对于 GET 请求，则为 URL 中间号 (?) 后面的字符串内容，例如：Limit=10&Offset=0。 注意：CanonicalQueryString 需要参考 RFC3986 进行 URLEncode 编码（特殊字符编码后需大写字母），字符集 UTF-8。推荐使用编程语言标准库进行编码。
CanonicalHeaders	参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入其他头部参与签名以提高自身请求的唯一性和安全性，此示例额外增加了接口名头部。 拼接规则： 1. 头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接； 2. 多个头部，按照头部 key（小写）的 ASCII 升序进行拼接。 此示例计算结果是 content-type:application/json; charset=utf-8\nhost:cvm.tencentcloudapi.com\nx-tc-action:describeinstances\n。 注意：content-type 必须和实际发送的相符合，有些编程语言网络库即使未指定也会自动添加 charset 值，如果签名时和发送时不一致，服务器会返回签名校验失败。
SignedHeaders	参与签名的头部信息，说明此次请求有哪些头部参与了签名，和 CanonicalHeaders 包含的头部内容是一一对应的。content-type 和 host 为必选头部。 拼接规则： 1. 头部 key 统一转成小写； 2. 多个头部 key（小写）按照 ASCII 升序进行拼接，并且以分号 (;) 分隔。 此示例为 content-type;host;x-tc-action
HashedRequestPayload	请求正文（payload，即 body，此示例为 {"Limit": 1, "Filters": [{"Values": [{"\u672a\u547d\u540d"}], "Name": "instance-name"}]}）的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，即对 HTTP 请求正文做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。对于 GET 请求，RequestPayload 固定为空字符串。此示例计算结果是 35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064。

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

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

content-type;host;x-tc-action
35e9c5b0e3ae67532d3c9f17ead6c90222632e5b1ff7f6e89887f1398934f064
```

2. 拼接待签名字符串

按如下格式拼接待签名字符串：

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

字段名称	解释
Algorithm	签名算法，目前固定为 TC3-HMAC-SHA256。
RequestTimestamp	请求时间戳，即请求头部的公共参数 X-TC-Timestamp 取值，取当前时间 UNIX 时间戳，精确到秒。此示例取值为 1551113065。
CredentialScope	凭证范围，格式为 Date/service/tc3_request，包含日期、所请求的服务和终止字符串（tc3_request）。Date 为 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为产品名，必须与调用的产品域名一致。此示例计算结果是 2019-02-25/cvm/tc3_request。
HashedCanonicalRequest	前述步骤拼接所得规范请求串的哈希值，计算伪代码为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。此示例计算结果是 7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84。

注意：

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

根据以上规则，示例中得到的待签名字符串如下：

```
TC3-HMAC-SHA256  
1551113065  
2019-02-25/cvm/tc3_request  
7019a55be8395899b900fb5564e4200d984910f34794a27cb3fb7d10ff6a1e84
```

3. 计算签名

1) 计算派生签名密钥，伪代码如下：

```
SecretKey = "*****"  
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
```

```
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

派生出的密钥 SecretDate、SecretService 和 SecretSigning 是二进制的数，可能包含不可打印字符，将其转为十六进制字符串打印的输出分别为：da98fb70dcf6b112dc21038d1eeeb3a95c74b4dcb12c1131f864f6066bd02be0，8d70cbefb03939f929db64d32dc2ba89b1095620119fe3e050e2b18c5bd2752f，b596b923aad85185e2d1f6659d2a062e0a86731226e021e61bfe06f7ed05f5af。

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

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

2) 计算签名，伪代码如下：

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

此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

4. 拼接 Authorization

按如下格式拼接 Authorization：

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

字段名称	解释
Algorithm	签名方法，固定为 TC3-HMAC-SHA256。
SecretId	密钥对中的 SecretId，即 AKID*****。
CredentialScope	见上文，凭证范围。此示例计算结果是 2019-02-25/cvm/tc3_request。
SignedHeaders	见上文，参与签名的头部信息。此示例取值为 content-type;host;x-tc-action。
Signature	签名值。此示例计算结果是 10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f。

根据以上规则，示例中得到的值为：

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

最终完整的调用信息如下：

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

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

⚠ 注意：

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

签名演示

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

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

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

- [Signature Demo](#)

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

Java

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

public class TencentCloudAPITC3Demo {
    private final static Charset UTF8 = StandardCharsets.UTF_8;
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    private final static String SECRET_ID = System.getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    private final static String SECRET_KEY = System.getenv("TENCENTCLOUD_SECRET_KEY");
    private final static String CT_JSON = "application/json; charset=utf-8";

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

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

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.tencentcloudapi.com";
        String region = "ap-guangzhou";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1551113065";
        //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
    }
}
```

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

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

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

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

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

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

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Content-Type", CT_JSON);
headers.put("Host", host);
```

```
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

StringBuilder sb = new StringBuilder();
sb.append("curl -X POST https://").append(host)
.append(" -H \"Authorization: ").append(authorization).append("\")")
.append(" -H \"Content-Type: application/json; charset=utf-8\"")
.append(" -H \"Host: ").append(host).append("\")")
.append(" -H \"X-TC-Action: ").append(action).append("\")")
.append(" -H \"X-TC-Timestamp: ").append(timestamp).append("\")")
.append(" -H \"X-TC-Version: ").append(version).append("\")")
.append(" -H \"X-TC-Region: ").append(region).append("\")")
.append(" -d '").append(payload).append("'");
System.out.println(sb.toString());
}
}
```

Python

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

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

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

# ***** 步骤 1: 拼接规范请求串 *****
http_request_method = "POST"
```



```
canonical_uri = "/"
canonical_querystring = ""
ct = "application/json; charset=utf-8"
payload = json.dumps(params)
canonical_headers = "content-type:%s\nhost:%s\nx-tc-action:%s\n" % (ct, host, action.lower())
signed_headers = "content-type;host;x-tc-action"
hashed_request_payload = hashlib.sha256(payload.encode("utf-8")).hexdigest()
canonical_request = (http_request_method + "\n" +
canonical_uri + "\n" +
canonical_querystring + "\n" +
canonical_headers + "\n" +
signed_headers + "\n" +
hashed_request_payload)
print(canonical_request)

# ***** 步骤 2：拼接待签名字符串 *****
credential_scope = date + "/" + service + "/" + "tc3_request"
hashed_canonical_request = hashlib.sha256(canonical_request.encode("utf-8")).hexdigest()
string_to_sign = (algorithm + "\n" +
str(timestamp) + "\n" +
credential_scope + "\n" +
hashed_canonical_request)
print(string_to_sign)

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

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

print('curl -X POST ' + endpoint
+ ' -H "Authorization: ' + authorization + '"
+ ' -H "Content-Type: application/json; charset=utf-8"')
```

```
+ ' -H "Host: ' + host + '"'  
+ ' -H "X-TC-Action: ' + action + '"'  
+ ' -H "X-TC-Timestamp: ' + str(timestamp) + '"'  
+ ' -H "X-TC-Version: ' + version + '"'  
+ ' -H "X-TC-Region: ' + region + '"'  
+ " -d '" + payload + '"')
```

Golang

```
package main  
  
import (  
    "crypto/hmac"  
    "crypto/sha256"  
    "encoding/hex"  
    "fmt"  
    "os"  
    "strings"  
    "time"  
)  
  
func sha256hex(s string) string {  
    b := sha256.Sum256([]byte(s))  
    return hex.EncodeToString(b[:])  
}  
  
func hmacsha256(s, key string) string {  
    hashed := hmac.New(sha256.New, []byte(key))  
    hashed.Write([]byte(s))  
    return string(hashed.Sum(nil))  
}  
  
func main() {  
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****  
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")  
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****  
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")  
    host := "cvm.tencentcloudapi.com"  
    algorithm := "TC3-HMAC-SHA256"  
    service := "cvm"  
    version := "2017-03-12"  
    action := "DescribeInstances"  
    region := "ap-guangzhou"  
    //var timestamp int64 = time.Now().Unix()  
    var timestamp int64 = 1551113065
```

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

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

// step 3: sign string
secretDate := hmacsha256(date, "TC3"+secretKey)
secretService := hmacsha256(service, secretDate)
secretSigning := hmacsha256("tc3_request", secretService)
signature := hex.EncodeToString([]byte(hmacsha256(string2sign, secretSigning)))
fmt.Println(signature)

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

```
fmt.Println(authorization)

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

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

// step 1: build canonical request string
$httpRequestMethod = "POST";
$canonicalUri = "/";
$canonicalQueryString = "";
$canonicalHeaders = implode("\n", [
    "content-type:application/json; charset=utf-8",
    "host:". $host,
    "x-tc-action:". strtolower($action),
    ""
]);
$signedHeaders = implode(";", [
    "content-type",
    "host",
    "x-tc-action",
]);
$payload = '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-nam
```

```
e"}}}';
$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}\nx-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=#{signed_headers}, Signature=#{signature}"
puts authorization

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

DotNet

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

public class Application
```

```
{
public static string SHA256Hex(string s)
{
using (SHA256 algo = SHA256.Create())
{
byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));
StringBuilder builder = new StringBuilder();
for (int i = 0; i < hashbytes.Length; ++i)
{
builder.Append(hashbytes[i].ToString("x2"));
}
return builder.ToString();
}
}

public static byte[] HmacSHA256(byte[] key, byte[] msg)
{
using (HMACSHA256 mac = new HMACSHA256(key))
{
return mac.ComputeHash(msg);
}
}

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



```
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);

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

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

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

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

public static void Main(string[] args)
{
// 密钥参数
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
```

```
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

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

// 此处由于示例规范的原因, 采用时间戳2019-02-26 00:44:25, 此参数作为示例, 如果在项目中, 您应当使用:
// DateTime date = DateTime.UtcNow;
// 注意时区, 建议此时间统一采用UTC时间戳, 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 OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX hmac;
HMAC_CTX_init(&hmac);
h = &hmac;
#else
h = HMAC_CTX_new();
#endif

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

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

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

return (ss.str());
}

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

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

```
}

int main()
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    string SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    string SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");

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

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

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

```
dCanonicalRequest;
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", &stime);
}

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 OPENSSL_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, "\nX-tc-action:");
char value[100] = {0};
lowercase(action, value);
strcat(canonical_headers, value);
strcat(canonical_headers, "\n");
const char* signed_headers = "content-type;host;x-tc-action";
const char* payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}";
char hashed_request_payload[100] = {0};
sha256_hex(payload, hashed_request_payload);

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

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

// ***** 步骤 3: 计算签名 *****
char k_key[64] = {0};
sprintf(k_key, "%s%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, &outp
```

```
ut_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。

错误码	错误描述
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。

签名方法

最近更新时间：2026-07-07 10:58:12

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

参数名称	中文	参数值
SecretId	密钥 ID	AKID*****
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	ap-guangzhou
InstanceIds.0	待查询的实例 ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

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

2.1. 对参数排序

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

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

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

2.2. 拼接请求字符串

此步骤生成请求字符串。

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

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

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

```
zhou&SecretId=AKID*****&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原字符串

此步骤生成签名原字符串。

签名原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 编码前需要先用 UTF-8 进行编码。

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

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

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

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

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

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

- [Signature Demo](#)

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

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-`

guangzhou&SecretId=AKID*****&Signature=7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D&Timestamp=1465185768&Version=2017-03-12。

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

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

注意：以下代码仅适用于 API 3.0，不能直接用于其他的签名流程，请以对应的实际文档为准。

Java

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

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

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

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

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

```
url.append(k).append("=").append(URLEncoder.encode(params.get(k).toString(), CHARSET)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

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

Python

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

```
# -*- coding: utf8 -*-
import base64
import hashlib
import hmac
import os
import time

import requests

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
secret_id = os.getenv("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
secret_key = os.getenv("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
// $params["Signature"] = $signature;
// $url = "https://cvm.tencentcloudapi.com/?".http_build_query($params);
// echo $url.PHP_EOL;
// $ch = curl_init();
// curl_setopt($ch, CURLOPT_URL, $url);
// $output = curl_exec($ch);
// curl_close($ch);
// echo json_decode($output);
```

Ruby

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

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
```

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

method = 'GET'
endpoint = 'cvm.tencentcloudapi.com'
data = {
  'Action' => 'DescribeInstances',
  'InstanceIds.0' => 'ins-09dx96dg',
  'Limit' => 20,
  'Nonce' => 11886,
  'Offset' => 0,
  'Region' => 'ap-guangzhou',
  'SecretId' => secret_id,
  'Timestamp' => 1465185768, # Time.now.to_i
  'Version' => '2017-03-12',
}
sign = method + endpoint + '/?'
params = []
data.sort.each do |item|
  params << "#{item[0]}=#{item[1]}"
end
sign += params.join('&')
digest = OpenSSL::Digest.new('sha1')
data['Signature'] = Base64.encode64(OpenSSL::HMAC.digest(digest, secret_key, sign))
puts data['Signature']

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

DotNet

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

public class Application {
  public static string Sign(string signKey, string secret)
  {

```

```
string signRet = string.Empty;
using (HMACSHA1 mac = new HMACSHA1(Encoding.UTF8.GetBytes(signKey)))
{
    byte[] hash = mac.ComputeHash(Encoding.UTF8.GetBytes(secret));
    signRet = Convert.ToBase64String(hash);
}
return signRet;
}

public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, string requestHost, string requestPath)
{
    string retStr = "";
    retStr += requestMethod;
    retStr += requestHost;
    retStr += requestPath;
    retStr += "?";
    string v = "";
    foreach (string key in requestParams.Keys)
    {
        v += string.Format("{0}={1}&", key, requestParams[key]);
    }
    retStr += v.TrimEnd('&');
    return retStr;
}

public static void Main(string[] args)
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

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



```
param.Add("Nonce", "11886");
// param.Add("Nonce", Math.Abs(new Random().Next()).ToString());

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

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

NodeJS

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

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

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

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

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

```
}

function main(){
  // 密钥参数
  // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
  const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
  // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
  const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

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

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

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

  // 2. 拼接签名原文字符串
  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：2026-07-07 10:58:12

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：2026-07-07 10:58:13

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。

转发规则相关接口

修改转发规则属性

最近更新时间：2026-07-07 10:57:50

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

ModifyRulesAttributes修改转发规则属性，本接口为异步接口，返回成功后需以返回的RequestID为入参，调用DescribeAsyncJobs接口查询本次任务是否成功。

一条规则最多支持10个转发条件（Conditions），5个转发动作（Actions）。

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：ModifyRulesAttributes。
Version	是	String	公共参数，本接口取值：2025-10-30。
Region	是	String	公共参数，详见产品支持的 地域列表 。
ListenerId	是	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
Rules.N	是	Array of RuleModify	转发规则列表。
DryRun	否	Boolean	是否只预检查此次请求。 示例值：false

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 修改转发规则属性

输入示例

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

<公共请求参数>

```
{
  "ListenerId": "lst-d9p3k7wa",
  "LoadBalancerId": "alb-f8q2xk9m",
  "Rules": [
    {
      "Actions": [
        {
          "FixedResponseConfig": {
            "Content": "OK",
            "ContentType": "text/html",
            "HttpCode": 200
          },
          "InsertHeaderConfig": {
            "Key": "ClientVer",
            "Value": "1.0",
            "ValueType": "UserDefined"
          },
          "Order": 1,
          "RedirectConfig": {
            "Host": "www.redirect.com",
            "HttpCode": 301,
            "Path": "/redirecturl",
            "Port": "443",
            "Protocol": "HTTPS",
            "Query": "${query}"
          },
          "RemoveHeaderConfig": {
            "Key": "ClientVer"
          },
          "RewriteConfig": {
            "Host": "www.myhost.com",
            "Path": "/redirecturl",
            "Query": "${query}"
          }
        }
      ]
    }
  ]
}
```



```
"TargetGroupConfig": {
  "TargetGroupStickySession": {
    "Enabled": false,
    "Timeout": 1000
  },
  "TargetGroups": [
    {
      "TargetGroupId": "lbtg-0zrnc9qa",
      "Weight": 10
    }
  ],
  "Type": "TargetGroup"
},
"Conditions": [
  {
    "CookieConfig": [
      {
        "Key": "ClientId",
        "Value": "xea4cd"
      }
    ],
    "HeaderConfig": {
      "Key": "ClientVer",
      "Values": [
        "1.0",
        "1.1"
      ]
    },
    "HostConfig": [
      "www.myhost.com"
    ],
    "MethodConfig": [
      "GET"
    ],
    "PathConfig": [
      "/url"
    ],
    "QueryStringConfig": [
      {
        "Key": "ClientVer",
        "Value": "1.1"
      }
    ]
  }
],
```

```
"SourceIpConfig": [
  "192.168.1.3/32"
],
"Type": "Host"
},
],
"Priority": 100,
"RuleId": "rule-h8cy5gw1",
"RuleName": "newName"
}
]
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
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。
InvalidParameter	参数错误。
InvalidParameter.InvalidClientToken	ClientToken 格式无效。
InvalidParameter.InvalidFieldType	字段类型错误。
InvalidParameter.InvalidFieldValue	字段值错误。
InvalidParameter.InvalidQuotaCheckRequest	配额检查请求无效。
InvalidParameter.MissingAction	缺少 Action 参数。
InvalidParameter.MissingField	请填写必要参数。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceNotFound	资源不存在。
ResourceNotFound.Listener	监听器未找到。
ResourceNotFound.LoadBalancer	负载均衡资源不存在。
ResourceNotFound.Rule	转发规则未找到
ResourceUnavailable	资源不可用。
ResourceUnavailable.Listener	监听器处于不可用状态。
ResourceUnavailable.LoadBalancer	ALB状态不可用。

错误码	描述
ResourceUnavailable.Rule	规则状态不可用。
ResourceUnavailable.TargetGroup	目标组处于不可用状态
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

查询转发规则

最近更新时间：2026-07-07 10:57:50

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询转发规则

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：DescribeRules。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ListenerId	是	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
Filters.N	否	Array of Filter	支持的过滤条件如下：
MaxResults	否	Integer	返回列表的数量，默认为20，最大值为100。 示例值：20
NextToken	否	String	下一次查询的Token值。第一次查询和没有下一次查询时，无需填写。如果有下一次查询，取值为上一次 API 调用返回的 NextToken 值。 示例值：3974ae26e***
RuleIds.N	否	Array of String	转发规则 ID 列表，ID 格式为 rule- 后接 8 位字母数字。 示例值：["rule-h8cy5gwl"]

3. 输出参数

参数名称	类型	描述
NextToken	String	下一次查询的Token值，如果当前是最后一页，返回为空。 示例值：3974ae26e***
Rules	Array of RuleOutput	转发规则列表。
TotalCount	Integer	总的转发规则个数（根据监听器ID、规则ID等条件过滤后）。 示例值：3
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 查询规则列表

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "ListenerId": "lst-d9p3k7wa",
  "MaxResults": 100
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "NextToken": "",
    "Rules": [
      {
        "RuleId": "rule-h8cy5gwl",
        "Priority": 1,
        "RuleName": "rule1",
        "Direction": "Request",
        "Conditions": [
          {
            "Type": "Host",
```

```
"HostConfig": [
  "test.com"
],
"Actions": [
  {
    "Type": "TargetGroup",
    "Order": 1,
    "TargetGroupConfig": {
      "TargetGroups": [
        {
          "TargetGroupId": "lbtg-0zrnc9qa",
          "Weight": 10
        }
      ]
    }
  },
  {
    "Status": "Active",
    "Tags": []
  }
],
"RequestId": "ed300290-70cc-41f6-af02-50e151f91f90"
}
```

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

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidFilter	无效的过滤器
InvalidParameter	参数错误。
InvalidParameter.InvalidFieldType	字段类型错误。
InvalidParameter.InvalidFieldValue	字段值错误。
InvalidParameter.MissingAction	缺少 Action 参数。
InvalidParameter.MissingField	请填写必要参数。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceNotFound	资源不存在。
ResourceNotFound.Listener	监听器未找到。
ResourceNotFound.LoadBalancer	负载均衡资源不存在。
ResourceNotFound.Rule	转发规则未找到
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.ServiceRoleUnauthorized	服务角色未授权
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

删除转发规则

最近更新时间：2026-07-07 10:57:51

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

DeleteRules删除转发规则，本接口为异步接口，返回成功后需以返回的RequestId为入参，调用DescribeAsyncJobs接口查询本次任务是否成功。

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：DeleteRules。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ListenerId	是	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
RuleIds.N	是	Array of String	转发规则 ID 列表，ID 格式为 rule- 后接 8 位字母数字。 示例值：["rule-h8cy5gwl"]
DryRun	否	Boolean	是否只预检查此次请求。 示例值：false

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 删除转发规则

输入示例

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

{
  "ListenerId": "lst-d9p3k7wa",
  "LoadBalancerId": "alb-f8q2xk9m",
  "RuleIds": [
    "rule-h8cy5gw1"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.InvalidClientToken	ClientToken 格式无效。
InvalidParameter.InvalidFieldFormat	字段格式错误。
InvalidParameter.InvalidFieldType	字段类型错误。
InvalidParameter.InvalidFieldValue	字段值错误。
InvalidParameter.MissingAction	缺少 Action 参数。
InvalidParameter.MissingField	请填写必要参数。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceNotFound	资源不存在。
ResourceNotFound.Listener	监听器未找到。
ResourceNotFound.LoadBalancer	负载均衡资源不存在。
ResourceNotFound.Rule	转发规则未找到
ResourceUnavailable	资源不可用。
ResourceUnavailable.Listener	监听器处于不可用状态。

错误码	描述
ResourceUnavailable.LoadBalancer	ALB状态不可用。
ResourceUnavailable.PreviousTaskNotCompleted	上个任务未完成。
ResourceUnavailable.Rule	规则状态不可用。
ResourceUnavailable.UpstreamStatusBlocked	上级资源状态阻断。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.ServiceRoleUnauthorized	服务角色未授权
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

创建转发规则

最近更新时间：2026-07-09 01:09:58

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

CreateRules创建转发规则，本接口为异步接口，返回成功后需以返回的RequestId为入参，调用DescribeAsyncJobs接口查询本次任务是否成功。

一条规则最多支持10个转发条件（Conditions），5个转发动作（Actions）。

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：CreateRules。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ListenerId	是	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
Rules.N	是	Array of RuleInput	转发规则列表。
ClientToken	否	String	客户端Token，用于保证请求的幂等性。从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken只支持ASCII字符。若您未指定，则系统自动使用API请求的RequestId作为ClientToken标识。每次API请求的RequestId不一样。 示例值：3e727335-f94c-468f-9989-21759a3b6d**e8
DryRun	否	Boolean	是否只预检查此次请求。 示例值：false

3. 输出参数

参数名称	类型	描述
RuleIds	Array of String	转发规则 ID 列表，ID 格式为 rule- 后接 8 位字母数字。 示例值：["rule-h8cy5gwl"]
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 创建规则

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "ListenerId": "lst-d9p3k7wa",
  "Rules": [
    {
      "Direction": "Request",
      "Priority": 1,
      "RuleName": "rule1",
      "Conditions": [
        {
          "Type": "Host",
          "HostConfig": [
            "test.com"
          ]
        }
      ],
      "Actions": [
        {
          "Type": "TargetGroup",
          "Order": 1,
          "TargetGroupConfig": {
            "TargetGroups": [
              {
                "TargetGroupId": "lbtg-0zrnc9qa",
                "Weight": 10
              }
            ]
          }
        }
      ]
    }
  ]
}
```

```
]
}
}
]
}
]
}
```

输出示例

```
{
  "Response": {
    "RuleIds": [
      "rule-h8cy5gw1"
    ],
    "RequestId": "d3614848-765c-4890-b381-72a192ac8c21"
  }
}
```

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

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
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。
FailedOperation.ResourceAlreadyExists	资源已存在。
FailedOperation.RuleAlreadyExists	规则优先级已存在。
InternalServerError	内部错误。
InvalidFilter	无效的过滤器
InvalidParameter	参数错误。
InvalidParameter.InvalidClientToken	ClientToken 格式无效。
InvalidParameter.InvalidFieldFormat	字段格式错误。
InvalidParameter.InvalidFieldType	字段类型错误。
InvalidParameter.InvalidFieldValue	字段值错误。
InvalidParameter.MissingAction	缺少 Action 参数。
InvalidParameter.MissingField	请填写必要参数。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.BusinessParameterMismatch	业务参数不一致。
LimitExceeded	超过配额限制。
LimitExceeded.Quota	配额超限。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceNotFound	资源不存在。
ResourceNotFound.Listener	监听器未找到。
ResourceNotFound.LoadBalancer	负载均衡资源不存在。
ResourceNotFound.TargetGroup	目标组未找到。

错误码	描述
ResourceUnavailable	资源不可用。
ResourceUnavailable.Listener	监听器处于不可用状态。
ResourceUnavailable.LoadBalancer	ALB状态不可用。
ResourceUnavailable.PreviousTaskNotCompleted	上个任务未完成。
ResourceUnavailable.TargetGroup	目标组处于不可用状态
ResourceUnavailable.UpstreamStatusBlocked	上级资源状态阻断。
ResourcesSoldOut	资源售罄。
UnauthorizedOperation	未授权操作。
UnauthorizedOperation.ServiceRoleUnauthorized	服务角色未授权
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedOperation.InvalidStateTransition	无效的状态转换。

目标组相关接口

从目标组内移除后端服务

最近更新时间：2026-07-09 01:09:59

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

从目标组内移除后端服务

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：RemoveTargetsFromTargetGroup。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TargetGroupId	是	String	目标组 ID，格式为 lbtg- 后接 8 位字母数字。 示例值：lbtg-0zrnc9qa
Targets.N	是	Array of TargetToRemove	需要从目标组移除的后端服务列表。单次请求最多移除 50 个后端服务。
DryRun	否	Boolean	是否预览此次请求。 - false （默认）：发送普通请求，直接移除后端服务。 - true ：发送预览请求，检查移除后端服务的参数、格式、业务限制等是否符合要求。 示例值：false

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 从目标组内移除后端服务

输入示例

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

```
{
  "TargetGroupId": "lbtg-0zrnc9qa",
  "Targets": [
    {
      "Port": 80,
      "TargetIp": "10.0.0.1"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

修改目标组内后端服务信息

最近更新时间：2026-07-09 01:09:59

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

修改目标组内后端服务信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyTargetsInTargetGroup。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TargetGroupId	是	String	目标组 ID，格式为 lbtg- 后接 8 位字母数字。 示例值：lbtg-0zrnc9qa
Targets.N	是	Array of TargetToModify	需要修改的后端服务列表。
DryRun	否	Boolean	是否预览此次请求。 - false （默认）：发送普通请求，直接修改后端服务信息。 - true ：发送预览请求，检查修改后端服务的参数、格式、业务限制等是否符合要求。 示例值：false

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 修改目标组内后端服务信息

输入示例

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

<公共请求参数>

```
{
  "TargetGroupId": "lbtg-0zrnc9qa",
  "Targets": [
    {
      "Port": 80,
      "TargetIp": "10.0.0.1",
      "Weight": 100
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

修改目标组

最近更新时间：2026-07-09 01:10:00

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

修改目标组。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifyTargetGroupAttributes。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
DryRun	否	Boolean	是否预览此次请求。 false（默认）：发送普通请求，直接修改目标组。 true：发送预览请求，检查修改目标组的参数、格式、业务限制等是否符合要求。 示例值：true
HealthCheckConfig	否	HealthCheckConfig	健康检查配置。
KeepaliveEnabled	否	Boolean	是否开启长连接。 示例值：true
SchedulerAlgorithm	否	String	调度算法。取值： wrr：加权轮询，按照权重选择后端服务器，权重越高的服务器被轮询到的概率越高。 wlc：加权最小连接数，当不同后端服务器权重值相同时，当前连接数越小的后端服务器被轮询到的概率越高。 示例值：wrr

参数名称	必选	类型	描述
StickySessionConfig	否	StickySessionConfig	会话保持配置。
TargetGroupId	否	String	目标组 ID，格式为 lbtg- 后接 8 位字母数字。 示例值：lbtg-0zrnc9qa
TargetGroupName	否	String	目标组名称。长度为 1~255 个字符，可包含数字、大小写字母、中文、半角句号（.）、下划线（_）和短划线（-）。不传目标组名称时默认使用ID作为目标组名称。 示例值：test-group

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 修改目标组

输入示例

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

```
{
  "HealthCheckConfig": {
    "HealthCheckCodes": [
      "http_1xx"
    ],
    "HealthCheckEnabled": true,
    "HealthCheckHealthyThreshold": 2,
    "HealthCheckHost": "example.com",
    "HealthCheckHttpVersion": "HTTP1.1",
    "HealthCheckInterval": 10,
    "HealthCheckMethod": "GET",
    "HealthCheckPath": "/hello/index.html",
    "HealthCheckPort": 80,
    "HealthCheckProtocol": "HTTP",
```

```
"HealthCheckTimeout": 2,
"HealthCheckUnhealthyThreshold": 2
},
"KeepaliveEnabled": true,
"SchedulerAlgorithm": "wrr",
"StickySessionConfig": {
  "Cookie": "SERVERID",
  "CookieTimeout": 20,
  "StickySessionEnabled": true,
  "StickySessionType": "Insert"
},
"TargetGroupId": "lbtg-0zrnc9qa",
"TargetGroupName": "test-group"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

根据后端服务查询绑定的目标组

最近更新时间：2026-07-07 10:57:53

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

根据子机查询绑定的目标组

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeTargetGroupsByTarget。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TargetId.N	否	Array of String	后端服务实例 ID，CVM 实例格式为 ins- 后接 8 位字母数字。 示例值：["ins-x8q2m4pa"]

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	总数。 示例值：10
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 根据后端服务查询绑定的目标组

输入示例

```
POST / HTTP/1.1
Host: alb.tencentcloudapi.com
```

```
Content-Type: application/json
X-TC-Action: DescribeTargetGroupsByTarget
<公共请求参数>
```

```
{
  "TargetId": [
    "ins-x8q2m4pa"
  ]
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 1,
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

查询目标组列表

最近更新时间：2026-07-07 10:57:53

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询目标组列表

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeTargetGroups。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Filters.N	否	Array of Filter	过滤器。通过指定的过滤条件来查询后端服务，支持： – Name的值为VpcId。通过VPC实例来筛选目标组。Values的值为VPC唯一ID列表。 – Name的值为TargetType。通过后端服务类型来筛选目标组。Values的值可以取为Instance。 – Name的值为TargetGroupName。通过目标组名称来筛选目标组。Values的值为目标组名称列表。 – Name的值为Protocol。通过目标组后端服务协议来筛选目标组。Values的值为目标组后端服务协议列表。 – 通过标签方式筛选。
MaxResults	否	Integer	返回列表的数量，默认为20，最大值为100。 示例值：20
NextToken	否	String	下一次查询的Token值。第一次查询和没有下一次查询时，无需填写。如果有下一次查询，取值为上一次 API 调用返回的 NextToken 值。 示例值：3974ae26e***
TargetGroupIds.N	否	Array of String	目标组 ID 列表，ID 格式为 lbtg- 后接 8 位字母数字。 示例值：["lbtg-0zrnc9qa"]

3. 输出参数

参数名称	类型	描述
NextToken	String	下一次查询的Token值，如果当前是最后一页，返回为空。 注意：此字段可能返回 null，表示取不到有效值。 示例值：3974ae26e***
TargetGroups	Array of TargetGroupOutput	目标组信息。
TotalCount	Integer	目标组总数。 示例值：3
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 查询目标组列表

输入示例

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

{
  "Filters": [
    {
      "Name": "Protocol",
      "Values": [
        "HTTP",
        "HTTPS"
      ]
    }
  ],
  "MaxResults": 100
}
```

输出示例

```
{
  "Response": {
    "TotalCount": 2,
    "TargetGroups": [
```



```
{
  "TargetGroupId": "lbtg-0zrnc9qa",
  "TargetGroupName": "group1",
  "TargetType": "Instance",
  "Protocol": "HTTP",
  "VpcId": "vpc-92hffaxb",
  "SchedulerAlgorithm": "wrr",
  "CreateTime": "2026-06-03T11:18:25+08:00",
  "KeepaliveEnabled": false,
  "Tags": [],
  "RelatedLoadBalancersCount": 2,
  "TargetGroupStatus": "Active",
  "HealthCheckConfig": {
    "HealthCheckEnabled": true,
    "HealthCheckPort": 0,
    "HealthCheckProtocol": "HTTP",
    "HealthCheckCodes": [
      "http_1xx",
      "http_2xx",
      "http_3xx",
      "http_4xx"
    ],
    "HealthCheckInterval": 5,
    "HealthCheckHttpVersion": "HTTP1.1",
    "HealthCheckMethod": "HEAD",
    "HealthCheckPath": "/",
    "HealthCheckTimeout": 2,
    "HealthCheckHealthyThreshold": 3,
    "HealthCheckUnhealthyThreshold": 3,
    "HealthCheckHost": "aaa.com"
  },
  "StickySessionConfig": {
    "StickySessionEnabled": false
  },
  {
    "TargetGroupId": "lbtg-0zrnc9qa",
    "TargetGroupName": "group2",
    "TargetType": "Instance",
    "Protocol": "HTTP",
    "VpcId": "vpc-92hffaxb",
    "SchedulerAlgorithm": "wrr",
    "CreateTime": "2026-06-03T11:17:34+08:00",
    "KeepaliveEnabled": false,
    "Tags": [],
```

```
"RelatedLoadBalancersCount": 0,
"TargetGroupStatus": "Active",
"HealthCheckConfig": {
  "HealthCheckEnabled": true,
  "HealthCheckPort": 0,
  "HealthCheckProtocol": "HTTP",
  "HealthCheckCodes": [
    "http_1xx",
    "http_2xx",
    "http_3xx",
    "http_4xx"
  ],
  "HealthCheckInterval": 5,
  "HealthCheckHttpVersion": "HTTP1.1",
  "HealthCheckMethod": "HEAD",
  "HealthCheckPath": "/",
  "HealthCheckTimeout": 2,
  "HealthCheckHealthyThreshold": 3,
  "HealthCheckUnhealthyThreshold": 3,
  "HealthCheckHost": "aaa.com"
},
"StickySessionConfig": {
  "StickySessionEnabled": false
}
},
"RequestId": "fb47afdf-9728-4362-ad51-7049878fca3a"
}
```

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](#)。

查询目标组内后端服务

最近更新时间：2026-07-07 10:57:54

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询目标组内后端服务

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：DescribeTargetGroupTargets。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TargetGroupId	是	String	目标组 ID，格式为 lbtg- 后接 8 位字母数字。 示例值：lbtg-0zrnc9qa
Filters.N	否	Array of Filter	过滤器。通过指定的过滤条件来查询后端服务，支持： – Name的值为TargetId。通过资源ID来筛选后端服务，当目标组后端类型为Instance时生效。Values的值为Cvm或Eni的资源ID。 – Name的值为TargetIp。通过资源IP来筛选后端服务，当目标组后端类型为Ip时生效。Values的值为后端服务的IP。 – 通过标签方式筛选。
MaxResults	否	Integer	返回列表的数量，默认为20，最大值为100。 示例值：20
NextToken	否	String	下一次查询的Token值。第一次查询和没有下一次查询时，无需填写。 如果有下一次查询，取值为上一次 API 调用返回的 NextToken 值。 示例值：3974ae26e***

3. 输出参数

参数名称	类型	描述
NextToken	String	下一次查询的Token值，如果当前是最后一页，返回为空。 示例值：3974ae26e***

参数名称	类型	描述
Targets	Array of TargetOutput	后端服务信息。
TotalCount	Integer	目标组内后端服务总数。 示例值：3
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 查询目标组内后端服务

输入示例

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

{
  "MaxResults": 20,
  "TargetGroupId": "lbtg-0zrnc9qa"
}
```

输出示例

```
{
  "Response": {
    "NextToken": "",
    "Targets": [
      {
        "TargetIp": "10.0.0.1",
        "Port": 80,
        "Weight": 10,
        "TargetStatus": "Active",
        "TargetType": "CVM",
        "TargetId": "ins-x8q2m4pa",
        "TargetName": "",
        "EniId": "eni-p4q7x2ma"
      }
    ],
    "TotalCount": 1,
  }
}
```

```
"RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
}
```

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](#)。

删除目标组

最近更新时间：2026-07-07 10:57:54

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

删除目标组。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteTargetGroups。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
DryRun	否	Boolean	是否预览此次请求。 – false （默认）：发送普通请求，直接删除目标组。 – true ：发送预览请求，检查删除目标组的参数、格式、业务限制等是否符合要求。 示例值：true
TargetGroupIds.N	否	Array of String	目标组 ID 列表，ID 格式为 lbtg- 后接 8 位字母数字。 示例值：["lbtg-0zrnc9qa"]

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 删除目标组

输入示例

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

```
{
  "TargetGroupIds": [
    "lbtg-0zrnc9qa"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

创建目标组

最近更新时间：2026-07-09 01:10:01

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

目标组相关接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateTargetGroup。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TargetType	是	String	目标组类型。取值： Instance（默认）：Cvm服务器类型或者Eni网卡类型。 示例值：Instance
VpcId	是	String	私有网络 ID。 示例值：vpc-92hffaxb
DryRun	否	Boolean	是否预览此次请求。 false（默认）：发送普通请求，直接创建目标组。 true：发送预览请求，检查创建目标组的参数、格式、业务限制等是否符合要求。 示例值：false
HealthCheckConfig	否	HealthCheckConfig	健康检查配置。
KeepaliveEnabled	否	Boolean	是否开启长连接。

参数名称	必选	类型	描述
			示例值：true
Protocol	否	String	后端服务协议类型。取值： HTTP（默认）：支持绑定HTTP、HTTPS的监听器 HTTPS：支持绑定HTTPS类型的监听器 GRPC：支持绑定HTTPS类型的监听器 GRPCS：支持绑定HTTPS类型的监听器 示例值：HTTP
SchedulerAlgorithm	否	String	调度算法。取值： wrr（默认）：加权轮询，按照权重选择后端服务器，权重越高的服务器被轮询到的概率越高。 wlc：加权最小连接数，当不同后端服务器权重值相同时，当前连接数越小的后端服务器被轮询到的概率越高。 示例值：wrr
StickySessionConfig	否	StickySessionConfig	会话保持配置。
Tags.N	否	Array of TagInfo	标签。
TargetGroupName	否	String	目标组名称。默认为目标组ID。长度为 1-255 个字符，可包含数字、大小写字母、中文、半角句号（.）、下划线（_）和短划线（-）。 示例值：test-group

3. 输出参数

参数名称	类型	描述
TargetGroupId	String	目标组 ID，格式为 lbtg- 后接 8 位字母数字。 示例值：lbtg-0zrnc9qa
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 创建目标组

输入示例

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

{
  "HealthCheckConfig": {
    "HealthCheckCodes": [
      "http_1xx"
    ],
    "HealthCheckEnabled": true,
    "HealthCheckHealthyThreshold": 2,
    "HealthCheckHost": "example.com",
    "HealthCheckHttpVersion": "HTTP1.1",
    "HealthCheckInterval": 10,
    "HealthCheckMethod": "GET",
    "HealthCheckPath": "/hello/index.html",
    "HealthCheckPort": 80,
    "HealthCheckProtocol": "HTTP",
    "HealthCheckTimeout": 2,
    "HealthCheckUnhealthyThreshold": 2
  },
  "KeepaliveEnabled": true,
  "Protocol": "HTTP",
  "SchedulerAlgorithm": "wrr",
  "StickySessionConfig": {
    "Cookie": "SERVERID",
    "CookieTimeout": 20,
    "StickySessionEnabled": true,
    "StickySessionType": "Insert"
  },
  "Tags": [
    {
      "TagKey": "key-xxx",
      "TagValue": "value-xxx"
    }
  ],
  "TargetGroupName": "test-group",
  "TargetType": "Instance",
  "VpcId": "vpc-92hffaxb"
}
```

输出示例

```
{
  "Response": {
    "TargetGroupId": "lbtg-0zrnc9qa",
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

向目标组内添加后端服务

最近更新时间：2026-07-09 01:10:02

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

向目标组内添加后端服务

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：AddTargetsToTargetGroup。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
TargetGroupId	是	String	目标组 ID，格式为 lbtg- 后接 8 位字母数字。 示例值：lbtg-0zrnc9qa
Targets.N	是	Array of TargetToAdd	需要添加至目标组的后端服务列表。单次请求最多支持添加 50 个后端服务。
DryRun	否	Boolean	是否预览此次请求。 - false （默认）：发送普通请求，直接添加后端服务至目标组。 - true ：发送预览请求，检查添加后端服务的参数、格式、业务限制等是否符合要求。 示例值：false

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 向目标组内添加后端服务

输入示例

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

<公共请求参数>

```
{
  "TargetGroupId": "lbtg-0zrnc9qa",
  "Targets": [
    {
      "Port": 80,
      "TargetIp": "10.0.0.1",
      "Weight": 100
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

监听器相关接口

修改监听器

最近更新时间：2026-07-07 10:57:56

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

修改监听器属性

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：ModifyListenerAttributes。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ListenerId	是	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
CaCertificateIds.N	否	Array of String	监听器配置的CA证书ID列表。目前仅支持添加1个CA证书。 示例值：["Fm8Wp7x8"]
CaEnabled	否	Boolean	是否开启双向认证。 取值： true：开启。 false（默认值）：不开启。 示例值：false
CertificateIds.N	否	Array of String	服务器证书 ID 列表。 示例值：["Fm8Wp7x8"]
ClientToken	否	String	客户端Token，用于保证请求的幂等性。 从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken只支持ASCII字符。

参数名称	必选	类型	描述
			示例值：3e727335-f94c-468f-9989-21759a3b6de8
DefaultActions.N	否	Array of DefaultAction	默认转发规则动作列表。目前监听器仅支持添加 1 个默认转发规则动作。
GzipEnabled	否	Boolean	是否启用 Gzip 压缩。 示例值：false
Http2Enabled	否	Boolean	是否开启HTTP/2特性。只有 HTTPS 协议支持此参数。 示例值：true
IdleTimeout	否	Integer	指定连接空闲超时时间。单位：秒。 取值范围：1~600。 默认值：15。 如果在设定时间内一直没有访问请求，负载均衡会暂时断开当前连接，下次请求来临时重新建立新的连接。 示例值：15
ListenerName	否	String	自定义监听名称。长度为 1~255 个字符，必须是中文和无害字符串中的字符，可包含中文、字母、数字、短划线（-）、正斜线（/）、半角句号（.）、下划线（_）。 示例值：http_listener
RequestTimeout	否	Integer	指定请求超时时间。单位：秒。 取值：1~600。 默认值：60。 如果在超时时间内后端服务器一直没有响应，负载均衡将放弃等待，并给客户端返回HTTP 504错误码。 示例值：60
SecurityPolicyId	否	String	安全策略 ID，格式为 tls- 后接 8 位字母数字。 示例值：tls-t2ckyduq
XForwardedForConfig	否	XForwardedForConfig	XForwardedFor配置。

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 修改监听器

输入示例

```
POST / HTTP/1.1
Host: alb.tencentcloudapi.com
```

Content-Type: application/json

X-TC-Action: ModifyListenerAttributes

<公共请求参数>

```
{
  "CaCertificateIds": [
    "Fm8Wp7x8"
  ],
  "CaEnabled": false,
  "CertificateIds": [
    "Fm8Wp7x8"
  ],
  "DefaultActions": [
    {
      "TargetGroupConfig": {
        "TargetGroupStickySession": {
          "Enabled": false,
          "Timeout": 1000
        },
        "TargetGroups": [
          {
            "TargetGroupId": "lbtg-0zrnc9qa",
            "Weight": 10
          }
        ]
      },
      "Type": "TargetGroup"
    }
  ],
  "GzipEnabled": false,
  "Http2Enabled": true,
  "IdleTimeout": 15,
  "ListenerId": "lst-d9p3k7wa",
  "ListenerName": "http_listener",
  "LoadBalancerId": "alb-f8q2xk9m",
  "RequestTimeout": 60,
  "SecurityPolicyId": "tls-t2ckydug",
  "XForwardedForConfig": {
    "XForwardedForAlbIdEnabled": false,
    "XForwardedForClientSrcPortEnabled": false,
    "XForwardedForHostEnabled": false,
    "XForwardedForMode": "append",
    "XForwardedForPortEnabled": false,
    "XForwardedForProtoEnabled": false,
    "XTencentClientIDNEnabled": false,
```

```
"XTencentClientSDNEnabled": false,
"XTencentClientSerialEnabled": false,
"XTencentClientVerifyEnabled": false
}
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

监听器解绑扩展证书

最近更新时间：2026-07-07 10:57:56

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

DisassociateListenerAdditionalCertificates属于异步接口，即系统返回一个请求 ID，但该扩展证书尚未解绑成功，系统后台的解绑任务仍在进行。您可以调用DescribeListenerCertificates接口查询证书的解绑状态：若证书处于Disassociating状态，则证书正在解绑中。

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：DisassociateListenerAdditionalCertificates。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
CertificateIds.N	是	Array of String	待解绑的扩展证书 ID 列表。 示例值：["Fm8Wp7x8"]
ListenerId	是	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
ClientToken	否	String	客户端 Token，用于保证请求的幂等性。从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken 只支持 ASCII 字符。 若您未指定，则系统自动使用 API 请求的 RequestId 作为 ClientToken 标识。每次 API 请求的 RequestId 不一样。 示例值：3e727335-f94c-468f-9989-21759a3b6de8
DryRun	否	String	是否只预检此次请求，取值： true：发送检查请求，不会从 HTTPS和QUIC监听器解绑扩展证书。检查项包括是否填写了必需参数、请求格式、业务限制。如果检查不通过，则返回对应错误。如果检查通过，则返回错误码DryRunOperation。 false（默认值）：发送正常请求，通过检查后返回HTTP 2xx状态码并直接进行操作

参数名称	必选	类型	描述
			作。 示例值：true

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 监听器解绑扩展证书

输入示例

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

{
  "CertificateIds": [
    "Fm8Wp7x8"
  ],
  "ListenerId": "lst-d9p3k7wa",
  "LoadBalancerId": "alb-f8q2xk9m"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
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。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceNotFound	资源不存在。
ResourceNotFound.Listener	监听器未找到。

错误码	描述
ResourceNotFound.LoadBalancer	负载均衡资源不存在。
ResourceUnavailable	资源不可用。
ResourceUnavailable.Certificate	证书处于不可用状态（更新中/删除中）。
ResourceUnavailable.Listener	监听器处于不可用状态。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedOperation.CertificateNotBound	监听器不允许解绑未绑定的证书。
UnsupportedOperation.DisassociateDefaultCertificate	监听器不允许解绑默认证书。
UnsupportedOperation.UnsupportedProtocol	监听器协议不支持此操作。

查询监听器列表

最近更新时间：2026-07-07 10:57:56

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询监听器列表

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：DescribeListeners。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
Filters.N	否	Array of Filter	过滤条件列表，最大支持20个。支持以下几个字段 – Protocol : 协议类型 – Tags : 标签
ListenerIds.N	否	Array of String	监听器 ID 列表，ID 格式为 lst- 后接 8 位字母数字。 示例值：["lst-d9p3k7wa"]
MaxResults	否	Integer	本次读取的最大数据记录数量。 取值：1~100。 默认值：20 示例值：20
NextToken	否	String	下一次查询的令牌（Token）。为空时查询第一页。 示例值：2c44874464923a9f

3. 输出参数

参数名称	类型	描述
Listeners	Array of ListenerOutput	监听器信息。
LoadBalancerId	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
MaxResults	Integer	本次读取的最大数据记录数量。 示例值：20
NextToken	String	下一次查询的令牌。 示例值：2c44874464923a9f
TotalCount	Integer	总条目数。 示例值：200
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询监听器列表

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "MaxResults": 10
}
```

输出示例

```
{
  "Response": {
    "LoadBalancerId": "alb-f8q2xk9m",
    "Listeners": [
      {
        "ListenerId": "lst-d9p3k7wa",
        "ListenerName": "443",
        "ListenerProtocol": "HTTPS",
        "ListenerPort": 80,
```

```
"ListenerStatus": "Active",
"GzipEnabled": false,
"Http2Enable": false,
"CaEnable": false,
"RequestTimeout": 60,
"IdleTimeout": 15,
"TlsSecurityPolicyId": "tls-t2ckyduq",
"XForwardedForConfig": {
  "XForwardedForMode": "append",
  "XForwardedForProtoEnabled": false,
  "XForwardedForPortEnabled": false,
  "XForwardedForHostEnabled": false,
  "XForwardedForClientSrcPortEnabled": false,
  "XTencentClientSDNEnabled": false,
  "XTencentClientIDNEnabled": false,
  "XTencentClientSerialEnabled": false,
  "XTencentClientVerifyEnabled": false
},
"Tags": []
},
],
"TotalCount": 1,
"MaxResults": 1,
"NextToken": "",
"RequestId": "1d92eab2-8e5b-4c29-b1b8-4454479c7196"
}
```

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](#)。

查询监听器健康状态

最近更新时间：2026-07-07 10:57:57

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询监听器健康状态。

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：DescribeListenerHealthStatus。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ListenerId	是	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
IncludeRule	否	Boolean	是否包含转发规则的健康检查结果。为false时只返回默认转发规则的健康状态，为true时返回全部规则的健康状态(包含默认规则)。 取值： true：是。 false（默认值）：否。 示例值：false
MaxResults	否	Integer	本次读取的最大数据记录数量。 取值：1~100。 默认值：20 示例值：20
NextToken	否	String	下一页查询的Token值。第一次查询时，无需填写。 示例值：2c44874464923a9f

3. 输出参数

参数名称	类型	描述
ListenerId	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
ListenerPort	String	监听器端口。 示例值：80
ListenerProtocol	String	监听器协议。 示例值：HTTP
NextToken	String	下一次查询的令牌（Token）。为空时表示这是最后一页。 示例值：2c44874464923a9f
RuleHealthStatusInfos	Array of RuleHealthStatusInfo	转发规则健康状态。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询监听器健康状态

输入示例

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

{
  "IncludeRule": false,
  "ListenerId": "lst-d9p3k7wa",
  "LoadBalancerId": "alb-f8q2xk9m",
  "MaxResults": 20
}
```

输出示例

```
{
  "Response": {
    "ListenerId": "lst-d9p3k7wa",
    "ListenerPort": "80",
    "ListenerProtocol": "HTTP",
    "NextToken": "2c44874464923a9f",
    "RuleHealthStatusInfos": [
```

```
{
  "IsDefaultRule": "false",
  "RuleId": "rule-h8cy5gw1",
  "TargetGroupHealthInfos": [
    {
      "HealthCheckEnabled": true,
      "TargetGroupId": "lbtg-0zrnc9qa",
      "TargetHealthStatusInfos": [
        {
          "Status": "Healthy",
          "TargetId": "ins-x8q2m4pa",
          "TargetIp": "10.0.0.1",
          "TargetPort": 80
        }
      ],
      "Type": "TargetGroup"
    }
  ],
  "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
}
```

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](#)。

查询监听器详细信息

最近更新时间：2026-07-07 10:57:57

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询单个监听器详情

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：DescribeListenerDetail。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ListenerId	是	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m

3. 输出参数

参数名称	类型	描述
CaCertificateIds	Array of String	监听器绑定的CA证书ID列表。 示例值：["Fm8Wp7x8"]
CaEnabled	Boolean	是否开启双向认证。 示例值：false
CertificateIds	Array of String	服务器证书 ID 列表。 示例值：["Fm8Wp7x8"]
CreateTime	Timestamp ISO8601	监听器实例的创建时间。格式：ISO 8601（例如 2025-01-01T08:30:00+08:00） 示例值：2025-01-01T08:30:00+08:00

参数名称	类型	描述
DefaultActions	Array of DefaultAction	规则动作列表。
GzipEnabled	Boolean	是否启用 Gzip 压缩。 示例值：false
Http2Enabled	Boolean	是否开启HTTP/2特性。 示例值：true
IdleTimeout	Integer	指定连接空闲超时时间。单位：秒。 示例值：15
ListenerId	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
ListenerName	String	自定义监听名称。 示例值：http_listener
ListenerPort	Integer	负载均衡实例前端使用的端口。 示例值：80
ListenerProtocol	String	监听协议。 示例值：HTTP
ListenerStatus	String	监听器状态。取值：= – Active: 运行中。 – Provisioning: 创建中。 – Configuring: 变配中。 – ProvisionFailed: 创建失败 示例值：Active
LoadBalancerId	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
ModifyTime	Timestamp ISO8601	监听器实例的最后变更时间。格式：ISO 8601（例如 2025-01-01T08:30:00+08:00） 示例值：2025-01-01T08:30:00+08:00
RequestTimeout	Integer	请求超时时间。单位：秒。 示例值：60
SecurityPolicyId	String	安全策略 ID，格式为 tls- 后接 8 位字母数字。 示例值：tls-t2ckyduq
Tags	Array of TagInfo	标签。
XForwardedForConfig	XForwardedForConfig	XForwardedFor配置。
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询监听器详情

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "ListenerId": "lst-d9p3k7wa"
}
```

输出示例

```
{
  "Response": {
    "LoadBalancerId": "alb-f8q2xk9m",
    "ListenerId": "lst-d9p3k7wa",
    "ListenerName": "443",
    "ListenerPort": 80,
    "ListenerProtocol": "HTTPS",
    "ListenerStatus": "Active",
    "CertificateIds": [
      "Fm8Wp7x8"
    ],
    "CaCertificateIds": [],
    "GzipEnabled": false,
    "CaEnabled": false,
    "Http2Enabled": false,
    "SecurityPolicyId": "tls-t2ckydog",
    "IdleTimeout": 15,
    "RequestTimeout": 60,
    "XForwardedForConfig": {
      "XForwardedForMode": "append",
      "XForwardedForProtoEnabled": false,
      "XForwardedForPortEnabled": false,
      "XForwardedForHostEnabled": false,
      "XForwardedForClientSrcPortEnabled": false,
      "XTencentClientSDNEnabled": false,
      "XTencentClientIDNEnabled": false,
      "XTencentClientSerialEnabled": false,
      "XTencentClientVerifyEnabled": false
    }
  }
}
```

```
},
"DefaultActions": [
{
"Type": "TargetGroup",
"TargetGroupConfig": {
"TargetGroups": [
{
"TargetGroupId": "lbtg-0zrnc9qa",
"Weight": 10
}
]
}
},
],
"Tags": [],
"RequestId": "cb7e17c7-9e71-42d2-9fa6-ed98bef10c93"
}
```

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](#)。

查询监听器绑定证书列表

最近更新时间：2026-07-07 10:57:58

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

根据实例id和监听器id，查询指定监听器绑定的证书列表
若输入CertificateType为SVR，返回扩展服务器证书与默认服务器证书的信息
若输入CertificateType为CA，返回默认CA证书的信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeListenerCertificates。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
CertificateType	是	String	证书类型。取值：CA或SVR（服务器证书）。 示例值：SVR
ListenerId	是	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
MaxResults	否	Integer	本次读取的最大数据记录数量。取值：1~100。默认值：20。 示例值：20
NextToken	否	String	下一次查询的令牌（Token）。取值： 第一次查询和没有下一次查询时，均无需填写。 如果有下一次查询，取值为上一次 API 调用返回的 NextToken 值。 示例值：2c44874464923a9f

3. 输出参数

参数名称	类型	描述
Certificates	Array of CertificateInfo	监听器绑定的证书信息列表。
MaxResults	Integer	本次读取的最大数据记录数量。 示例值：20
NextToken	String	下一次查询的令牌。 注意：此字段可能返回 null，表示取不到有效值。 示例值：2c44874464923a9f
TotalCount	Integer	监听器绑定的证书总量。 示例值：200
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询监听器绑定证书列表

输入示例

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

{
  "CertificateType": "SVR",
  "ListenerId": "lst-d9p3k7wa",
  "LoadBalancerId": "alb-f8q2xk9m",
  "MaxResults": 20
}
```

输出示例

```
{
  "Response": {
    "Certificates": [
      {
        "CertificateId": "Fm8Wp7x8",
        "CertificateType": "SVR",
        "AssociatedTime": "2024-06-12T10:34:56Z",
        "IsDefault": true,
        "Status": "Associated"
      }
    ]
  }
}
```

```
}
],
"MaxResults": 20,
"NextToken": "2c44874464923a9f",
"TotalCount": 200,
"RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
}
```

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

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

错误码	描述
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

删除监听器

最近更新时间：2026-07-07 10:57:58

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

删除监听器

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DeleteListener。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ListenerIds.N	是	Array of String	监听器 ID 列表，ID 格式为 lst- 后接 8 位字母数字。 示例值：["lst-d9p3k7wa"]
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
ClientToken	否	String	客户端Token，用于保证请求的幂等性。 从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken只支持ASCII字符。 示例值：3e727335-f94c-468f-9989-21759a3b6de8

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 删除监听器

输入示例

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

{
  "ListenerIds": [
    "lst-d9p3k7wa"
  ],
  "LoadBalancerId": "alb-f8q2xk9m"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

创建监听器

最近更新时间：2026-07-07 10:57:59

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

创建监听器

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateListener。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
DefaultActions.N	是	Array of DefaultAction	默认转发规则动作列表。目前监听器仅支持添加 1 个默认转发规则动作。
ListenerPort	是	Integer	负载均衡实例前端使用的端口。取值：1~65535。 示例值：80
ListenerProtocol	是	String	监听协议。取值：HTTP、HTTPS 或 QUIC。 示例值：HTTP
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
CaCertificateIds.N	否	Array of String	监听器配置的CA证书ID列表。目前监听器仅支持添加 1 个 CA 证书。 当 CaEnabled 参数取值为 true 时，此参数必填。 示例值：["Fm8Wp7x8"]
CaEnabled	否	Boolean	是否开启双向认证。 取值： true：开启。 false（默认值）：不开启。 示例值：false
CertificateIds.N	否	Array of String	服务器证书 ID 列表。

参数名称	必选	类型	描述
			示例值: ["Fm8Wp7x8"]
ClientToken	否	String	客户端Token，用于保证请求的幂等性。 从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken只支持ASCII字符。 示例值: 3e727335-f94c-468f-9989-21759a3b6de8
GzipEnabled	否	Boolean	是否开启Gzip压缩。取值:true(默认值):是。false:否 示例值: false
Http2Enabled	否	Boolean	是否开启HTTP/2特性。HTTP 协议默认 false，HTTPS 协议默认 true。只有 HTTPS 协议支持此参数。 示例值: true
IdleTimeout	否	Integer	连接空闲超时时间。单位：秒。 取值范围：1~600。 默认值：15。 如果在超时时间内一直没有访问请求，负载均衡会断开当前连接，在下次请求到来时创建新的连接。 示例值: 15
ListenerName	否	String	自定义监听名称。长度为 1~255 个字符，必须是中文和无害字符串中的字符，可包含中文、字母、数字、短划线（-）、正斜线（/）、半角句号（.）、下划线（_）。 示例值: http_listener
RequestTimeout	否	Integer	请求超时时间。单位：秒。 取值：1~600。 默认值：60。 如果在超时时间内后端服务器没有返回响应，负载均衡将放弃等待，并给客户端返回HTTP 504错误码。 示例值: 60
SecurityPolicyId	否	String	安全策略 ID，格式为 tls- 后接 8 位字母数字。 示例值: tls-t2ckydu9
Tags.N	否	Array of TagInfo	标签列表。最大支持20个。
XForwardedForConfig	否	XForwardedForConfig	X-Forwarded-For配置

3. 输出参数

参数名称	类型	描述
ListenerId	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值: lst-d9p3k7wa
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 创建监听器

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "ListenerProtocol": "HTTPS",
  "ListenerPort": 80,
  "ListenerName": "443",
  "DefaultActions": [
    {
      "Type": "TargetGroup",
      "TargetGroupConfig": {
        "TargetGroups": [
          {
            "TargetGroupId": "lbtg-0zrnc9qa"
          }
        ]
      }
    }
  ],
  "CertificateIds": [
    "Fm8Wp7x8"
  ],
  "CaEnabled": false,
  "Http2Enabled": false,
  "GzipEnabled": false,
  "RequestTimeout": 60,
  "IdleTimeout": 15,
  "SecurityPolicyId": "tls-t2ckydg",
  "XForwardedForConfig": {
    "XForwardedForMode": "append",
    "XForwardedForProtoEnabled": false,
    "XForwardedForPortEnabled": false,
    "XForwardedForHostEnabled": false,
    "XForwardedForClientSrcPortEnabled": false,
    "XTencentClientSDNEnabled": false,
    "XTencentClientIDNEnabled": false,
```

```
"XTencentClientSerialEnabled": false,
"XTencentClientVerifyEnabled": false
}
}
```

输出示例

```
{
  "Response": {
    "ListenerId": "1st-d9p3k7wa",
    "RequestId": "e3bc3ab0-e453-4a83-92c4-8fcdda1e08c1"
  }
}
```

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

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

错误码	描述
UnsupportedOperation.TargetGroupProtocolMismatch	监听器协议与目标组后端转发协议不匹配
UnsupportedOperation.UnsupportedProtocol	监听器协议不支持此操作。

监听器绑定扩展证书

最近更新时间：2026-07-07 10:57:59

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

AssociateListenerAdditionalCertificates属于异步接口，即系统返回一个请求 ID，但该扩展证书尚未添加成功，系统后台的添加任务仍在进行。您可以调用DescribeListenerCertificates接口查询扩展证书的添加状态：

当HTTPS和QUIC监听器处于Associating状态时，表示扩展证书正在添加中。

当HTTPS和QUIC监听器处于Associated状态时，表示扩展证书添加成功。

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：AssociateListenerAdditionalCertificates。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
CertificateIds.N	是	Array of String	扩展证书 ID 列表。 示例值：["Fm8Wp7x8"]
ListenerId	是	String	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
ClientToken	否	String	客户端 Token，用于保证请求的幂等性。从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken 只支持 ASCII 字符。 若您未指定，则系统自动使用 API 请求的 RequestId 作为 ClientToken 标识。每次 API 请求的 RequestId 不一样。 示例值：3e727335-f94c-468f-9989-21759a3b6de8
DryRun	否	String	是否只预检此次请求，取值： true：发送检查请求，不会为HTTPS和QUIC监听器添加扩展证书。检查项包括是否填写了必需参数、请求格式、业务限制。如果检查不通过，则返回对应错误。如果检查通过，则返回错误码DryRunOperation。 false（默认值）：发送正常请求，通过检查后返回HTTP2xx状态码并直接进行操

参数名称	必选	类型	描述
			作。 示例值：true

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 监听器绑定扩展证书

输入示例

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

{
  "CertificateIds": [
    "Fm8Wp7x8"
  ],
  "ListenerId": "lst-d9p3k7wa",
  "LoadBalancerId": "alb-f8q2xk9m"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
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。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceNotFound	资源不存在。
ResourceNotFound.Certificate	证书未找到。

错误码	描述
ResourceNotFound.Listener	监听器未找到。
ResourceNotFound.LoadBalancer	负载均衡资源不存在。
ResourceUnavailable	资源不可用。
ResourceUnavailable.Certificate	证书处于不可用状态（更新中/删除中）。
ResourceUnavailable.Listener	监听器处于不可用状态。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedOperation.CertificateAlreadyBound	监听器不允许绑定已绑定的证书。
UnsupportedOperation.UnsupportedProtocol	监听器协议不支持此操作。

实例相关接口

更新负载均衡实例属性

最近更新时间：2026-07-09 01:10:06

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

ModifyLoadBalancerAttributes接口属于异步接口，即系统返回一个请求ID，但该应用型负载均衡实例属性尚未修改成功，系统后台的修改任务仍在进行。您可以调用[DescribeLoadBalancerDetail](#)查询应用型负载均衡实例属性的修改状态：

- 当应用型负载均衡实例属性处于**Configuring**状态时，表示应用型负载均衡实例属性正在修改中。
- 当应用型负载均衡实例属性处于**Active**状态时，表示应用型负载均衡实例属性修改成功。

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： ModifyLoadBalancerAttributes。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
ClientToken	否	String	客户端Token，用于保证请求的幂等性。 从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken只支持ASCII字符。 > 若您未指定，则系统自动使用API请求的RequestId作为ClientToken标识。每次API请求的RequestId不一样。 示例值：328980d0-b3d9-4368-96e8-1c3864bfc4b3
DeletionProtection	否	DeletionProtectionConfig	删除保护配置

参数名称	必选	类型	描述
DryRun	否	Boolean	是否只预检此次请求，取值： – true：发送检查请求，不会修改应用型负载均衡实例的属性。检查项包括是否填写了必需参数、请求格式、业务限制。如果检查不通过，则返回对应错误。如果检查通过，则返回错误码DryRunOperation。 – false（默认值）：发送正常请求，通过检查后返回HTTP_2xx状态码并直接进行操作。 示例值：true
LoadBalancerName	否	String	应用型负载均衡实例名称。长度为1~80个字符，可包含中文、字母、数字、短划线（-）、正斜线（/）、半角句号（.）和下划线（_）。 示例值：alb-f8q2xk9m-test

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 修改实例名称

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "LoadBalancerName": "test-name"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "ed530899-814d-4b1f-90f6-832a89458728"
  }
}
```

```
}  
  
}
```

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](#)。

修改负载均衡实例的网络类型

最近更新时间：2026-07-07 10:58:00

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

前提条件：

您已经创建应用型负载均衡实例，具体操作，请参见 CreateLoadBalancer。

当您需要通过此接口将应用型负载均衡实例的网络类型由私网变为公网时，您需要先创建一个弹性公网 IP。

使用说明：

ModifyLoadBalancerAddressType 接口属于异步接口，即系统返回一个请求 ID，但该应用型负载均衡实例的网络类型尚未变更成功，系统后台的变更任务仍在进行。您可以调用 DescribeLoadBalancerDetail 查询应用型负载均衡实例的网络类型的变更状态：

当应用型负载均衡实例处于 Configuring 状态时，表示应用型负载均衡实例的网络类型正在变更中。

当应用型负载均衡实例处于 Active 状态时，表示应用型负载均衡实例的网络类型变更成功。

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：ModifyLoadBalancerAddressType。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
AddressType	是	String	目标网络类型。取值： – Internet（公网型） 负载均衡实例分配公网 IP 地址，域名（DNS）解析至公网 IP，可在公网环境中直接访问，适用于对外提供服务的业务场景。 – Intranet（内网型） 负载均衡实例仅分配私网 IP 地址，域名（DNS）解析至私网 IP，仅支持在负载均衡实例所属 VPC 内的内网环境访问，适用于内部业务或对安全性要求较高的场景。 示例值：Internet
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
BandwidthPackageId	否	String	共享带宽包 ID。

参数名称	必选	类型	描述
			示例值：bwp-q8m2x4pa
DryRun	否	Boolean	是否只预检此次请求。取值： – true ：发送检查请求，不会更新实例的网络类型。检查项包括是否填写了必需参数、请求格式、业务限制。如果检查不通过，则返回对应错误。如果检查通过，则返回错误码 DryRunOperation。 – false （默认值）：发送正常请求，通过检查后返回 HTTP 2xx 状态码并直接进行操作。 示例值：false
ZoneMappings.N	否	Array of ZoneMappingsItem	可用区及子网映射结构体。 若当前地域支持 2 个及以上的可用区，至少需要添加 2 个可用区。

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 将实例类型修改为内网

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "AddressType": "Intranet"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "a44ec7d9-94ce-4c45-8fb0-bab95dbb5122"
  }
}
```

```
}  
  
}
```

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](#)。

查询负载均衡列表

最近更新时间：2026-07-30 01:09:07

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询实例配置。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeLoadBalancers。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Filters.N	否	Array of Filter	查询过滤条件，支持以下几个字段 - LoadBalancerId: 负载均衡实例 ID - LoadBalancerName: 负载均衡名称 - LoadBalancerStatus: 负载均衡状态 - VpcId: 私有网络 ID - tag:tag-key: 按标签键值对筛选，tag-key 请替换为实际的标签键。例如 tag:env 表示按标签键 env 筛选。 - AddressType: 网络类型 - Intranet: 内网 - Internet: 公网 - AddressIpVersion: - IPv4: IPv4 地址 - IPv6: IPv6 地址 - SecurityGroupId: 安全组ID
MaxResults	否	Integer	分批次查询时每次显示的条目数。取值范围：1~100，默认值：20。 示例值：20
NextToken	否	String	是否拥有下一次查询的令牌（Token）。取值： 第一次查询和没有下一次查询时，均无需填写。

参数名称	必选	类型	描述
			如果有下一次查询，取值为上一次API调用返回的NextToken值。 示例值：FFmyTO70tTpLG6l3FmYAXGKPd8r6k

3. 输出参数

参数名称	类型	描述
LoadBalancers	Array of LoadBalancer	应用型负载均衡实例列表。
MaxResults	Integer	分批次查询时每次显示的条目数。 示例值：20
NextToken	String	是否拥有下一次查询的令牌（Token）。取值： - 如果NextToken为空表示没有下一次查询。 - 如果NextToken有返回值，该取值表示下一次查询开始的令牌。 示例值：FFmyTO70tTpLG6l3FmYAXGKPd8r6k
TotalCount	Integer	列表条目数。 示例值：100
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 查询负载均衡实例列表

输入示例

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

{
  "MaxResults": 1
}
```

输出示例

```
{
  "Response": {
    "LoadBalancers": [
      {
        "AddressIpVersion": "IPv4",
        "AddressType": "Internet",
        "CreateTime": "2026-05-25 18:27:42",
        "DeletionProtection": {
          "DeletionProtectionEnabled": false,
          "Reason": ""
        },
        "Domain": "alb-f8q2xk9m-c0rpri8t217fy3ii.gz.tencentalb.com",
        "LoadBalancerBillingConfig": {
          "BandwidthPackageId": "bwp-q8m2x4pa",
          "ChargeType": "POSTPAID_BY_HOUR"
        },
        "LoadBalancerId": "alb-f8q2xk9m",
        "LoadBalancerName": "alb-f8q2xk9m",
        "LoadBalancerOperationLocks": [],
        "LoadBalancerStatus": "Active",
        "ModificationProtection": {
          "ModificationProtectionEnabled": false,
          "OperatorUin": "",
          "Reason": ""
        },
        "Tags": [],
        "VpcId": "vpc-92hffaxb"
      }
    ],
    "MaxResults": 1,
    "NextToken": "eyJJsYXN0X2Jlc291cmNlX2lkIjogImFsYi01cWw0MHZzayIsICJsYXN0X2RiX2lkIjogNjY1MCwgInRvdGFsX2NvdW50IjogMjd9",
    "TotalCount": 27,
    "RequestId": "d635fb06-fa8a-41fe-bb34-ef1b0615364f"
  }
}
```

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](#)。

查询负载均衡实例详细信息

最近更新时间：2026-07-07 10:58:01

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询指定负载均衡实例的详细信息。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeLoadBalancerDetail。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m

3. 输出参数

参数名称	类型	描述
LoadBalancerDetail	LoadBalancerDetail	负载均衡详细信息
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 查询实例详情

输入示例

```
POST / HTTP/1.1
Host: alb.tencentcloudapi.com
```



```
Content-Type: application/json
X-TC-Action: DescribeLoadBalancerDetail
```

<公共请求参数>

```
{
  "LoadBalancerId": "alb-f8q2xk9m"
}
```

输出示例

```
{
  "Response": {
    "LoadBalancerDetail": {
      "AccessLogConfig": {
        "LogSetId": "",
        "LogTopicId": ""
      },
      "AddressIpVersion": "IPv4",
      "AddressType": "Intranet",
      "CreateTime": "2025-10-30T12:00:00+08:00",
      "DeletionProtection": {
        "DeletionProtectionEnabled": false,
        "Reason": ""
      },
      "Domain": "alb-f8q2xk9m-vcpcq5puad9vp3xm.gz.tencentalb.com",
      "LoadBalancerBillingConfig": {
        "BandwidthPackageId": "bwp-q8m2x4pa",
        "ChargeType": "POSTPAID_BY_HOUR"
      },
      "LoadBalancerId": "alb-f8q2xk9m",
      "LoadBalancerName": "test-name",
      "LoadBalancerOperationLocks": [],
      "LoadBalancerStatus": "Active",
      "ModificationProtection": {
        "ModificationProtectionEnabled": false,
        "OperatorUin": "",
        "Reason": ""
      },
      "SecurityGroupIds": [],
      "Tags": [],
      "VpcId": "vpc-92hffaxb",
      "ZoneMappings": [
        {
          "LoadBalancerAddress": {
            "IPv4Address": [
```

```
{
  "Address": "172.16.80.118",
  "AddressId": ""
},
{
  "IPv6Address": []
},
{
  "Status": "Active",
  "SubnetId": "subnet-bi75pogm",
  "ZoneId": "ap-guangzhou-3"
},
{
  "LoadBalancerAddress": {
    "IPv4Address": [
      {
        "Address": "172.16.32.181",
        "AddressId": ""
      }
    ],
    "IPv6Address": []
  },
  "Status": "Active",
  "SubnetId": "subnet-bi75pogm",
  "ZoneId": "ap-guangzhou-2"
}
],
"RequestId": "f87fe97c-d4b3-4c1e-9256-36af067be287"
}
```

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](#)。

删除应用型负载均衡实例

最近更新时间：2026-07-09 01:10:07

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

DeleteLoadBalancers接口属于异步接口，即系统返回一个请求ID，但该应用型负载均衡实例尚未删除成功，系统后台的删除任务仍在进行。您可以调用DescribeLoadBalancerDetail查询应用型负载均衡实例的删除状态：

- 当应用型负载均衡实例处于Deleting状态时，表示应用型负载均衡实例正在删除中。
- 当查询不到指定的应用型负载均衡实例时，表示应用型负载均衡实例删除成功。

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：DeleteLoadBalancers。
Version	是	String	公共参数，本接口取值：2025-10-30。
Region	是	String	公共参数，详见产品支持的 地域列表。
LoadBalancerIds.N	是	Array of String	负载均衡实例 ID 列表，ID 格式为 alb- 后接 8 位字母数字。 示例值：["alb-f8q2xk9m"]
ClientToken	否	String	客户端Token，用于保证请求的幂等性。 从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken 只支持ASCII字符。 示例值：5A2CFF0E-5718-45B5-9D4D-70B3FF3898
DryRun	否	Boolean	是否只预检此次请求，取值： - true：发送检查请求，不会删除应用型负载均衡实例。检查项包括是否填写了必需参数、请求格式和业务限制。如果检查不通过，则返回对应错误。如果检查通过，则返回错误码DryRunOperation。 - false（默认值）：发送正常请求，通过检查后返回HTTP 2xx状态码并直接进入

参数名称	必选	类型	描述
			行操作。 示例值：true

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 删除实例

输入示例

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

{
  "LoadBalancerIds": [
    "alb-f8q2xk9m"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "a458011e-dc04-43bd-9851-3cc2276fb6fb"
  }
}
```

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

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

错误码	描述
MissingParameter	缺少参数错误。
ResourceInUse.LoadBalancerHasListeners	实例下还存在监听器，无法删除。

创建应用型负载均衡实例

最近更新时间：2026-07-09 01:10:08

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

CreateLoadBalancer接口属于异步接口，即系统返回一个实例ID，但该应用型负载均衡实例尚未创建成功，系统后台的创建任务仍在进行。您可以调用DescribeLoadBalancerDetail查询应用型负载均衡实例的创建状态：

- 当应用型负载均衡实例处于Provisioning状态时，表示应用型负载均衡实例正在创建中。
- 当应用型负载均衡实例处于Active状态时，表示应用型负载均衡实例创建成功。

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：CreateLoadBalancer。
Version	是	String	公共参数，本接口取值：2025-10-30。
Region	是	String	公共参数，详见产品支持的 地域列表 。
AddressType	是	String	应用型负载均衡的地址类型。取值： – Internet: 负载均衡具有公网IP地址，DNS域名被解析到公网IP，因此可以在公网环境访问。 – Intranet: 负载均衡只有私网IP地址，DNS域名被解析到私网IP，因此只能被负载均衡所在VPC的内网环境访问。 示例值：Internet
LoadBalancerBillingConfig	是	LoadBalancerBillingConfig	应用型负载均衡实例计费配置。
VpcId	是	String	私有网络 ID。 示例值：vpc-92hffaxb
ZoneMappings.N	是	Array of ZoneMappingsItem	可用区及私有网络子网映射列表，最多支持添加10个可用区。若当前地域支持2个及以上的可用区，至少需要添加2个可用区。

参数名称	必选	类型	描述
AddressIpVersion	否	String	IP 地址版本，取值 IPv4 或 IPv6。 示例值：IPv4
ClientToken	否	String	客户端Token，用于保证请求的幂等性。 从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken只支持ASCII字符。 示例值：3e727335-f94c-468f-9989-21759a3b6de8
DeleteProtection	否	DeletionProtectionConfig	删除保护配置。
DryRun	否	Boolean	是否只预检此次请求，取值： – true ：发送检查请求，不会创建应用型负载均衡实例。检查项包括是否填写了必需参数、请求格式和业务限制。如果检查不通过，则返回对应错误。如果检查通过，则返回错误码 DryRunOperation。 – false （默认值）：发送正常请求，通过检查后返回HTTP 2xx状态码并直接进行操作。 示例值：false
InternetAddressType	否	String	EIP 地址类型，可取值： – EIP ：普通弹性公网 IP – AntiDDoSEIP ：高防EIP – AnycastEIP ：加速 EIP – HighQualityEIP ：精品 IP。仅新加坡和中国香港支持精品IP。 – ResidentialEIP ：原生 IP 不传默认是EIP。 示例值：EIP
LoadBalancerName	否	String	应用型负载均衡实例名称。长度为1~80个字符，可包含中文、字母、数字、短划线（-）、正斜线（/）、半角句号（.）和下划线（_）。 示例值：alb1
Tags.N	否	Array of TagInfo	标签。

3. 输出参数

参数名称	类型	描述
LoadBalancerId	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
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 创建公网CLB

输入示例

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

{
  "VpcId": "vpc-92hffaxb",
  "AddressType": "Internet",
  "ZoneMappings": [
    {
      "SubnetId": "subnet-bi75pogm",
      "ZoneId": "ap-guangzhou-2"
    },
    {
      "SubnetId": "subnet-bi75pogm",
      "ZoneId": "ap-guangzhou-3"
    }
  ],
  "LoadBalancerBillingConfig": {
    "ChargeType": "POSTPAID_BY_HOUR"
  }
}
```

输出示例

```
{
  "Response": {
    "LoadBalancerId": "alb-f8q2xk9m",
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

安全策略相关接口

修改安全策略

最近更新时间：2026-07-09 01:10:08

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

修改自定义安全策略的属性，包括策略名称、TLS 协议版本和加密套件。修改后的配置将立即应用到所有关联该策略的 HTTPS 监听器。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：ModifySecurityPolicyAttributes。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SecurityPolicyId	是	String	安全策略 ID，格式为 tls- 后接 8 位字母数字。 示例值：tls-t2ckydu9
Ciphers.N	否	Array of String	修改后的加密套件列表。加密套件用于协商客户端与服务端之间的加密算法。 配置说明： - 加密套件的可选范围取决于所选的 TLS 协议版本（TLSVersions 参数）。 - 只要加密套件被任意一个已选 TLS 版本支持，即可添加到列表中。 - 若 TLSVersions 包含 TLSv1.3：可不指定 TLSv1.3 专属加密套件（系统将自动补全全部 TLSv1.3 套件）；若指定，则必须包含全部 TLSv1.3 专属加密套件，不支持仅指定部分。 获取可用加密套件： 请调用 DescribeSecurityPolicyCapabilities 接口查询各 TLS 版本支持的加密套件列表。 注意： 若不传此参数，则保持原有配置不变。 示例值：["TLS_AES_128_GCM_SHA256"]

参数名称	必选	类型	描述
DryRun	否	Boolean	是否仅执行预检请求。取值： • true：仅执行预检请求，不实际修改资源。预检请求将验证参数格式、权限及配置有效性等，帮助您在正式操作前发现潜在问题。 • false（默认）：执行正常请求，通过预检后将直接修改安全策略。 示例值：false
SecurityPolicyName	否	String	修改后的安全策略名称。用于标识和区分不同的安全策略。 命名规则： • 长度为 2~128 个字符。 • 必须以英文字母或中文开头。 • 可包含英文字母、中文、数字、半角句号（.）、下划线（_）和短划线（-）。 注意： 若不传此参数，则保持原有名称不变。 示例值：test-policy2
TLSVersions.N	否	Array of String	修改后的 TLS 协议版本列表。TLS（Transport Layer Security）协议用于保障客户端与负载均衡之间的通信安全。 可选值： • TLSv1.0：兼容性最好，但安全性较低，不推荐在生产环境使用。 • TLSv1.1：安全性略优于 TLSv1.0，但仍不推荐。 • TLSv1.2：目前主流的安全协议版本，兼顾安全性与兼容性。 • TLSv1.3：最新版本，安全性最高，性能更优，推荐优先使用。 注意： • 若不传此参数，则保持原有配置不变。 • 修改 TLS 版本时，请同步检查 Ciphers 参数的配置是否兼容。 示例值：["TLSv1.1", "TLSv1.2"]

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 修改安全策略

输入示例

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

<公共请求参数>

```
{
  "Ciphers": [
    "TLS_AES_128_GCM_SHA256"
  ],
  "SecurityPolicyId": "tls-t2ckyduq",
  "SecurityPolicyName": "test-policy2",
  "TLSVersions": [
    "TLSv1.1",
    "TLSv1.2"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
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。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

查询系统安全策略

最近更新时间：2026-07-07 10:58:02

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询系统安全策略。

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：DescribeSystemSecurityPolicies。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
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
SecurityPolicies	Array of SecurityPolicyInfo	系统安全策略列表。
TotalCount	Integer	安全策略总数。 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询系统安全策略

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "SecurityPolicies": [
      {
        "SecurityPolicyId": "tls-t2ckyduq",
        "SecurityPolicyName": "test-policy",
        "Status": "Active",
        "TLSVersions": [
          "TLSv1.2",
          "TLSv1.3"
        ],
        "Ciphers": [
          "TLS_AES_128_GCM_SHA256"
        ],
        "CreateTime": "2025-01-01T08:30:00+08:00",
        "Tags": []
      }
    ],
    "TotalCount": 1,
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

查询安全策略关联关系

最近更新时间：2026-07-07 10:58:03

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询安全策略的关联关系，即安全策略被哪些 HTTPS 监听器引用。在删除或修改安全策略前，建议先调用此接口确认影响范围。

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：DescribeSecurityPolicyRelations。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SecurityPolicyIds.N	是	Array of String	安全策略 ID 列表，ID 格式为 tls- 后接 8 位字母数字。 示例值：["tls-t2ckydug"]

3. 输出参数

参数名称	类型	描述
SecurityPolicyRelations	Array of SecurityPolicyRelations	安全策略关联的监听器列表。返回每个安全策略所关联的 HTTPS 监听器信息。
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 查询安全策略关联关系

输入示例

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

<公共请求参数>

```
{
  "SecurityPolicyIds": [
    "tls-t2ckyduq"
  ]
}
```

输出示例

```
{
  "Response": {
    "SecurityPolicyRelations": [
      {
        "SecurityPolicyId": "tls-t2ckyduq",
        "RelatedListeners": [
          {
            "LoadBalancerId": "alb-f8q2xk9m",
            "ListenerId": "lst-d9p3k7wa",
            "ListenerProtocol": "HTTPS",
            "ListenerPort": 443
          }
        ]
      }
    ],
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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

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

错误码	描述
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

获取支持的安全策略加密套件设置

最近更新时间：2026-07-07 10:58:03

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询当前地域支持的安全策略配置能力，包括可选的 TLS 协议版本及各版本对应的加密套件列表。在创建或修改自定义安全策略前，建议先调用此接口获取可用的配置选项。

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：DescribeSecurityPolicyCapabilities。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
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
SecurityPolicyCapabilities	Array of SecurityPolicyCapability	安全策略配置能力列表。返回当前地域支持的所有 TLS 版本及其对应的加密套件信息。 返回内容包含： - 支持的 TLS 协议版本（如 TLSv1.0、TLSv1.1、TLSv1.2、TLSv1.3）。 - 每个 TLS 版本支持的加密套件列表。 使用场景： - 在创建安全策略（CreateSecurityPolicy）前，调用此接口获取可选的加密套件。 - 在修改安全策略（ModifySecurityPolicyAttributes）前，确认新配置的有效性。
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

参数名称	类型	描述
		RequestId)。定位问题时需要提供该次请求的RequestId。

4. 示例

示例1 获取支持的安全策略加密套件设置

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "SecurityPolicyCapabilities": [
      {
        "TLSVersion": "TLSv1.3",
        "Ciphers": [
          "TLS_AES_128_GCM_SHA256"
        ]
      }
    ],
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

查询自定义安全策略

最近更新时间：2026-07-07 10:58:04

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询自定义安全策略列表，支持按安全策略 ID、名称或标签进行筛选，并支持分页查询。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeSecurityPolicies。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Filters.N	否	Array of Filter	过滤条件列表，用于筛选符合指定条件的安全策略。多个过滤条件之间为"与"关系。 支持的过滤条件： – SecurityPolicyNames：按安全策略名称筛选，支持模糊匹配。 – tag:tag-key：按标签键值对筛选，tag-key 请替换为实际的标签键。例如 tag:env 表示按标签键 env 筛选。 说明： 每个过滤条件最多支持 10 个取值。
MaxResults	否	Integer	单次请求返回的最大结果数。用于分页查询，与 NextToken 配合使用。 取值范围： 1~100。 默认值： 20。 示例值： 20
NextToken	否	String	分页查询的起始令牌。用于获取下一页结果数据。 使用说明： – 首次查询时无需设置此参数。 – 如果上一次查询返回了 NextToken，表示还有更多数据，请将该值传入此参数以获取下一页。

参数名称	必选	类型	描述
			– 若上一次查询未返回 NextToken 或返回为空，表示已是最后一页。 示例值：3974ae26e***
SecurityPolicyIds.N	否	Array of String	安全策略 ID 列表，ID 格式为 tls- 后接 8 位字母数字。 示例值：["tls-t2ckyduq"]

3. 输出参数

参数名称	类型	描述
NextToken	String	下一页查询的起始令牌。 – 若返回值不为空，表示还有更多数据，可将此值作为下一次请求的 NextToken 参数继续查询。 – 若返回值为空或未返回此字段，表示已是最后一页。 示例值：3974ae26e***
SecurityPolicies	Array of SecurityPolicyInfo	安全策略信息列表。包含每个安全策略的详细配置，如策略 ID、名称、TLS 版本、加密套件等。
TotalCount	Integer	符合过滤条件的安全策略总数。 说明： 此值表示满足查询条件的总记录数，而非本次返回的记录数。可用于计算分页信息。 示例值：3
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 查询自定义安全策略

输入示例

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

{
  "MaxResults": 20,
  "SecurityPolicyIds": [
    "tls-t2ckyduq"
```

```
]
}
```

输出示例

```
{
  "Response": {
    "NextToken": "",
    "SecurityPolicies": [
      {
        "SecurityPolicyId": "tls-t2ckydg",
        "SecurityPolicyName": "test-policy",
        "Status": "Active",
        "TLSVersions": [
          "TLSv1.2",
          "TLSv1.3"
        ],
        "Ciphers": [
          "TLS_AES_128_GCM_SHA256"
        ],
        "CreateTime": "2025-01-01T08:30:00+08:00",
        "Tags": []
      }
    ],
    "TotalCount": 1,
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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

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

错误码	描述
FailedOperation	操作失败。
InternalServerError	内部错误。
InvalidFilter	无效的过滤器
InvalidParameter	参数错误。
InvalidParameter.InvalidFieldValue	字段值错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

删除安全策略

最近更新时间：2026-07-07 10:58:04

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

删除一个或多个自定义安全策略。删除安全策略前，请确保该策略未被任何 HTTPS 监听器引用，否则删除操作将失败。

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：DeleteSecurityPolicy。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
SecurityPolicyIds.N	是	Array of String	安全策略 ID 列表，ID 格式为 tls- 后接 8 位字母数字。 示例值：["tls-t2ckyduq"]
DryRun	否	Boolean	是否仅执行预检请求。取值： - true ：仅执行预检请求，不实际删除资源。预检请求将验证参数格式、权限及安全策略是否被引用等，帮助您在正式操作前发现潜在问题。 - false （默认）：执行正常请求，通过预检后将直接删除安全策略。 示例值：false

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 删除安全策略

输入示例

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

<公共请求参数>

```
{
  "SecurityPolicyIds": [
    "tls-t2ckydug"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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

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

错误码	描述
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。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
ResourceInUse	资源被占用。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

创建自定义安全策略

最近更新时间：2026-07-09 01:10:11

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

创建自定义安全策略，用于配置 HTTPS 监听器的 TLS 协议版本和加密套件。通过安全策略，您可以灵活控制客户端与负载均衡之间 HTTPS 通信的安全级别。

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：CreateSecurityPolicy。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Ciphers.N	是	Array of String	安全策略支持的加密套件列表。加密套件用于协商客户端与服务端之间的加密算法。 配置说明： - 加密套件的可选范围取决于所选的 TLS 协议版本（TLSVersions 参数）。 - 只要加密套件被任意一个已选 TLS 版本支持，即可添加到列表中。 - 若 TLSVersions 包含 TLSv1.3：可不指定 TLSv1.3 专属加密套件（系统将自动补全全部 TLSv1.3 套件）；若指定，则必须包含全部 TLSv1.3 专属加密套件，不支持仅指定部分。 获取可用加密套件： 请调用 DescribeSecurityPolicyCapabilities 接口查询各 TLS 版本支持的加密套件列表。 示例值：["TLS_AES_128_GCM_SHA256"]
TLSVersions.N	是	Array of String	安全策略支持的 TLS 协议版本列表。TLS（Transport Layer Security）协议用于保障客户端与负载均衡之间的通信安全。 可选值： TLSv1.0：兼容性最好，但安全性较低，不推荐在生产环境使用。

参数名称	必选	类型	描述
			TLSv1.1: 安全性略优于 TLSv1.0, 但仍不推荐。 TLSv1.2: 目前主流的安全协议版本, 兼顾安全性与兼容性。 TLSv1.3: 最新版本, 安全性最高, 性能更优, 推荐优先使用。 建议: 生产环境建议至少选择 TLSv1.2, 若客户端支持, 优先启用 TLSv1.3。 示例值: ["TLSv1.2", "TLSv1.3"]
ClientToken	否	String	客户端幂等性令牌。 用于保证请求的幂等性, 防止因网络超时或客户端重试导致的重复创建。建议使用 UUID 作为令牌值。相同的 ClientToken 在有效期内重复请求时, 服务端将返回相同的结果。 示例值: d2e3f6c4-5b87-4f3e-8a19-7b9f9f3a8b45
DryRun	否	Boolean	是否仅执行预检请求。取值: true: 仅执行预检请求, 不实际创建资源。预检请求将验证参数格式、权限及资源配额等, 帮助您在正式操作前发现潜在问题。 false (默认): 执行正常请求, 通过预检后将直接创建安全策略。 示例值: false
SecurityPolicyName	否	String	安全策略名称。用于标识和区分不同的安全策略。 命名规则: - 长度为 2~128 个字符。 - 必须以英文字母或中文开头。 - 可包含英文字母、中文、数字、半角句号 (.)、下划线 (_) 和短划线 (-)。 建议: 使用具有业务含义的名称, 例如 "prod-high-security" 或 "测试环境策略"。 示例值: test-policy
Tags.N	否	Array of TagInfo	安全策略的标签列表。标签用于对资源进行分类和管理, 便于按业务、环境、部门等维度筛选和组织资源。 每个标签由键值对 (Key-Value) 组成, 同一资源下标签键不可重复。

3. 输出参数

参数名称	类型	描述
SecurityPolicyId	String	安全策略 ID, 格式为 tls- 后接 8 位字母数字。 示例值: tls-t2ckyduq

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 创建自定义安全策略

输入示例

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

{
  "Ciphers": [
    "TLS_AES_128_GCM_SHA256"
  ],
  "SecurityPolicyName": "test-policy",
  "TLSVersions": [
    "TLSv1.2",
    "TLSv1.3"
  ],
  "Tags": [
    {
      "TagKey": "key-xxx",
      "TagValue": "value-xxx"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "SecurityPolicyId": "tls-t2ckydug",
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
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。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameter.InvalidFieldValue	字段值错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。

错误码	描述
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

其他相关接口

设置负载均衡实例的安全组

最近更新时间：2026-07-09 01:10:11

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

SetLoadBalancerSecurityGroups 接口支持对一个公网负载均衡实例执行设置（绑定、解绑）安全组操作。查询一个负载均衡实例目前已绑定的安全组，可使用 [DescribeLoadBalancerDetail](#) 接口。本接口是set语义，绑定操作时，入参需要传入负载均衡实例要绑定的所有安全组（已绑定的+新增绑定的）。解绑操作时，入参需要传入负载均衡实例执行解绑后所绑定的所有安全组；如果要解绑所有安全组，可不传此参数，或传入空数组。默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：SetLoadBalancerSecurityGroups。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
SecurityGroups.N	是	Array of String	安全组 ID 列表。 示例值：["sg-0936o7sd", "sg-m2k9q7pa"]

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 从负载均衡实例解绑一个安全组

负载均衡实例原来绑定了两个安全组：sg-0936o7sd、sg-m2k9q7pa，执行本操作后，安全组 sg-m2k9q7pa 将从此负载均衡实例解绑。

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "SecurityGroups": [
    "sg-0936o7sd"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "00ca7fca-90f1-47fe-a724-5d7e96d04644"
  }
}
```

示例2 绑定多个安全组至负载均衡实例

负载均衡实例原来未绑定任何安全组，执行本操作后，安全组 sg-0936o7sd、sg-m2k9q7pa 将被绑定至此负载均衡实例。

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "SecurityGroups": [
    "sg-0936o7sd",
    "sg-m2k9q7pa"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "00ca7fca-90f1-47fe-a724-5d7e96d04633"
  }
}
```

示例3 解绑所有安全组

负载均衡实例原来已绑定若干个安全组，执行本操作后，所有安全组都将从此负载均衡实例解绑。

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "SecurityGroups": []
}
```

输出示例

```
{
  "Response": {
    "RequestId": "00ca7fca-90f1-47fe-a724-5d7e96d04655"
  }
}
```

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](#)。

通知负载均衡解绑后端服务

最近更新时间：2026-07-07 10:58:06

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

通知负载均衡解绑后端服务

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：NotifyUnbindTarget。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Ips.N	是	Array of String	后端服务的IP列表。 > VpcId（NumericVpcId）、Ips必须同时设置。 示例值：["192.168.0.1"]
NumericVpcId	是	Integer	后端服务所属VPC的数字ID。 > VpcId（NumericVpcId）、Ips必须同时设置。 示例值：143563231

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 通知负载均衡解绑后端服务

输入示例


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

```
{
  "Ips": [
    "192.168.0.1"
  ],
  "NumericVpcId": 143563231
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

设置负载均衡实例操作保护

最近更新时间：2026-07-09 01:10:12

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

设置负载均衡实例修改保护。

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：ModifyLoadBalancerModificationProtection。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
ModificationProtectionEnabled	是	Boolean	是否开启修改保护。开启后，可防止实例被意外修改或删除。 - true：开启修改保护 - false：关闭修改保护 示例值：true
DryRun	否	Boolean	是否只预检此次请求。取值： - true：仅执行预检，不实际操作资源。检查参数完整性、请求格式及业务限制，通过返回 DryRunOperation，不通过返回对应错误。 - false（默认）：执行正常请求，检查通过后直接操作资源。 示例值：false
Reason	否	String	开启修改保护的原因说明。 长度为 1~255 个字符，必须是中文和无害字符串中的字符，可包含中文、字母、数字、短划线（-）、正斜线（/）、半角句号（.）、下划线（_）。 示例值：生产环境实例，防止误操作

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 设置负载均衡实例操作保护

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "ModificationProtectionEnabled": true,
  "Reason": "prod_protection"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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

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

错误码	描述
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
RegionError	地域错误
ResourceNotFound	资源不存在。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

创建负载均衡实例询价

最近更新时间：2026-07-07 10:58:06

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

InquirePriceCreateLoadBalancer接口查询创建负载均衡的价格。

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：InquirePriceCreateLoadBalancer。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ChargeType	否	String	实例的计费类型。默认POSTPAID_BY_HOUR，仅取值 POSTPAID_BY_HOUR：表示按量计费。 示例值：POSTPAID_BY_HOUR

3. 输出参数

参数名称	类型	描述
Price	Price	询价结果。
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 创建负载均衡实例询价

输入示例

```
POST / HTTP/1.1
Host: alb.tencentcloudapi.com
```

```
Content-Type: application/json
X-TC-Action: InquirePriceCreateLoadBalancer
<公共请求参数>

{}
```

输出示例

```
{
  "Response": {
    "Price": {
      "InstancePrice": {
        "Discount": 100,
        "UnitPrice": 0.2,
        "UnitPriceDiscount": 0.2
      },
      "LcuPrice": {
        "Discount": 100,
        "UnitPrice": 0.049,
        "UnitPriceDiscount": 0.049
      }
    },
    "RequestId": "255917ce-96f9-46f6-9c13-8f498469e824"
  }
}
```

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](#)。

查询可用区信息

最近更新时间：2026-07-07 10:58:07

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询可用区

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：DescribeZones。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
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
Zones	Array of Zone	可用区列表
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 查询所有可用区

查询所有可用区

输入示例

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

```
X-TC-Action: DescribeZones
```

```
<公共请求参数>
```

```
{}
```

输出示例

```
{
  "Response": {
    "Zones": [
      {
        "ZoneId": "ap-guangzhou-2",
        "LocalName": "广州二区",
        "ZoneStatus": "available"
      },
      {
        "ZoneId": "ap-guangzhou-3",
        "LocalName": "广州三区",
        "ZoneStatus": "available"
      }
    ],
    "RequestId": "4e5df572-e085-4e30-a3bb-d520768d1473"
  }
}
```

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](#)。

查询配额

最近更新时间：2026-07-07 10:58:07

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询当前账号的 ALB 配额配置。支持按配额类型查询，也支持传入资源ID查询资源级配额；可通过 DisplayFields 按需返回已使用量和剩余可用量。

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：DescribeQuota。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
QuotaTypes.N	是	Array of String	配额类型列表。支持同时传入多个配额类型。查询资源级配额时，可配合 ResourceIds 传入对应资源ID；如需返回已使用量和可用量，可在 DisplayFields 中传入 used、available。 枚举说明： - alb_quota_loadbalancers_num：每个地域可创建的 ALB 实例数。 - alb_quota_targetgroups_num：每个地域可创建的 ALB 目标组数。 - alb_quota_loadbalancer_listeners_num：每个 ALB 实例可创建的监听器数，ResourceIds 填写 ALB 实例 ID。 - alb_quota_loadbalancer_rules_num：每个 ALB 实例可添加的转发规则数，不计入默认规则，ResourceIds 填写 ALB 实例 ID。 - alb_quota_loadbalancer_certificates_num：每个 ALB 实例可添加的扩展证书数，不计入默认证书，ResourceIds 填写 ALB 实例 ID。 - alb_quota_loadbalancer_targetgroup_num：每个 ALB 实例可绑定的目标组数，ResourceIds 填写 ALB 实例 ID。 - alb_quota_loadbalancer_servers_num：每个 ALB 实例可添加的后端服务器数，ResourceIds 填写 ALB 实例 ID。 - alb_quota_server_added_num：单个后端服务器 IP 可被添加到 ALB 后端目标组的次数。 - alb_quota_targetgroup_attached_num：每个目标组可被 ALB 转发规则关联的次数，ResourceIds 填写目标组 ID。 - alb_quota_targetgroup_targets_num：每个目标组支持的后端服务器数，适

参数名称	必选	类型	描述
			用于 IP 和端口类型后端，ResourceIds 填写目标组 ID。 - alb_quota_targetgroup_targets_num_scf: 每个目标组支持的 SCF 函数后端数，ResourceIds 填写目标组 ID。 - alb_quota_max_request_timeout: 创建监听器时可配置的连接请求最大超时时间。 - alb_quota_max_idle_timeout: 创建监听器时可配置的连接空闲最大超时时间。 - alb_quota_listener_certificates_num: 单个监听器可添加的证书数量，ResourceIds 填写监听器 ID。 - alb_quota_rule_targetgroups_num: 单条转发规则可绑定的目标组数量。 - alb_quota_rule_conditions_num: 单条转发规则可添加的匹配条件条目数。 - alb_quota_rule_wildcards_num: 单条转发规则可添加的包含通配符的匹配条目数。 - alb_quota_rule_actions_num: 单条转发规则可添加的动作条目数。 - alb_quota_cipher_template_listeners_num: 单个加密套件模板可关联的监听器数量。 - alb_quota_healthcheck_templates_num: 每个地域可创建的健康检查模板数。 - alb_quota_securitygroup_templates_num: 单个 ALB 实例支持绑定的安全组数量。 - alb_quota_securitygroup_rules_per_sg_num: 单个 ALB 实例中单个安全组支持的规则条目数。 - alb_quota_security_policies_num: 每个地域可创建的自定义安全策略数。 示例值: ["alb_quota_loadbalancer_listeners_num"]
DisplayFields.N	否	Array of String	显示字段列表，用于控制是否额外返回用量信息。支持 used、available: used 表示返回当前已使用量，available 表示返回当前剩余可用量。QuotaType 和 Limit 总是返回；ResourceId 会在请求传入 ResourceIds 时返回。 示例值: ["used", "available"]
ResourceIds.N	否	Array of String	资源ID列表。用于查询具体资源维度的配额和用量；不传时查询账号或地域维度的默认配额配置。资源ID的类型由 QuotaTypes 决定，例如 ALB 实例级配额填写 ALB 实例 ID，监听器级配额填写监听器 ID，目标组级配额填写目标组 ID。 示例值: ["alb-f8q2xk9m"]

3. 输出参数

参数名称	类型	描述
Quotas	Array of QuotaInfo	配额列表。每个元素表示一个配额类型的查询结果；当请求传入 ResourceIds 时，每个元素表示一个配额类型和一个资源ID组合的查询结果。
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 查询每个地域可创建的 ALB 实例数

输入示例

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

```
{
  "QuotaTypes": [
    "alb_quota_loadbalancers_num"
  ]
}
```

输出示例

```
{
  "Response": {
    "Quotas": [
      {
        "Available": null,
        "Limit": 60,
        "QuotaType": "alb_quota_loadbalancers_num",
        "ResourceId": "alb1cu-e8p3x9mq",
        "Used": null
      }
    ],
    "RequestId": "c6585b63-353a-48cb-93ec-b23321e3b7ea"
  }
}
```

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](#)。

查询异步任务信息

最近更新时间：2026-07-07 10:58:07

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

异步任务查询接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeAsyncJobs。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
MaxResults	否	Integer	分批次查询时每次显示的条目数。取值范围：1~100，默认值：20。 示例值：20
NextToken	否	String	是否拥有下一次查询的令牌（Token）。取值：第一次查询和没有下一次查询时，均无需填写。如果有下一次查询，取值为上一次 API 调用返回的 NextToken 值。 示例值：FFmyTO70tTpLG6l3FmYAXGKPd8r6k
RequestIds.N	否	Array of String	异步请求返回的RequestId列表 示例值：["a5c5d5d5-d5d5-4d5d-8d5d-d5d5d5d5d5d5"]

3. 输出参数

参数名称	类型	描述
Jobs	Array of Job	任务列表。
MaxResults	Integer	分批次查询时每次显示的条目数。 示例值：20
NextToken	String	是否拥有下一次查询的令牌（Token）。取值：如果 NextToken 为空表示没有下一次查询。如果 NextToken 有返回值，该取值表示下一次查询开始的令牌。 示例值：FFmyTO70tTpLG6l3FmYAXGKPd8r6k

参数名称	类型	描述
TotalCount	Integer	列表条目数。 示例值：100
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 查询异步任务信息

输入示例

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

{
  "RequestIds": [
    "c318a537-9bfa-40c2-b6d5-e6f872ad4b1f"
  ]
}
```

输出示例

```
{
  "Response": {
    "Jobs": [
      {
        "ApiName": "DisassociateBandwidthPackageFromLoadBalancer",
        "RequestId": "c318a537-9bfa-40c2-b6d5-e6f872ad4b1f",
        "ResourceIds": [
          "alb-f8q2xk9m"
        ],
        "Status": "Succeeded"
      }
    ],
    "MaxResults": 20,
    "NextToken": "",
    "TotalCount": 1,
    "RequestId": "e4bf85e4-739f-4a3e-9ce1-d98d1fa2a876"
  }
}
```

```
}  
  
}
```

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

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

错误码	描述
FailedOperation	操作失败。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

共享带宽包相关接口

实例解绑共享带宽包

最近更新时间：2026-07-07 10:58:08

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

将共享带宽包从应用型负载均衡实例解绑。

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：DisassociateBandwidthPackageFromLoadBalancer。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
BandwidthPackageId	是	String	共享带宽包 ID。 示例值：bwp-q8m2x4pa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
ClientToken	否	String	客户端Token，用于保证请求的幂等性。 从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken 只支持ASCII字符。 > 若您未指定，则系统自动使用API请求的RequestId作为ClientToken标识。每次API请求的RequestId不一样。 示例值：7a9d4f63-c8e2-42b1-9f05-d6a7c3e8b241
DryRun	否	Boolean	是否只预检此次请求。取值： - true：发送检查请求，不会将共享带宽包从负载均衡实例中移除。检查项包括是否填写了必需参数、请求格式、业务限制。如果检查不通过，则返回对应错误。如果检查通过，则返回错误码DryRunOperation。 - false（默认值）：发送正常请求，通过检查后返回HTTP 2xx状态码并直接

参数名称	必选	类型	描述
			进行操作。 示例值：true

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 实例解绑共享带宽包

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "BandwidthPackageId": "bwp-q8m2x4pa"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "c318a537-9bfa-40c2-b6d5-e6f872ad4b1f"
  }
}
```

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](#)。

实例绑定共享带宽包

最近更新时间：2026-07-07 10:58:08

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

将共享带宽包绑定到应用型负载均衡实例。

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：AssociateBandwidthPackageWithLoadBalancer。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
BandwidthPackageId	是	String	共享带宽包 ID。 示例值：bwp-q8m2x4pa
LoadBalancerId	是	String	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
ClientToken	否	String	客户端Token，用于保证请求的幂等性。 从您的客户端生成一个参数值，确保不同请求间该参数值唯一。ClientToken 只支持ASCII字符。 > 若您未指定，则系统自动使用API请求的RequestId作为ClientToken标识。每次API请求的RequestId不一样。 示例值：8c5f6a42-9d3e-4b71-a2c8-f0e9d7b6a5c4
DryRun	否	Boolean	是否只预检此次请求。取值： - true：发送检查请求，不会将共享带宽包绑定到负载均衡实例。检查项包括是否填写了必需参数、请求格式、业务限制。如果检查不通过，则返回对应错误。如果检查通过，则返回错误码DryRunOperation。 - false（默认值）：发送正常请求，通过检查后返回HTTP 2xx状态码并直接进行操作。 示例值：true

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 实例绑定共享带宽包

输入示例

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

{
  "LoadBalancerId": "alb-f8q2xk9m",
  "BandwidthPackageId": "bwp-q8m2x4pa"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "aa1ca3f7-f928-4cb3-b6e1-a75b684225ab"
  }
}
```

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](#)。

健康检查模板相关接口

修改健康检查模板

最近更新时间：2026-07-15 01:09:04

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

修改健康检查模板

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：ModifyHealthCheckTemplate。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
HealthCheckTemplateId	是	String	健康检查模板 ID，格式为 hct- 后接字母数字。 示例值：hct-m4x8q2pa
DryRun	否	Boolean	是否预览此次请求。 false（默认）：发送普通请求，直接修改健康检查模板。 true：发送预览请求，检查修改健康检查模板的参数、格式、业务限制等是否符合要求。 示例值：true
HealthCheckCodes.N	否	Array of String	健康检查状态码。取值： - 当健康检查协议为HTTP/HTTPS时： - http_1xx - http_2xx（默认值） - http_3xx - http_4xx - http_5xx

参数名称	必选	类型	描述
			- 当健康检查协议为GRPC/GRPCS时：默认值为12，数值范围为0-99，输入值可为数值、多个数值或者范围以及相互组合，如： "20" "0-99" 示例值：["http_1xx"]
HealthCheckHealthyThreshold	否	Integer	判定后端服务健康的阈值，当健康检查连续成功多少次后，后端服务的状态由 不健康 变为 健康 。 取值范围： 2-10 。 默认值： 2 。 示例值：2
HealthCheckHost	否	String	健康检查域名。 长度限制为 1-255 个字符。 可包含小写字母、数字、短划线（-）和半角句号（.）。 仅当 HealthCheckProtocol 设置为 HTTP/HTTPS/GRPC/GRPCS 时，该参数生效。 示例值：example.com
HealthCheckHttpVersion	否	String	健康检查 HTTP 协议版本，取值： HTTP1.1（默认） HTTP1.0 仅当HealthCheckProtocol设置为HTTP 或 HTTPS 时，该参数生效。 示例值：HTTP1.1
HealthCheckInterval	否	Integer	健康检查的时间间隔。单位：秒。取值范围： 2-300 。默认值： 5 。 示例值：10
HealthCheckMethod	否	String	健康检查方法，取值：- GET - HEAD （默认值） 仅当HealthCheckProtocol设置为HTTP 或 HTTPS 时，该参数生效。

参数名称	必选	类型	描述
			示例值：GET
HealthCheckPath	否	String	健康检查的转发规则路径。长度为 1-80 个字符，只能使用字母、数字、字符 <code>-/.%?#&=</code> 以及扩展字符 <code>_;~!()*[]@\$^:'',+。</code> URL 必须以正斜线 (<code>/</code>) 开头。 仅当HealthCheckProtocol为HTTP/HTTPS/GRPC/GRPCS时，转发规则路径参数生效。 示例值：/hello/index.html
HealthCheckPort	否	Integer	健康检查访问后端服务器的端口。取值范围：0-65535。默认值：0，表示后端服务器的端口。 示例值：80
HealthCheckProtocol	否	String	健康检查协议。取值： HTTP（默认）：通过发送 HEAD 或 GET 请求模拟浏览器的访问行为来检查服务器应用是否健康。 HTTPS：通过发送 HEAD 或 GET 请求模拟浏览器的访问行为来检查服务器应用是否健康。（数据加密，相比 HTTP 更安全。） TCP：通过发送 SYN 握手报文来检测服务器端口是否存活。 GRPC：通过发送 POST 或 GET 请求来检查服务器应用是否健康。 GRPCS：通过发送 POST 或 GET 请求来检查服务器应用是否健康。 示例值：HTTP
HealthCheckTemplateName	否	String	健康检查模板名称。长度为 1-255 个字符，可包含数字、大小写字母、中文、半角句号 (<code>.</code>)、下划线 (<code>_</code>) 和短划线 (<code>-</code>)。 示例值：lbtg-0zrnc9qa-template
HealthCheckTimeout	否	Integer	健康检查的响应超时时间。单位：秒。 取值范围：2-60。 默认值：2。 示例值：2
HealthCheckUnhealthyThreshold	否	Integer	判定后端服务不健康的阈值，当健康检查连续失败多少次后，后端服务的状态由健康变为不健康。 取值范围：2-10。 默认值：2。

参数名称	必选	类型	描述
			示例值：2

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 修改健康检查模板的名称

输入示例

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

{
  "HealthCheckTemplateId": "hct-fb7tuf7o",
  "HealthCheckTemplateName": "修改名称"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "e109ac6e-a12b-4fb2-885e-e63d63c4d883"
  }
}
```

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

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
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。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RegionError	地域错误
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。

错误码	描述
UnsupportedOperation	操作不支持。

查询健康检查模板列表

最近更新时间：2026-07-09 01:10:15

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

查询健康检查模板列表

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：DescribeHealthCheckTemplates。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Filters.N	否	Array of Filter	过滤器。通过指定的过滤条件来查询健康检查模板，支持： - Name的值为HealthCheckTemplateName。通过名称来筛选健康检查模板。Values的值为模板名称列表。 - Name的值为HealthCheckProtocol。通过健康检查协议来筛选健康检查模板。Values的值为协议列表。 - 通过标签方式筛选。
HealthCheckTemplateIds.N	否	Array of String	健康检查模板 ID 列表，ID 格式为 hct- 后接字母数字。 示例值：["hct-m4x8q2pa"]
MaxResults	否	String	返回列表的数量，默认为20，最大值为100。 示例值：20
NextToken	否	String	下一次查询的Token值。第一次查询和没有下一次查询时，无需填写。 如果有下一次查询，取值为上一次 API 调用返回的 NextToken 值。 示例值：3974ae26e***

3. 输出参数

参数名称	类型	描述
HealthCheckTemplates	Array of HealthCheckTemplate	健康检查模板列表。
NextToken	String	下一次查询的Token值，如果当前是最后一页，返回为空。 示例值：3974ae26e***
TotalCount	Integer	经过筛选后查询到的健康检查模板总数。 示例值：1
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询健康检查模板列表

输入示例

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

{
  "HealthCheckTemplateIds": [
    "hct-m4x8q2pa"
  ],
  "MaxResults": "20"
}
```

输出示例

```
{
  "Response": {
    "HealthCheckTemplates": [
      {
        "HealthCheckCodes": [
          "http_1xx"
        ],

```



```
"HealthCheckHealthyThreshold": 2,
"HealthCheckHost": "example.com",
"HealthCheckHttpVersion": "HTTP1.1",
"HealthCheckInterval": 10,
"HealthCheckMethod": "GET",
"HealthCheckPath": "/hello/index.html",
"HealthCheckPort": 80,
"HealthCheckProtocol": "HTTP",
"HealthCheckTemplateId": "hct-m4x8q2pa",
"HealthCheckTemplateName": "target-group-health-check",
"CreateTime": "2025-01-01T08:30:00+08:00",
"ModifyTime": "2025-01-01T08:30:00+08:00",
"HealthCheckTimeout": 2,
"HealthCheckUnhealthyThreshold": 2,
"Tags": [
{
"TagKey": "key-xxx",
"TagValue": "value-xxx"
}
],
"NextToken": "3974ae26e***",
"TotalCount": 1,
"RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
}
```

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](#)。

删除健康检查模板

最近更新时间：2026-07-09 01:10:15

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

删除健康检查模板

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：DeleteHealthCheckTemplates。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
HealthCheckTemplateIds.N	是	Array of String	健康检查模板 ID 列表，ID 格式为 hct- 后接字母数字。 示例值：["hct-p7q4x2ma"]
DryRun	否	Boolean	是否预览此次请求。 - false （默认）：发送普通请求，直接删除模板。 - true ：发送预览请求，检查删除模板的参数、格式、业务限制等是否符合要求。 示例值：true

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 删除健康检查模板

输入示例

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

<公共请求参数>

```
{
  "HealthCheckTemplateIds": [
    "hct-p7q4x2ma"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
  }
}
```

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](#)。

创建健康检查模板

最近更新时间：2026-07-07 10:58:10

1. 接口描述

接口请求域名：alb.tencentcloudapi.com。

创建健康检查模板

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：CreateHealthCheckTemplate。
Version	是	String	公共参数 ，本接口取值：2025-10-30。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
DryRun	否	Boolean	是否预览此次请求。 – false （默认）：发送普通请求，直接修改健康检查模板。 – true ：发送预览请求，检查修改健康检查模板的参数、格式、业务限制等是否符合要求。 示例值：true
HealthCheckCodes.N	否	Array of String	健康检查状态码。取值： – 当健康检查协议为HTTP/HTTPS时： – http_1xx – http_2xx （默认值） – http_3xx – http_4xx – http_5xx – 当健康检查协议为GRPC/GRPCS时：默认值为12，数值范围为0-99，输入值可为数值、多个数值或者范围以及相互组合，如： – "20" – "0-99" 示例值：["http_1xx"]
HealthCheckHealthyThreshold	否	Integer	判定后端服务健康的阈值，当健康检查连续成功多少次后，后端服务的状态由不健康变为健康。 取值范围：2-10。

参数名称	必选	类型	描述
			默认值：2。 示例值：2
HealthCheckHost	否	String	健康检查域名。 长度限制为 1-255 个字符。 可包含小写字母、数字、短划线（-）和半角句号（.）。 > 仅当 HealthCheckProtocol 设置为 HTTP/HTTPS/GRPC/GRPCS 时，该参数生效。 示例值：example.com
HealthCheckHttpVersion	否	String	健康检查 HTTP 协议版本，取值： - HTTP1.1（默认） - HTTP1.0 > 仅当 HealthCheckProtocol 设置为 HTTP 或 HTTPS 时，该参数生效。 示例值：HTTP1.1
HealthCheckInterval	否	Integer	健康检查的时间间隔。单位：秒。取值范围：2-300。默认值：5。 示例值：10
HealthCheckMethod	否	String	健康检查方法，取值：- GET - HEAD（默认值） > 仅当 HealthCheckProtocol 设置为 HTTP 或 HTTPS 时，该参数生效。 示例值：GET
HealthCheckPath	否	String	健康检查的转发规则路径。长度为 1-80 个字符，只能使用字母、数字、字符 - / . % ? # & = 以及扩展字符 _ ; ~ ! () * [] @ \$ ^ : ' , + 。 URL 必须以正斜线 (/) 开头。 > 仅当 HealthCheckProtocol 为 HTTP/HTTPS/GRPC/GRPCS 时，转发规则路径参数生效。 示例值：/hello/index.html
HealthCheckPort	否	Integer	健康检查访问后端服务器的端口。取值范围：0-65535。默认值：0，表示后端服务器的端口。 示例值：80
HealthCheckProtocol	否	String	健康检查协议。取值： - HTTP（默认）：通过发送 HEAD 或 GET 请求模拟浏览器的访问行为来检查服务器应用是否健康。 - HTTPS：通过发送 HEAD 或 GET 请求模拟浏览器的访问行为来检查服务器应用是否健康。（数据加密，相比 HTTP 更安全。） - TCP：通过发送 SYN 握手报文来检测服务器端口是否存活。 - GRPC：通过发送 POST 或 GET 请求来检查服务器应用是否健康。 - GRPCS：通过发送 POST 或 GET 请求来检查服务器应用是否健康。 示例值：HTTP

参数名称	必选	类型	描述
HealthCheckTemplateName	否	String	健康检查模板名称。长度为 1-255 个字符，可包含数字、大小写字母、中文、半角句号（.）、下划线（_）和短划线（-）。 示例值：lbtg-0zrnc9qa-template
HealthCheckTimeout	否	Integer	健康检查的响应超时时间。单位：秒。 取值范围：2-60。 默认值：2。 示例值：2
HealthCheckUnhealthyThreshold	否	Integer	判定后端服务不健康的阈值，当健康检查连续失败多少次后，后端服务的状态由健康变为不健康。 取值范围：2-10。 默认值：2。 示例值：2
Tags.N	否	Array of TagInfo	标签。

3. 输出参数

参数名称	类型	描述
HealthCheckTemplateId	String	健康检查模板 ID，格式为 hct- 后接字母数字。所有接口（创建、查询、修改、删除）均使用 hct- 前缀。 示例值：hct-m4x8q2pa
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 创建健康检查模板

输入示例

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

{
  "HealthCheckCodes": [
    "http_1xx"
  ],
  "HealthCheckHealthyThreshold": 2,
  "HealthCheckHost": "example.com",
  "HealthCheckHttpVersion": "HTTP1.1",
```



```
"HealthCheckInterval": 10,
"HealthCheckMethod": "GET",
"HealthCheckPath": "/hello/index.html",
"HealthCheckPort": 80,
"HealthCheckProtocol": "HTTP",
"HealthCheckTemplateName": "lbtg-0zrnc9qa-template",
"HealthCheckTimeout": 2,
"HealthCheckUnhealthyThreshold": 2,
"Tags": [
{
"TagKey": "key-xxx",
"TagValue": "value-xxx"
}
]
}
```

输出示例

```
{
"Response": {
"HealthCheckTemplateId": "hct-m4x8q2pa",
"RequestId": "3b848733-70e5-4558-ae39-4b9938eb7609"
}
}
```

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：2026-07-15 01:09:05

AccessLogConfig

访问日志配置。

被如下接口引用：DescribeLoadBalancerDetail, DescribeLoadBalancers。

名称	类型	必选	描述
LogSetId	String	否	负载均衡日志服务(CLS)的日志集ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：3e727335-f94c-468f-9989-21759a3b6de8
LogTopicId	String	否	负载均衡日志服务(CLS)的日志主题ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：3e727335-f94c-468f-9989-21759a3b6de8

CertificateInfo

证书信息

被如下接口引用：DescribeListenerCertificates。

名称	类型	描述
AssociatedTime	Timestamp ISO8601	证书绑定时间。 示例值：2024-06-12T10:34:56Z
CertificateId	String	证书 ID。 示例值：Fm8Wp7x8
CertificateType	String	证书类型。取值：CA或SVR（服务器证书）。 示例值：SVR
IsDefault	Boolean	是否为监听器默认证书。取值： true：默认证书。 false：扩展证书。 示例值：true
Status	String	证书与监听器的绑定状态。取值：Associated（已关联）、Associating（关联中）、Disassociating（解除关联中）、Error（异常）。 示例值：Associated

DefaultAction

监听器默认规则动作

被如下接口引用：CreateListener, DescribeListenerDetail, ModifyListenerAttributes。

名称	类型	必选	描述
TargetGroupConfig	TargetGroupConfig	是	转发目标组配置。创建监听器时转发动作中的目标组配置仅支持单个目标组。
Type	String	是	转发动作类型。创建监听器时，默认转发动作类型仅支持转发至目标组。 示例值：TargetGroup

DeletionProtectionConfig

删除保护状态信息。

被如下接口引用：CreateLoadBalancer, DescribeLoadBalancerDetail, DescribeLoadBalancers, ModifyLoadBalancerAttributes。

名称	类型	必选	描述
DeletionProtectionEnabled	Boolean	否	是否开启删除保护。开启后，可防止实例被意外删除。 – true：开启删除保护 – false：关闭删除保护 示例值：false
Reason	String	否	开启修改保护的原因说明。 长度为 1~255 个字符，必须是中文和无害字符串中的字符，可包含中文、字母、数字、短划线（-）、正斜线（/）、半角句号（.）、下划线（_）。 示例值：防止误操作

Filter

过滤器条件

被如下接口引用：DescribeHealthCheckTemplates, DescribeListeners, DescribeLoadBalancers, DescribeRules, DescribeSecurityPolicies, DescribeTargetGroupTargets, DescribeTargetGroups。

名称	类型	必选	描述
Name	String	是	过滤器的名称 示例值：tag-key
Values	Array of String	是	过滤器的值数组 示例值：["1.0"]

FixedResponseInfo

固定应答信息

被如下接口引用：CreateRules, DescribeRules, ModifyRulesAttributes。

名称	类型	必选	描述
HttpCode	Integer	是	返回的HTTP响应码，支持 2xx、4xx、5xx。 示例值：200
Content	String	否	返回的固定内容。只支持 ASCII 字符，最大1KB。 示例值：OK
ContentType	String	否	返回固定内容的格式。 取值：text/plain、text/css、text/html、application/javascript或application/json。 示例值：text/plain

HTTPCookieInfo

HTTP Cookie信息

被如下接口引用：CreateRules, DescribeRules, ModifyRulesAttributes。

名称	类型	必选	描述
Key	String	是	Cookie的键，长度1~64个字符，支持字母、数字、下划线。 示例值：ClientVer
Value	String	是	Cookie的值，长度1~128个字符，支持可打印字符。 示例值：1.0

HTTPHeaderInfo

HTTP Header信息

被如下接口引用：CreateRules, DescribeRules, ModifyRulesAttributes。

名称	类型	必选	描述
Key	String	是	HTTP Header的键，长度1~40个字符，支持的字符集为：a-z A-Z 0-9 - _。 不支持中文，不支持Host, Cookie。 示例值：ClientVer
Values	Array of String	是	HTTP Header的值，长度1~128个字符，支持可打印字符。 不支持"，开头和结尾不能是空格，结尾不能是\。 示例值：["1.0"]

HTTPQueryStringInfo

HTTP查询字符串信息

被如下接口引用：CreateRules, DescribeRules, ModifyRulesAttributes。

名称	类型	必选	描述
Key	String	是	查询字符串的键，长度1~16个字符。支持可打印字符，不支持空格和#[] <>&。 支持 * 多字符通配，? 单字符通配。

名称	类型	必选	描述
			示例值：ClientVer
Value	String	是	查询字符串的值，长度1～128字符，支持可打印字符，不支持空格和#[]{} <>&。 支持 * 多字符通配，? 单字符通配。 示例值：1.0

HTTPRedirectInfo

HTTP重定向信息

被如下接口引用：CreateRules, DescribeRules, ModifyRulesAttributes。

名称	类型	必选	描述
HttpCode	Integer	是	重定向的HTTP码，支持301、302、303、307、308。 示例值：200
Host	String	否	重定向的主机地址，默认值\${host}。长度3～128个字符，支持的字符集为：a-z 0-9 _ . -。 示例值：example.com
Path	String	否	重定向的路径，默认值\${path}。长度1～128个字符，支持的字符集为：a-z A-Z 0-9 ? = _ . - / : 。 示例值：/hello/index.html
Port	String	否	重定向的端口，默认值 \${port}。取值1～65535。 示例值：80
Protocol	String	否	重定向的协议，取值：HTTP,HTTPS，默认值\${protocol}。 示例值：HTTP
Query	String	否	重定向的查询字符串，默认值\${query}。长度1～128字符，支持可打印字符，不支持 #[]{} <>& 和空格。 示例值：\${query}

HTTPRewriteInfo

HTTP重写信息

被如下接口引用：CreateRules, DescribeRules, ModifyRulesAttributes。

名称	类型	必选	描述
Host	String	否	重写的主机地址，默认值\${host}。长度3 ~ 128个字符，支持的字符集为：a-z 0-9 _ . -。 示例值：example.com
Path	String	否	重写的路径，默认值\${path}。长度1 ~ 128个字符，支持的字符集为：a-z A-Z 0-9 ? = _ . - / :。 示例值：/hello/index.html
Query	String	否	重写的查询字符串，默认值\${query}。长度1 ~ 128字符，支持可打印字符，不支持 #[]{} <>& 和空格。 示例值：\${query}

HealthCheckConfig

健康检查配置

被如下接口引用：CreateTargetGroup, DescribeTargetGroups, ModifyTargetGroupAttributes。

名称	类型	必选	描述
HealthCheckEnabled	Boolean	是	是否开启健康检查。 - true : 开启。 - false : 不开启。 示例值：true
HealthCheckCodes	Array of String	否	健康检查状态码。取值： - 当健康检查协议为HTTP/HTTPS时： - http_1xx - http_2xx（默认值） - http_3xx - http_4xx - http_5xx - 当健康检查协议为gRPC时：默认值为12，数值范围为0-99，输入值可为数值、多个数值或者范围以及相互组合，如： - "20" - "0-99" > 仅当HealthCheckProtocol设置为HTTP、HTTPS、GRPC 或者GRPCS时，该参数生效。 示例值：["http_2xx"]
HealthCheckHealthyThreshold	Integer	否	判定后端服务健康的阈值，当健康检查连续成功多少次后，后端服务的状态由不健康变为健康。 取值范围：2-10。 默认值：2。 示例值：2
HealthCheckHost	String	否	健康检查域名。该参数不设置时默认使用后端服务的内网IP作为健康检查地址。 域名限制：

名称	类型	必选	描述
			– 长度限制为 1-255 个字符。 – 可包含小写字母、数字、短划线 (-) 和半角句号 (.)。 – 至少包含一个半角句号 (.)，半角句号 (.) 不能出现在开头或结尾。 – 最右侧的域标签，只能包含字母，不能包含数字或短划线 (-)。 – 短划线 (-) 不能出现在开头或结尾。 > 仅当 HealthCheckProtocol 设置为 HTTP、HTTPS、GRPC、GRPCS 时，该参数生效。 示例值：example.com
HealthCheckHttpVersion	String	否	健康检查 HTTP 协议版本，取值： – HTTP1.1 (默认) – HTTP1.0 > 仅当 HealthCheckProtocol 设置为 HTTP 或 HTTPS 时，该参数生效。 示例值：HTTP1.1
HealthCheckInterval	Integer	否	健康检查的时间间隔。单位：秒。 取值范围：2-300。 默认值：5。 示例值：10
HealthCheckMethod	String	否	健康检查方法，取值： – GET – HEAD (默认值) > 仅当 HealthCheckProtocol 设置为 HTTP 或 HTTPS 时，该参数生效。 示例值：GET
HealthCheckPath	String	否	健康检查的转发规则路径。 长度为 1~80 个字符，只能使用字母、数字、字符 - / . % ? # & = 以及扩展字符 _ ; ~ ! () * [] @ \$ ^ : ' , + 。 URL 必须以正斜线 (/) 开头。 > 仅当 HealthCheckProtocol 为 HTTP、HTTPS、GRPC、GRPCS 时，转发规则路径参数生效。 示例值：/hello/index.html
HealthCheckPort	Integer	否	健康检查访问后端服务器的端口。 取值范围：0-65535。 默认值：0，表示后端服务器的端口。 示例值：80
HealthCheckProtocol	String	否	健康检查协议。取值： – HTTP (默认)：通过发送 HEAD 或 GET 请求模拟浏览器的访问行为来检查服务器应用是否健康。 – HTTPS：通过发送 HEAD 或 GET 请求模拟浏览器的访问行为来检查服务器应用是否健康。（数据加密，相比 HTTP 更安全。） – TCP：通过发送 SYN 握手报文来检测服务器端口是否存活。 – GRPC：通过发送 POST 请求来检查服务器应用是否健

名称	类型	必选	描述
			康。 – GRPCS ：通过发送 POST 请求来检查服务器应用是否健康。 示例值：HTTP
HealthCheckTimeout	Integer	否	健康检查的响应超时时间。单位：秒。 取值范围： 2–60 。 默认值： 2 。 示例值：2
HealthCheckUnhealthyThreshold	Integer	否	判定后端服务不健康的阈值，当健康检查连续失败多少次后，后端服务的状态由 健康 变为 不健康 。 取值范围： 2–10 。 默认值： 2 。 示例值：2

HealthCheckTemplate

健康检查模板信息

被如下接口引用：DescribeHealthCheckTemplates。

名称	类型	描述
CreateTime	Timestamp ISO8601	创建时间。 示例值：2025-01-01T08:30:00+08:00
HealthCheckCodes	Array of String	健康检查状态码。取值： – 当健康检查协议为 HTTP/HTTPS 时： – http_1xx – http_2xx （默认值） – http_3xx – http_4xx – http_5xx – 当健康检查协议为 GRPC/GRPCS 时：默认值为 12 ，数值范围为 0–99 ，输入值可为数值、多个数值或者范围以及相互组合，如： – "20" – "0–99" 示例值：["http_2xx"]
HealthCheckHealthyThreshold	Integer	判定后端服务健康的阈值，当健康检查连续成功多少次后，后端服务的状态由 不健康 变为 健康 。 取值范围： 2–10 。 默认值： 2 。 示例值：2
HealthCheckHost	String	健康检查域名。 长度限制为 1–255 个字符。 可包含小写字母、数字、短划线（-）和半角句号（.）。 > 仅当 HealthCheckProtocol 设置为 HTTP/HTTPS/GRPC/GRPCS 时，该参数生效。 示例值：example.com

名称	类型	描述
HealthCheckHttpVersion	String	健康检查 HTTP 协议版本，取值： – HTTP1.1（默认） – HTTP1.0 > 仅当HealthCheckProtocol设置为HTTP 或 HTTPS 时，该参数生效。 示例值：HTTP1.1
HealthCheckInterval	Integer	健康检查的时间间隔。单位：秒。 取值范围：2–300。 默认值：5。 示例值：10
HealthCheckMethod	String	健康检查方法，取值： – GET – HEAD（默认值） > 仅当HealthCheckProtocol设置为HTTP 或 HTTPS 时，该参数生效。 示例值：GET
HealthCheckPath	String	健康检查的转发规则路径。长度为 1–80 个字符，只能使用字母、数字、字符-/.%?#&=以及扩展字符_~!()*[]@\$^:'',+。URL 必须以正斜线 (/) 开头。 > 仅当HealthCheckProtocol为 HTTP/HTTPS/GRPC/GRPCS时，转发规则路径参数生效。 示例值：/hello/index.html
HealthCheckPort	Integer	健康检查访问后端服务器的端口。 取值范围：0–65535。 默认值：0，表示后端服务器的端口。 示例值：80
HealthCheckProtocol	String	健康检查协议。取值： – HTTP（默认）：通过发送 HEAD 或 GET 请求模拟浏览器的访问行为来检查服务器应用是否健康。 – HTTPS：通过发送 HEAD 或 GET 请求模拟浏览器的访问行为来检查服务器应用是否健康。（数据加密，相比 HTTP 更安全。） – TCP：通过发送 SYN 握手报文来检测服务器端口是否存活。 – GRPC：通过发送 POST 或 GET 请求来检查服务器应用是否健康。 – GRPCS：通过发送 POST 或 GET 请求来检查服务器应用是否健康。 示例值：HTTP
HealthCheckTemplateId	String	健康检查模板 ID，格式为 hct- 后接字母数字。所有接口（创建、查询、修改、删除）均使用 hct- 前缀。 示例值：hct-m4x8q2pa
HealthCheckTemplateName	String	健康检查模板名称。长度为 1–255 个字符，可包含数字、大小写字母、中文、半角句号（.）、下划线（_）和短划线（-）。 示例值：target-group-health-check

名称	类型	描述
HealthCheckTimeout	Integer	健康检查的响应超时时间。单位：秒。 取值范围： 2-60 。 默认值： 2 。 示例值：2
HealthCheckUnhealthyThreshold	Integer	判定后端服务不健康的阈值，当健康检查连续失败多少次后，后端服务的状态由 健康 变为 不健康 。 取值范围： 2-10 。 默认值： 2 。 示例值：2
ModifyTime	Timestamp ISO8601	修改时间。 示例值：2025-01-01T08:30:00+08:00
Tags	Array of TagInfo	标签。

IPAddressInfo

应用型负载均衡可用区子网映射中的 IP 信息数据结构

被如下接口引用：CreateLoadBalancer, ModifyLoadBalancerAddressType。

名称	类型	必选	描述
Address	String	否	IP 地址 示例值：user.address
AddressId	String	否	EIP AddressId 示例值：eip-q7m2x4pa

InsertHTTPHeaderInfo

插入HTTP Header信息

被如下接口引用：CreateRules, DescribeRules, ModifyRulesAttributes。

名称	类型	必选	描述
Key	String	是	插入的HTTP Header键，长度1 ~ 40个字符，支持的字符集为：a-z A-Z 0-9 - _ 。不支持中文，不支持Cookie,Host,Content-Length,Connection,Upgrade,transfer-encoding,keep-alive,te,authority,x-forwarded-for,x-forwarded-proto,x-forwarded-host,x-forwarded-port。 示例值：ClientVer
Value	String	是	HTTP Header值的类型。 ValueType为SystemDefined时，取值范围 ClientPort：客户端端口，ClientIp：客户端 IP 地址，Protocol：客户端请求的协议，CLBPort：负载均衡实例监听端口。 ValueType为用户Defined时，长度1 ~ 128的可打印字符，不支持"，开头和结尾不能为空格，结尾不能为\。 ValueType为ReferenceHeader时，引用请求头中的某一个header，长度1~128的可打

名称	类型	必选	描述
			印字符，不支持"，开头和结尾不能为空格，结尾不能为\。 示例值：1.0
ValueType	String	是	HTTP Header值的类型，取值： SystemDefined：系统定义的header。 UserDefined：用户自定义的header。 ReferenceHeader：引用请求头中的某一个header。 示例值：UserDefined

Job

异步任务信息

被如下接口引用：DescribeAsyncJobs。

名称	类型	必选	描述
ApiName	String	否	操作接口名称。 示例值：NotifyUnbindTarget
FlowId	Integer	否	任务流Id 示例值：12345
RequestId	String	否	任务请求Id。 示例值：req-f8q2xk9m9
ResourceIds	Array of String	否	资源ID列表 示例值：[]
Status	String	否	任务状态。取值：Processing、Succeeded、Failed。 示例值：Succeeded

ListenerOutput

监听器简要信息出参

被如下接口引用：DescribeListeners。

名称	类型	必选	描述
CaEnable	Boolean	否	是否开启双向认证。 示例值：false
CreateTime	Timestamp ISO8601	否	监听器实例的创建时间。格式：ISO 8601（例如 2025-01-01T08:30:00+08:00） 示例值：2025-01-01T08:30:00+08:00
GzipEnabled	Boolean	否	是否启用 Gzip 压缩。 示例值：false
Http2Enable	Boolean	否	是否启用http2。 示例值：false

名称	类型	必选	描述
IdleTimeout	Integer	否	空闲超时时间。 示例值：15
ListenerId	String	否	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
ListenerName	String	否	监听器名称。 示例值：http-listener
ListenerPort	Integer	否	监听器端口。 示例值：80
ListenerProtocol	String	否	监听器协议。 示例值：HTTPS
ListenerStatus	String	否	监听器状态。取值： – Active : 运行中。 – Provisioning : 创建中。 – Configuring : 变配中。 – ProvisionFailed : 创建失败 示例值：Active
ModifyTime	Timestamp ISO8601	否	监听器实例的最后变更时间。格式：ISO 8601（例如 2025-01-01T08:30:00+08:00） 示例值：2025-01-01T08:30:00+08:00
RequestTimeout	Integer	否	请求超时时间。 示例值：60
Tags	Array of TagInfo	否	标签。
TlsSecurityPolicyId	String	否	安全策略 ID。 示例值：test-policy
XForwardedForConfig	XForwardedForConfig	否	XForwardedFor配置。

LoadBalancer

列表展示的应用型负载均衡实例结构。

被如下接口引用：DescribeLoadBalancers。

名称	类型	必选	描述
AccessLogConfig	AccessLogConfig	否	访问日志配置结构。
AddressIpVersion	String	否	IP 地址版本，取值 IPv4 或 IPv6。 注意：此字段可能返回 null，表示取不到有效值。 示例值：IPv4
AddressType	String	否	负载均衡的地址类型。取值：

名称	类型	必选	描述
			– Internet: 负载均衡具有公网IP地址，DNS域名被解析到公网IP，因此可以在公网环境访问。 – Intranet: 负载均衡只有私网IP地址，DNS域名被解析到私网IP，因此只能被负载均衡所在VPC的内网环境访问。 注意：此字段可能返回 null，表示取不到有效值。 示例值：Internet
CreateTime	String	否	资源创建时间。 注意：此字段可能返回 null，表示取不到有效值。 示例值：2025-01-01T08:30:00+08:00
DeletionProtection	DeletionProtectionConfig	否	删除保护设置信息。
Domain	String	否	DNS域名。 注意：此字段可能返回 null，表示取不到有效值。 示例值：alb-f8q2xk9m-vcpcq5puad9vp3xm.gz-tencentalb.com
LoadBalancerBillingConfig	LoadBalancerBillingConfig	否	负载均衡实例计费配置。
LoadBalancerId	String	否	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 注意：此字段可能返回 null，表示取不到有效值。 示例值：alb-f8q2xk9m
LoadBalancerName	String	否	负载均衡实例名称。 注意：此字段可能返回 null，表示取不到有效值。 示例值：alb-f8q2xk9m-1
LoadBalancerOperationLocks	Array of LoadBalancerOperationLocksItem	否	负载均衡操作锁配置。 注意：此字段可能返回 null，表示取不到有效值。
LoadBalancerStatus	String	否	应用型负载均衡实例状态。取值： – Provisioning: 创建中。 – Active: 运行中。 – Configuring: 变配中。 – Deleting: 删除中。 – ProvisionFailed: 创建失败。 – ConfigureFailed: 变配失败。 – DeletionFailed: 删除失败。 – Abnormal: 异常状态，具体异常

名称	类型	必选	描述
			原因参见 LoadBalancerOperationLocks 字段。 注意：此字段可能返回 null，表示取不到有效值。 示例值：Active
ModificationProtection	ModificationProtectionInfo	否	修改保护设置信息。
Tags	Array of TagInfo	否	标签列表。 注意：此字段可能返回 null，表示取不到有效值。
VpcId	String	否	私有网络 ID。 注意：此字段可能返回 null，表示取不到有效值。 示例值：vpc-92hffaxb

LoadBalancerAddress

应用型负载均衡可用区子网映射中的 IP 信息数据结构

被如下接口引用：CreateLoadBalancer, DescribeLoadBalancerDetail, ModifyLoadBalancerAddressType。

名称	类型	必选	描述
IPv4Address	Array of IPAddressInfo	否	IPv4 地址列表
IPv6Address	Array of IPAddressInfo	否	IPv6 地址列表

LoadBalancerBillingConfig

应用型负载均衡实例计费配置。

被如下接口引用：CreateLoadBalancer, DescribeLoadBalancerDetail, DescribeLoadBalancers。

名称	类型	必选	描述
ChargeType	String	是	实例的计费类型。 取值POSTPAID_BY_HOUR：表示按量计费。 注意：此字段可能返回 null，表示取不到有效值。 示例值：POSTPAID_BY_HOUR
BandwidthPackageId	String	否	共享带宽包 ID。 示例值：bwp-q8m2x4pa

LoadBalancerDetail

负载均衡详细信息

被如下接口引用：DescribeLoadBalancerDetail。

名称	类型	必选	描述
AccessLogConfig	AccessLogConfig	否	访问日志配置。
AddressIpVersion	String	否	IP 地址版本，取值 IPv4 或 IPv6。 示例值：IPv4
AddressType	String	否	应用型负载均衡实例的网络地址类型。取值： – Internet/Public ：负载均衡具有公网IP地址，DNS域名被解析到公网IP，因此可以在公网环境访问。 – Intranet/Internal ：负载均衡只有私网IP地址，DNS域名被解析到私网IP，因此只能被负载均衡所在VPC的内网环境访问。 示例值：Internet
CreateTime	Timestamp ISO8601	否	资源创建时间，格式为yyyy-MM-ddTHH:mm:ss±hh:mm。 示例值：2025-01-01T08:30:00+08:00
DeletionProtection	DeletionProtectionConfig	否	删除保护设置信息。
Domain	String	否	DNS域名。 示例值：alb-f8q2xk9m-vcpcq5puad9vp3xm.gz-tencentalb.com
LoadBalancerBillingConfig	LoadBalancerBillingConfig	否	负载均衡实例付计费配置信息
LoadBalancerId	String	否	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m
LoadBalancerName	String	否	实例名称。 长度为1~80个字符，可包含中文、字母、数字、短划线（-）、正斜线（/）、半角句号（.）和下划线（_）。 示例值：alb-f8q2xk9m-1
LoadBalancerOperationLocks	Array of LoadBalancerOperationLocksItem	否	应用型负载均衡操作锁配置。
LoadBalancerStatus	String	否	应用型负载均衡实例状态。取值： – Provisioning ：创建中。 – Active ：运行中。 – Configuring ：变配中。

名称	类型	必选	描述
			– Deleting: 删除中。 – ProvisionFailed: 创建失败。 – ConfigureFailed: 变配失败。 – DeletionFailed: 删除失败。 – Abnormal: 异常状态，具体异常原因参见 <code>LoadBalancerOperationLocks</code> 字段。 示例值：Active
ModificationProtection	ModificationProtectionInfo	否	修改保护设置信息。
SecurityGroupIds	Array of String	否	应用型负载均衡实例绑定的安全组ID集合。 示例值：[]
Tags	Array of TagInfo	否	标签。
VpcId	String	否	私有网络 ID。 示例值：vpc-92hffaxb
ZoneMappings	Array of ZoneMappingInfo	否	可用区及子网映射列表，最多返回10个可用区。若当前地域支持2个及以上可用区，至少返回2个及以上可用区。

LoadBalancerOperationLocksItem

应用型负载均衡操作锁配置。

被如下接口引用：DescribeLoadBalancerDetail, DescribeLoadBalancers。

名称	类型	必选	描述
LockReason	String	否	锁定的原因。在 LoadBalancerStatus 为 Abnormal 时有效。 注意：此字段可能返回 null，表示取不到有效值。 示例值：欠费隔离
LockType	String	否	锁定的类型。取值： – SecurityLocked: 安全锁定。 – RelatedResourceLocked: 关联锁定。 – FinancialLocked: 欠费锁定。 – ResidualLocked: 残留锁定。 注意：此字段可能返回 null，表示取不到有效值。 示例值：SecurityLocked

ModificationProtectionInfo

修改保护状态信息。

被如下接口引用：DescribeLoadBalancerDetail, DescribeLoadBalancers。

名称	类型	必选	描述
ModificationProtectionEnabled	Boolean	否	是否开启修改保护。开启后，可防止实例被意外修改或删除。 – true：开启修改保护 – false：关闭修改保护 示例值：false
OperatorUin	String	否	1238716123 示例值：设置修改保护的子账号UIN
Reason	String	否	开启修改保护的原因说明。 长度为 1~255 个字符，必须是中文和无害字符串中的字符，可包含中文、字母、数字、短划线（-）、正斜线（/）、半角句号（.）、下划线（_）。 示例值：生产环境实例，防止误操作

PostPayPriceInfo

描述了后付费计费项的价格信息

被如下接口引用：InquirePriceCreateLoadBalancer。

名称	类型	描述
Discount	Float	折扣，如20.0代表2折。 注意：此字段可能返回 null，表示取不到有效值。 示例值：80.0
UnitPrice	Float	单价，单位：元。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1.5
UnitPriceDiscount	Float	折扣单价，单位:元。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1.2

Price

表示负载均衡的价格

被如下接口引用：InquirePriceCreateLoadBalancer。

名称	类型	描述
InstancePrice	PostPayPriceInfo	描述了实例价格，单位：元/小时。
LcuPrice	PostPayPriceInfo	描述了lcu价格，单位：元/个。

QuotaInfo

单个配额项的查询结果。每个结果对应一个配额类型；当请求中传入 ResourceIds 时，每个结果还会对应一个具体资源。

被如下接口引用：DescribeQuota。

名称	类型	必选	描述
Available	Integer	否	当前剩余可用量，计算方式为 Limit - Used。仅当请求参数 DisplayFields 包含 available 时返回有效值；未请求时不返回或为空。 注意：此字段可能返回 null，表示取不到有效值。 示例值：40
Limit	Integer	否	配额上限值。不同配额类型的单位不同，通常表示资源个数；超时时间类配额表示秒。 示例值：60
QuotaType	String	否	配额类型，与请求参数 QuotaTypes 中的取值对应。每种配额类型的含义请参考 QuotaTypes 参数说明。 示例值：alb_quota_loadbalancers_num
ResourceId	String	否	资源 ID。 注意：此字段可能返回 null，表示取不到有效值。 示例值：alblcu-e8p3x9mq
Used	Integer	否	当前已使用量。仅当请求参数 DisplayFields 包含 used 时返回有效值；未请求时不返回或为空。 注意：此字段可能返回 null，表示取不到有效值。 示例值：20

RelatedListener

关联监听器信息

被如下接口引用：DescribeSecurityPolicyRelations。

名称	类型	必选	描述
ListenerId	String	否	监听器 ID，格式为 lst- 后接 8 位字母数字。 示例值：lst-d9p3k7wa
ListenerPort	Integer	否	监听器端口。 示例值：80
ListenerProtocol	String	否	监听器协议。 示例值：HTTPS
LoadBalancerId	String	否	负载均衡实例 ID，格式为 alb- 后接 8 位字母数字。 示例值：alb-f8q2xk9m

RemoveHTTPHeaderInfo

删除 HTTP Header 信息

被如下接口引用：CreateRules, DescribeRules, ModifyRulesAttributes。

名称	类型	必选	描述
Key	String	是	要删除的HTTP Header的键，长度1~40个字符，支持的字符集为：a-z A-Z 0-9 - _ 。不支持Cookie,Host,Content-Length,Connection,Upgrade,transfer-encoding,keep-alive,te,authority,x-forwarded-for,x-forwarded-proto,x-forwarded-host,x-forwarded-port,server 示例值：ClientVer

RuleAction

转发规则动作

被如下接口引用：CreateRules, DescribeRules, ModifyRulesAttributes。

名称	类型	必选	描述
Order	Integer	是	转发动作执行顺序，不能重复，按从小到大顺序执行。取值范围：1~50000。 示例值：1
Type	String	是	转发动作类型。取值： TargetGroup：转发至目标组。 Redirect：重定向。 FixedResponse：返回固定内容。 Rewrite：重写。 InsertHeader：写入HTTP Header。 RemoveHeader：删除HTTP Header。 转发动作必选包含 TargetGroup,Redirect,FixedResponse其中一个，并且执行顺序放在最后。 示例值：TargetGroup
FixedResponseConfig	FixedResponseInfo	否	固定响应内容配置。
InsertHeaderConfig	InsertHTTPHeaderInfo	否	插入HTTP Header配置。
RedirectConfig	HTTPRedirectInfo	否	重定向配置。除HttpCode外，其他配置不能都使用默认值。
RemoveHeaderConfig	RemoveHTTPHeaderInfo	否	删除HTTP Header配置。
RewriteConfig	HTTPRewriteInfo	否	重写配置。
TargetGroupConfig	TargetGroupConfig	否	转发目标组配置。

RuleCondition

转发规则条件

被如下接口引用：CreateRules, DescribeRules, ModifyRulesAttributes。

名称	类型	必选	描述
Type	String	是	转发条件类型。取值： Host：主机。

名称	类型	必选	描述
			Path: 路径。 Header: HTTP Header字段。 QueryString: HTTP查询字符串。 Method: 请求方法。 Cookie: Cookie。 SourceIp: 源 IP。 示例值: Host
CookieConfig	Array of HTTPCookieInfo	否	Cookie配置。
HeaderConfig	HTTPHeaderInfo	否	HTTP Header配置。
HostConfig	Array of String	否	主机名。主机配置在一个规则中只能出现一次，长度3 ~ 128个字符，支持精确匹配，正则匹配，通配匹配。 不能以半角句号（.）、下划线（_）开头或结尾。 精确匹配，支持的字符集为：a-z 0-9 . - _。 正则匹配，波浪线（~）开头为正则匹配，支持的字符集为：a-z 0-9 . - ? = ~ _ - + \ ^ * ! \$ & () []。 通配匹配，星号（*）多字符通配，半角问号（?）单字符通配，支持的字符集为：a-z 0-9 . - _ * ?。 示例值: ["www.myhost.com"]
MethodConfig	Array of String	否	请求方法。取值：HEAD、GET、POST、OPTIONS、PUT、PATCH、DELETE。 示例值: ["GET"]
PathConfig	Array of String	否	转发路径。长度1 ~ 128个字符，支持精确匹配，正则匹配，通配匹配。 精确匹配，支持的字符集为：a-z A-Z 0-9 . - _ / = :。 正则匹配，需以 ~ 开头，~ 开头表示区分大小写，~* 开头表示不区分大小写，支持的字符集为：a-z A-Z 0-9 . - _ / = ? ~ ^ * \$: () [] + 。 通配匹配，* 表示多字符通配，? 表示单字符通配，支持的字符集为：a-z A-Z 0-9 . - _ / = :。 示例值: ["/url"]
QueryStringConfig	Array of HTTPQueryStringInfo	否	查询字符串配置。
SourceIpConfig	Array of String	否	源IP匹配配置。CIDR格式，IP地址x.x.x.x/32，IP网段x.x.x.x/24。 示例值: ["192.168.1.3/32"]

RuleHealthStatusInfo

规则健康检查状态

被如下接口引用：DescribeListenerHealthStatus。

名称	类型	描述
IsDefaultRule	String	是否为默认转发规则。

名称	类型	描述
		示例值：true
RuleId	String	转发规则 ID，格式为 rule- 后接 8 位字母数字。 示例值：rule-h8cy5gwl
TargetGroupHealthInfos	Array of TargetGroupHealthInfo	目标组健康状态。

RuleInput

转发规则创建信息

被如下接口引用：CreateRules。

名称	类型	必选	描述
Actions	Array of RuleAction	是	转发规则动作列表。
Conditions	Array of RuleCondition	是	转发规则条件列表。
Priority	Integer	是	优先级，值越小优先级越高，不能重复，取值范围：1~10000。 示例值：100
Direction	String	否	转发规则的方向。Request：客户端到负载均衡的请求方向，Response：后端服务器到负载均衡的响应方向。默认Request。 示例值：Request
RuleName	String	否	转发规则名称。长度为 1~255 个字符，可包含数字、大小写字母、中文、半角句号（.）、下划线（_）和短划线（-）。 示例值：rule-h8cy5gwl
Tags	Array of TagInfo	否	标签。

RuleModify

转发规则修改信息

被如下接口引用：ModifyRulesAttributes。

名称	类型	必选	描述
Actions	Array of RuleAction	否	转发规则动作列表。
Conditions	Array of RuleCondition	否	转发规则条件列表。
Priority	Integer	否	优先级，值越小优先级越高，取值范围：1~10000。 示例值：100
RuleId	String	否	转发规则 ID，格式为 rule- 后接 8 位字母数字。 示例值：rule-h8cy5gwl
RuleName	String	否	转发规则名称。 示例值：rule-h8cy5gwl

RuleOutput

转发规则信息

被如下接口引用：DescribeRules。

名称	类型	描述
Actions	Array of RuleAction	转发规则动作列表。
Conditions	Array of RuleCondition	转发规则条件列表。
CreateTime	Timestamp ISO8601	创建时间。 示例值：2025-01-01T08:30:00+08:00
Direction	String	转发规则的方向。Request：客户端到负载均衡的请求方向，Response：后端服务器到负载均衡的响应方向。 示例值：Request
ModifyTime	Timestamp ISO8601	最后修改时间。 示例值：2025-01-01T08:30:00+08:00
Priority	Integer	优先级，值越小优先级越高，取值范围：1~10000。 示例值：100
RuleId	String	转发规则 ID，格式为 rule- 后接 8 位字母数字。 示例值：rule-h8cy5gwl
RuleName	String	转发规则名称。 示例值：rule-h8cy5gwl
Status	String	转发规则状态，Provisioning：创建中，Active：运行中，Configuring：配置中。 示例值：Active
Tags	Array of TagInfo	标签列表。

SecurityPolicyCapability

不同TLS版本支持的加密套件信息。

被如下接口引用：DescribeSecurityPolicyCapabilities。

名称	类型	必选	描述
Ciphers	Array of String	否	支持的加密套件列表。 示例值：["TLS_AES_128_GCM_SHA256"]
TLSVersion	String	否	支持的 TLS 协议版本。可选值包括：TLSv1.0、TLSv1.1、TLSv1.2、TLSv1.3。 示例值：TLSv1.3

SecurityPolicyInfo

安全策略信息。

被如下接口引用：DescribeSecurityPolicies, DescribeSystemSecurityPolicies。

名称	类型	描述
Ciphers	Array of String	支持的加密套件列表。 支持的加密套件，具体依赖 TLSVersions 的取值。 Cipher 只要被任何一个传入的 TLSVersions 支持即可。 说明：若选择了 TLSv1.3，那么 Cipher 列表必须包含 TLSv1.3 支持的 Cipher。 请调用 DescribeSecurityPolicyCapabilities 接口获取支持的加密套件列表。 示例值：["TLS_AES_128_GCM_SHA256"]
CreateTime	String	创建时间。 示例值：2025-01-01T08:30:00+08:00
SecurityPolicyId	String	安全策略 ID，格式为 tls- 后接 8 位字母数字。 示例值：tls-t2ckyduq
SecurityPolicyName	String	安全策略名称。长度为2~128个英文或中文字符，必须以字母或中文开头，可包含数字、半角句号（.）、下划线（_）和短划线（-）。 示例值：test-policy
Status	String	安全策略状态。当前接口最常返回 Active，表示安全策略处于可用状态。 示例值：Active
TLSVersions	Array of String	支持的 TLS 协议版本列表。可选值包括：TLSv1.0、TLSv1.1、TLSv1.2、TLSv1.3。 示例值：["TLSv1.2", "TLSv1.3"]
Tags	Array of TagInfo	标签信息。

SecurityPolicyRelations

安全策略与监听的关联关系列表。

被如下接口引用：DescribeSecurityPolicyRelations。

名称	类型	必选	描述
RelatedListeners	Array of RelatedListener	否	安全策略与监听的关联关系列表。
SecurityPolicyId	String	否	安全策略 ID，格式为 tls- 后接 8 位字母数字。 示例值：tls-t2ckyduq

StickySessionConfig

会话保持配置。

被如下接口引用：CreateTargetGroup, DescribeTargetGroups, ModifyTargetGroupAttributes。

名称	类型	必选	描述
StickySessionEnabled	Boolean	是	是否开启会话保持。 – true : 开启。 – false : 不开启。 注意：此字段可能返回 null，表示取不到有效值。 示例值：false
Cookie	String	否	自定义 Cookie 名称。 长度为 1~255 个字符，只能包含英文字母和数字字符，且不能为 <code>tgw_l17_tg_route</code> ，该字段为目标组间会话保持Cookie保留字段。 >仅当 StickySessionEnabled 为 true 时该参数生效。 示例值：SERVERID
CookieTimeout	Integer	否	会话保持时间。 取值范围：1~86400，单位：秒。 默认值：1000。 >仅当 StickySessionEnabled 为 true 时，该参数生效。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1000
StickySessionType	String	否	会话保持类型（处理Cookie的方式）。 – Insert （默认值）：植入 Cookie。客户端第一次访问后端服务时，应用型负载均衡会在返回请求中植入 Cookie。下次客户端请求时携带该 Cookie，负载均衡将请求转发到上次请求的相同后端服务。 – Rewrite ：重写 Cookie。负载均衡会对用户自定义的 Cookie 进行重写，下次客户端请求时携带该 Cookie，负载均衡将请求转发到上次请求的相同后端服务。 >仅当 StickySessionEnabled 为 true 时该参数生效。 注意：此字段可能返回 null，表示取不到有效值。 示例值：Insert

TagInfo

标签信息

被如下接口引用：CreateHealthCheckTemplate, CreateListener, CreateLoadBalancer, CreateRules, CreateSecurityPolicy, CreateTargetGroup, DescribeHealthCheckTemplates, DescribeListenerDetail, DescribeListeners, DescribeLoadBalancerDetail, DescribeLoadBalancers, DescribeRules, DescribeSecurityPolicies, DescribeSystemSecurityPolicies, DescribeTargetGroups。

名称	类型	必选	描述
TagKey	String	是	标签的键 示例值：env
TagValue	String	是	标签的值 示例值：prod

TargetGroupConfig

目标组配置

被如下接口引用：CreateListener, CreateRules, DescribeListenerDetail, DescribeRules, ModifyListenerAttributes, ModifyRulesAttributes。

名称	类型	必选	描述
TargetGroups	Array of TargetGroupTuple	是	目标组列表。
TargetGroupStickySession	TargetGroupStickySession	否	目标组间会话保持

TargetGroupHealthInfo

目标组健康检查状态

被如下接口引用：DescribeListenerHealthStatus。

名称	类型	描述
HealthCheckEnabled	Boolean	是否开启健康检查。 示例值：true
TargetGroupId	String	目标组 ID，格式为 lbtg- 后接 8 位字母数字。 示例值：lbtg-0zrnc9qa
TargetHealthStatusInfos	Array of TargetHealthStatusInfo	服务健康检查状态列表。
Type	String	转发动作类型。取值： TargetGroup：转发至目标组。 Redirect：重定向。 FixedResponse：返回固定内容。 Rewrite：重写。 InsertHeader：写入HTTP Header。 RemoveHeader：删除HTTP Header。 转发动作必选包含 TargetGroup,Redirect,FixedResponse其中一个，并且 执行顺序放在最后。 示例值：TargetGroup

TargetGroupOutput

目标组简要信息出参

被如下接口引用：DescribeTargetGroups。

名称	类型	描述
CreateTime	Timestamp ISO8601	创建时间。 示例值：2025-01-01T08:30:00+08:00
HealthCheckConfig	HealthCheckConfig	健康检查配置。
KeepaliveEnabled	Boolean	是否开启长连接。 示例值：true

名称	类型	描述
Protocol	String	后端服务协议类型。取值： – HTTP（默认）：支持绑定HTTP、HTTPS的监听器 – HTTPS：支持绑定HTTPS类型的监听器 – GRPC：支持绑定HTTPS类型的监听器 – GRPCS：支持绑定HTTPS类型的监听器 示例值：HTTP
RelatedLoadBalancersCount	Integer	目标组关联的负载均衡数量。 示例值：0
SchedulerAlgorithm	String	调度算法。 示例值：wrr
StickySessionConfig	StickySessionConfig	会话保持配置。
Tags	Array of TagInfo	标签。
TargetGroupId	String	目标组 ID，格式为 lbtg- 后接 8 位字母数字。 示例值：lbtg-0zrnc9qa
TargetGroupName	String	目标组名称。默认为目标组ID。长度为 1–255 个字符，可包含数字、大小写字母、中文、半角句号（.）、下划线（_）和短划线（-）。 示例值：target-group-name
TargetGroupStatus	String	目标组状态。取值： – Provisioning：创建中。 – ProvisionFailed：创建失败。 – Active：运行中。 – Configuring：配置变更中。 示例值：Active
TargetType	String	目标组类型。取值： – Instance：Cvm服务器类型或Eni弹性网卡类型 示例值：CVM
VpcId	String	私有网络 ID。 示例值：vpc-92hffaxb

TargetGroupStickySession

目标组之间会话保持

被如下接口引用：CreateListener, DescribeListenerDetail, ModifyListenerAttributes。

名称	类型	必选	描述
Enabled	Boolean	是	是否开启会话保持，默认关闭。 示例值：false
Timeout	Integer	否	超时时间，单位秒，取值范围：1~86400，默认值：1000。 示例值：1000

TargetGroupTuple

目标组基础配置

被如下接口引用：CreateListener, DescribeListenerDetail, ModifyListenerAttributes。

名称	类型	必选	描述
TargetGroupId	String	是	目标组 ID，格式为 lbtg- 后接 8 位字母数字。 示例值：lbtg-0zrnc9qa
Weight	Integer	否	权重，取值范围：[0, 100]，默认为 10。 示例值：10

TargetHealthStatusInfo

服务健康状态信息

被如下接口引用：DescribeListenerHealthStatus。

名称	类型	描述
Status	String	后端服务健康状态。DescribeListenerHealthStatus 仅返回非健康后端时，该值为 UnHealthy。 示例值：UnHealthy
TargetId	String	后端服务实例 ID，CVM 实例格式为 ins- 后接 8 位字母数字。 示例值：ins-x8q2m4pa
TargetIp	String	后端目标服务 IP。 示例值：10.0.0.1
TargetPort	Integer	后端服务器端口。 示例值：80

TargetOutput

后端服务输出参数。

被如下接口引用：DescribeTargetGroupTargets。

名称	类型	描述
EniId	String	弹性网卡 ID。 示例值：eni-p4q7x2ma
Port	Integer	后端服务器使用的端口。取值范围：1 – 65535。 示例值：80
TargetId	String	后端服务实例 ID，CVM 实例格式为 ins- 后接 8 位字母数字。 示例值：ins-x8q2m4pa
TargetIp	String	后端服务 IP。TargetIp 和 TargetId 需要至少传一个。 – 当服务器组为 Instance 类型时，该参数为 Eni 的主内网 IP 或辅助内网 IP。

名称	类型	描述
		示例值：10.0.0.1
TargetName	String	后端服务名称。当前只有CVM后端类型后端服务返回有效名称。 示例值：test-xxxx
TargetStatus	String	后端服务的状态。取值： – Adding ：添加中。 – Active ：正常可用状态。 – Configuring ：配置中。 – Removing ：移除中。 示例值：Active
TargetType	String	后端服务类型。 示例值：CVM
Weight	Integer	后端服务的权重，取值范围： 0 – 100 。默认值为 100 。如果设置权重为 0 ，则不会将请求转发给该后端服务。 示例值：10

TargetToAdd

添加至目标组的后端服务

被如下接口引用：AddTargetsToTargetGroup。

名称	类型	必选	描述
Port	Integer	是	后端服务器使用的端口。取值范围： 1 – 65535 。 >当目标组的 targetType 取值为 Instance 时，该参数必传。 示例值：80
TargetIp	String	是	后端服务IP。 TargetIp 和 TargetId 需要至少传一个。 – 当服务器组为 Instance 类型时，该参数为 Eni 的主内网 IP 或辅助内网 IP。 示例值：10.0.0.1
Weight	Integer	否	后端服务的权重，取值范围： 0 – 100 。默认值为 10 。如果设置权重为 0 ，则不会将请求转发给该后端服务。 示例值：10

TargetToModify

需要修改的后端服务。

被如下接口引用：ModifyTargetsInTargetGroup。

名称	类型	必选	描述
TargetIp	String	是	后端服务IP。 TargetIp 和 TargetId 需要至少传一个。

名称	类型	必选	描述
			– 当服务器组为 Instance 类型时，该参数为 Eni 的主内网 IP 或辅助内网 IP。 示例值：10.0.0.1
Port	Integer	否	后端服务器使用的端口。取值范围： 1 – 65535 。 >当目标组的 targetType 取值为 Instance 时，该参数必传。 示例值：80
Weight	Integer	否	后端服务的权重，取值范围： 0 – 100 。如果设置权重为 0 ，则不会将请求转发给该后端服务。 示例值：10

TargetToRemove

从目标组移除的后端服务。

被如下接口引用：RemoveTargetsFromTargetGroup。

名称	类型	必选	描述
Port	Integer	是	后端服务器使用的端口。取值范围： 1 – 65535 。 >当目标组的 targetType 取值为 Instance 时，该参数必传。 示例值：80
TargetIp	String	是	后端服务IP。 TargetIp 和 TargetId 需要至少传一个。 – 当服务器组为 Instance 类型时，该参数为 Eni 的主内网 IP 或辅助内网 IP。 示例值：10.0.0.1

XForwardedForConfig

转发配置

被如下接口引用：CreateListener, DescribeListenerDetail, DescribeListeners, ModifyListenerAttributes。

名称	类型	必选	描述
XForwardedForAlbIdEnabled	Boolean	否	是否通过 ALB-ID 头字段获取负载均衡实例ID。 – true: 是。 – false: 否。 示例值：false
XForwardedForClientSrcPortEnabled	Boolean	否	是否通过X-Forwarded-Client-srcport头字段获取访问负载均衡实例客户端的端口。 – true: 是。 – false: 否。 示例值：false
XForwardedForHostEnabled	Boolean	否	是否开启通过X-Forwarded-Host头字段获取访问负载均衡实例客户端的域名。 – true: 是。

名称	类型	必选	描述
			– false: 否。 示例值: false
XForwardedForMode	String	否	指定如何处理 X-Forwarded-For (XFF) HTTP 头字段。 – append: 附加模式 (默认)，将客户端的真实 IP 地址附加到 X-Forwarded-For 头的末尾，保留原有的 XFF 链路信息 – remove: 删除模式，移除 X-Forwarded-For 头字段，不将该头传递给后端服务器 – passthrough: 透传模式，保持 X-Forwarded-For 头不变，直接透传给后端服务器，不做任何修改 示例值: append
XForwardedForPortEnabled	Boolean	否	是否通过 X-Forwarded-Port 头字段获取负载均衡实例的监听端口。 – true: 是。 – false: 否。 示例值: false
XForwardedForProtoEnabled	Boolean	否	是否通过 X-Forwarded-Proto 头字段获取负载均衡实例的监听协议。 – true: 是。 – false: 否。 示例值: false
XTencentClientIDNEnabled	Boolean	否	是否通过 X-Tencent-Client-IDN 头访问 客户端证书的颁发者 \$ssl_client_i_dn。 – true: 是。 – false: 否。 示例值: false
XTencentClientSDNEnabled	Boolean	否	是否通过 X-Tencent-Client-SDN 头访问客户端证书的主题 \$ssl_client_s_dn。 – true: 是。 – false: 否。 示例值: false
XTencentClientSerialEnabled	Boolean	否	是否通过 X-Tencent-Client-Serial 头访问 客户端证书的序列号 \$ssl_client_serial。 – true: 是。 – false: 否。 示例值: false
XTencentClientVerifyEnabled	Boolean	否	是否通过 X-Tencent-Client-Verify 头访问 客户端证书的验证结果 \$ssl_client_verify。 – true: 是。 – false: 否。

名称	类型	必选	描述
			示例值：false

Zone

可用区信息

被如下接口引用：DescribeZones。

名称	类型	描述
LocalName	String	可用区名称 示例值：广州三区
Zoneld	String	可用区 ID 示例值：ap-guangzhou-3
ZoneStatus	String	可用区状态 示例值：available

ZoneMappingInfo

可用区及子网映射结构体

被如下接口引用：DescribeLoadBalancerDetail。

名称	类型	必选	描述
SubnetId	String	是	子网 ID。 注意：此字段可能返回 null，表示取不到有效值。 示例值：subnet-bi75pogm
Zoneld	String	是	可用区ID。最多支持添加10个可用区。若当前地域支持2个及以上的可用区，至少需要添加2个可用区。 您可以通过调用 DescribeZones 接口获取可用区ID对应的可用区的信息。 注意：此字段可能返回 null，表示取不到有效值。 示例值：ap-guangzhou-3
LoadBalancerAddress	LoadBalancerAddress	否	负载均衡 VIP/EIP 信息
Status	String	否	可用区状态。取值： • Active：运行中。 • Stopped：已停止。 • Shifted：已移除。 • Starting：启动中。 • Stopping：停止中。

名称	类型	必选	描述
			示例值：Active

ZoneMappingsItem

用于购买或者修改使用的可用区及子网映射结构体

被如下接口引用：CreateLoadBalancer, ModifyLoadBalancerAddressType。

名称	类型	必选	描述
SubnetId	String	是	子网 ID。 注意：此字段可能返回 null，表示取不到有效值。 示例值：subnet-bi75pogm
ZoneId	String	是	可用区ID。最多支持添加10个可用区。若当前地域支持2个及以上的可用区，至少需要添加2个可用区。 您可以通过调用 DescribeZones 接口获取可用区ID对应的可用区的信息。 注意：此字段可能返回 null，表示取不到有效值。 示例值：ap-guangzhou-3
LoadBalancerAddress	LoadBalancerAddress	否	公网实例绑定的EIP实例ID。 注意：此字段可能返回 null，表示取不到有效值。

错误码

最近更新时间：2026-07-07 10:58:13

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 <code>Authorization</code> 不符合腾讯云标准。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。
AuthFailure.MFAFailure	MFA 错误。
AuthFailure.SecretIdNotFound	密钥不存在。请在 控制台 检查密钥是否已被删除或者禁用，如状态正常，请检查密钥是否填写正确，注意前后不得有空格。
AuthFailure.SignatureExpire	签名过期。Timestamp 和服务器时间相差不得超过五分钟，请检查本地时间是否和标准时间同步。
AuthFailure.SignatureFailure	签名错误。签名计算错误，请对照调用方式中的签名方法文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未授权。请参考 CAM 文档对鉴权的说明。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
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。
UnsupportedProtocol	http(s) 请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
AuthFailure	CAM签名/鉴权错误。
FailedOperation.ResourceAlreadyExists	资源已存在。
FailedOperation.RuleAlreadyExists	规则优先级已存在。
InvalidFilter	无效的过滤器
InvalidParameter.InvalidClientToken	ClientToken 格式无效。
InvalidParameter.InvalidFieldFormat	字段格式错误。
InvalidParameter.InvalidFieldType	字段类型错误。
InvalidParameter.InvalidFieldValue	字段值错误。
InvalidParameter.InvalidQuotaCheckRequest	配额检查请求无效。
InvalidParameter.MissingAction	缺少 Action 参数。
InvalidParameter.MissingField	请填写必要参数。
InvalidParameterValue.BusinessParameterMismatch	业务参数不一致。
LimitExceeded.Quota	配额超限。
OperationDenied	操作被拒绝。
RegionError	地域错误
ResourceInUse.LoadBalancerHasListeners	实例下还存在监听器，无法删除。
ResourceNotFound.Certificate	证书未找到。
ResourceNotFound.Listener	监听器未找到。
ResourceNotFound.LoadBalancer	负载均衡资源不存在。
ResourceNotFound.Rule	转发规则未找到
ResourceNotFound.TargetGroup	目标组未找到。
ResourceUnavailable.Certificate	证书处于不可用状态（更新中/删除中）。
ResourceUnavailable.Listener	监听器处于不可用状态。
ResourceUnavailable.LoadBalancer	ALB状态不可用。
ResourceUnavailable.PreviousTaskNotCompleted	上个任务未完成。
ResourceUnavailable.Rule	规则状态不可用。
ResourceUnavailable.TargetGroup	目标组处于不可用状态
ResourceUnavailable.UpstreamStatusBlocked	上级资源状态阻断。
ResourcesSoldOut	资源售罄。

错误码	说明
UnauthorizedOperation.ServiceRoleUnauthorized	服务角色未授权
UnsupportedOperation.CertificateAlreadyBound	监听器不允许绑定已绑定的证书。
UnsupportedOperation.CertificateNotBound	监听器不允许解绑未绑定的证书。
UnsupportedOperation.DisassociateDefaultCertificate	监听器不允许解绑默认证书。
UnsupportedOperation.InvalidStateTransition	无效的状态转换。
UnsupportedOperation.TargetGroupProtocolMismatch	监听器协议与目标组后端转发协议不匹配
UnsupportedOperation.UnsupportedProtocol	监听器协议不支持此操作。