

防火墙管理 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

企业安全组相关接口

创建企业安全组规则

删除企业安全组规则

查询企业安全组规则详情

查询企业安全组规则列表

查询规则下发状态

忽略策略问题

修改企业安全组规则

通用相关接口

查询防火墙管理规则下发账号

多账号管理相关接口

查询集团成员列表

获取集团概览

互联网边界规则相关接口

创建互联网边界规则

删除互联网边界规则

查询互联网边界规则列表

修改互联网边界规则

修改互联网边界规则序号

NAT边界规则相关接口

创建NAT边界规则

删除NAT边界规则

查询NAT边界规则列表

修改NAT边界规则

NAT边界规则排序

VPC边界规则相关接口

创建VPC边界规则

删除VPC边界规则

查询VPC边界规则列表

修改VPC边界规则

VPC边界规则排序

策略分析相关接口

查询账号+产品维度风险统计

查询风险分类统计数据

查询策略风险列表

实时分析风险详情

取消忽略策略风险

创建风险分析任务

规则组相关接口

创建NAT边界规则组

创建互联网边界规则组

创建VPC边界规则组

创建企业安全组规则组

删除规则组

修改规则组信息

规则管理相关接口

创建下发规则

删除下发规则

查询规则列表

查询下发规则信息

下发规则

编辑下发规则

下发规则快速排序

数据结构

错误码

API 文档

更新历史

最近更新时间：2026-07-06 15:58:37

第 1 次发布

发布时间：2026-07-06 15:58:06

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CancelIgnorePolicyRisk](#)
- [CreateAnalyzePolicyTask](#)
- [CreateEdgeAclRule](#)
- [CreateEdgeAclRuleGroup](#)
- [CreateNatAclRule](#)
- [CreateNatAclRuleGroup](#)
- [CreateSecurityGroupRule](#)
- [CreateSecurityGroupRuleGroup](#)
- [CreateStrategy](#)
- [CreateVpcAclRule](#)
- [CreateVpcAclRuleGroup](#)
- [DeleteEdgeAclRule](#)
- [DeleteNatAclRule](#)
- [DeleteRuleGroup](#)
- [DeleteSecurityGroupRule](#)
- [DeleteStrategy](#)
- [DeleteVpcAclRule](#)
- [DescribeEdgeAclRules](#)
- [DescribeNatAclRules](#)
- [DescribeOrganMembers](#)
- [DescribeOrganSummary](#)
- [DescribePolicyRiskAccountProductStats](#)
- [DescribeRiskAnalysisDetails](#)
- [DescribeRiskCategoryStats](#)
- [DescribeRiskList](#)
- [DescribeSecurityGroupRule](#)
- [DescribeSecurityGroupRules](#)
- [DescribeStrategies](#)
- [DescribeStrategy](#)

- DescribeStrategyAccounts
- DescribeStrategyDispatchStatus
- DescribeVpcAclRules
- DispatchStrategy
- IgnorePolicyRisk
- ModifyEdgeAclRule
- ModifyEdgeAclRuleSequence
- ModifyNatAclRule
- ModifyNatAclRuleSequence
- ModifyRuleGroup
- ModifySecurityGroupRule
- ModifyStrategy
- ModifyStrategySequence
- ModifyVpcAclRule
- ModifyVpcAclRuleSequence

新增数据结构:

- Account
- AccountGroupInfo
- AccountGroupQuotaDetail
- AccountProductDetailStats
- AccountStatsGroup
- AddressTemplateSpecification
- AnalysisSgRuleInfoResp
- CommonFilter
- EdgeAclRuleInfo
- EdgeAclRuleResp
- MemberInfo
- NatAclRule
- NatAclRuleResp
- OrganMemberItem
- OrganSummary
- PolicyRisk
- ReceiveAccount
- RiskCategoryItem
- SecGroupRuleResp
- SecurityGroupRiskPolicy
- SecurityGroupRule
- SequenceIndex
- ServiceTemplateSpecification
- SgDnsParseCount
- SgRuleResp

- [StrategyReq](#)
- [StrategyResp](#)
- [VpcAclRule](#)
- [VpcAclRuleResp](#)

简介

最近更新时间：2026-07-06 15:58:35

概述

防火墙管理（Firewall Manager，FWM）是一款防护策略管理产品，支持多账号、多产品、跨地域的安全防护策略集中化管理，确保全网安全策略的一致性和高效执行。产品支持智能分析识别规则冗余、冲突及无效配置，并提供优化建议，有效提升规则质量和管理效率。

本章节介绍的防火墙管理 API 接口均为 API 3.0接口。

您可以调用 API 对腾讯云防火墙管理进行操作，例如新增、编辑、删除、下发规则等。

防火墙管理支持的所有接口信息，请参见 [API 概览]。

术语表

防火墙管理 API 接口的常见术语请参见下表：

术语	描述
安全组	在防火墙管理中，安全组（Security Group）是一种有状态的包过滤功能的虚拟防火墙，它用于设置单台或多台云服务器的网络访问控制，可以将同一地域内具有相同网络安全隔离需求的云服务器实例加到同一个安全组内，通过安全组的网络策略对云服务器的出入流量进行安全过滤。
互联网边界	互联网边界（Internet Border）是私有网络（VPC）或云服务器实例直接与公共互联网连接的边界点。在此边界上，通常通过公网网关、弹性公网 IP（EIP）或负载均衡器等将内部资源暴露给公网访问，或允许内部资源主动访问公网。在此边界实施的安全策略主要用于防护来自或去往公网的南北向流量。
NAT 边界	NAT 边界（NAT Border）是网络地址转换（NAT）设备所处的位置，通常是连接私有网络（VPC）子网与公网的边界点。在此边界上，NAT 网关负责将私有子网内实例的私网 IP 地址转换为公网 IP 地址进行出站访问，也可将公网流量转发至指定的私网实例。安全策略可在此边界点应用于经过 NAT 转换的流量。
企业安全组	企业安全组（Enterprise Security Group）是安全组的高级形态，提供更强大的管理能力和更精细的访问控制。它允许管理员在 VPC 级别集中定义统一的安全策略模板，并支持跨实例、跨子网甚至跨可用区进行策略的批量管理和同步。企业安全组简化了大规模实例的安全策略配置和维护工作。
VPC 边界	VPC 边界（VPC Border）是腾讯云私有网络（VPC）内部不同子网之间进行网络通信的边界。通过配置 VPC 内的网络 ACL 或路由策略，可以在 VPC 边界实施网络访问控制，管理不同子网间的东西向流量，实现子网级别的隔离和访问策略。

使用限制

对于 API 接口的参数限制，请参考各接口文档中的参数说明。

API 快速入门

您可以使用 API Explorer 工具在线调用 API。

本文以创建互联网边界规则组为例，通过 API Explorer 工具调用 API 接口的步骤如下：

1. 进入 [API Explorer](#) 工具页面。更多 API Explorer 工具使用信息，请参见 [使用 API Explorer](#)。
2. 调用 [CreateEdgeAclRuleGroup] 接口，创建互联网边界规则组。
3. 创建互联网边界规则组后即可在控制台页面查看到创建好的互联网边界规则组，可以对互联网边界规则组进行配置，具体操作方式请参见[文档](#)。

API 概览

最近更新时间：2026-07-06 15:58:37

企业安全组相关接口

接口名称	接口功能	频率限制（次/秒）
CreateSecurityGroupRule	创建企业安全组规则	20
DeleteSecurityGroupRule	删除企业安全组规则	20
DescribeSecurityGroupRule	查询企业安全组规则详情	20
DescribeSecurityGroupRules	查询企业安全组规则列表	20
DescribeStrategyDispatchStatus	查询规则下发状态	20
IgnorePolicyRisk	忽略策略问题	20
ModifySecurityGroupRule	修改企业安全组规则	20

通用相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeStrategyAccounts	查询防火墙管理规则下发账号	20

多账号管理相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeOrganMembers	查询集团成员列表	20
DescribeOrganSummary	获取集团概览	20

互联网边界规则相关接口

接口名称	接口功能	频率限制（次/秒）
CreateEdgeAclRule	创建互联网边界规则	20
DeleteEdgeAclRule	删除互联网边界规则	20
DescribeEdgeAclRules	查询互联网边界规则列表	20
ModifyEdgeAclRule	修改互联网边界规则	20

接口名称	接口功能	频率限制（次/秒）
ModifyEdgeAclRuleSequence	修改互联网边界规则序号	20

NAT边界规则相关接口

接口名称	接口功能	频率限制（次/秒）
CreateNatAclRule	创建NAT边界规则	20
DeleteNatAclRule	删除NAT边界规则	20
DescribeNatAclRules	查询NAT边界规则列表	20
ModifyNatAclRule	修改NAT边界规则	20
ModifyNatAclRuleSequence	NAT边界规则排序	20

VPC边界规则相关接口

接口名称	接口功能	频率限制（次/秒）
CreateVpcAclRule	创建VPC边界规则	20
DeleteVpcAclRule	删除VPC边界规则	20
DescribeVpcAclRules	查询VPC边界规则列表	20
ModifyVpcAclRule	修改VPC边界规则	20
ModifyVpcAclRuleSequence	VPC边界规则排序	20

策略分析相关接口

接口名称	接口功能	频率限制（次/秒）
DescribePolicyRiskAccountProductStats	查询账号+产品维度风险统计	20
DescribeRiskCategoryStats	查询风险分类统计数据	20
DescribeRiskList	查询策略风险列表	20
DescribeRiskAnalysisDetails	实时分析风险详情	20
CancelIgnorePolicyRisk	取消忽略策略风险	20
CreateAnalyzePolicyTask	创建风险分析任务	20

规则组相关接口

接口名称	接口功能	频率限制（次/秒）
CreateNatAclRuleGroup	创建NAT边界规则组	20
CreateEdgeAclRuleGroup	创建互联网边界规则组	20
CreateVpcAclRuleGroup	创建VPC边界规则组	20
CreateSecurityGroupRuleGroup	创建企业安全组规则组	20
DeleteRuleGroup	删除规则组	20
ModifyRuleGroup	修改规则组信息	20

规则管理相关接口

接口名称	接口功能	频率限制（次/秒）
CreateStrategy	创建下发规则	20
DeleteStrategy	删除下发规则	20
DescribeStrategies	查询规则列表	20
DescribeStrategy	查询下发规则信息	20
DispatchStrategy	下发规则	20
ModifyStrategy	编辑下发规则	20
ModifyStrategySequence	下发规则快速排序	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：2026-07-06 15:58:35

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

接入地域	域名
南美地区（圣保罗）	fwm.sa-saopaulo.tencentcloudapi.com
欧洲地区（法兰克福）	fwm.eu-frankfurt.tencentcloudapi.com

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

金融区接入地域	金融区域名
华东地区（上海金融）	fwm.ap-shanghai-fsi.tencentcloudapi.com
华南地区（深圳金融）	fwm.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：2026-07-06 15:58:36

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

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

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

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

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

参数名称	类型	必选	描述
Action	String	是	HTTP 请求头：X-TC-Action。操作的接口名称。取值参考接口文档输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	HTTP 请求头：X-TC-Region。地域参数，用来标识希望操作哪个地域的数据。取值参考接口文档中输入参数章节关于公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	HTTP 请求头：X-TC-Timestamp。当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。 注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
Version	String	是	HTTP 请求头：X-TC-Version。操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKID***/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKID*** 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service 为具体产品名，通常为域名前缀。例如，域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 fwm；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，用于防止回放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见 文档 。
Version	String	是	操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

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

HTTP GET 请求结构示例：

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

```
Host: cvm.tencentcloudapi.com
```

HTTP POST 请求结构示例:

```
https://cvm.tencentcloudapi.com/  
  
Host: cvm.tencentcloudapi.com  
Content-Type: application/x-www-form-urlencoded  
  
Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334  
&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guang  
zhou&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
华南地区（广州）	ap-guangzhou

签名方法 v3

最近更新时间：2026-07-06 15:58:36

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

推荐使用 API Explorer

</> 点击调试

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

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

为什么要进行签名

签名通过以下方式帮助保护请求：

1. 验证请求者的身份

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

2. 保护传输中的数据

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

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

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v3”，可以对生成签名过程进行验证，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 8 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)、[Ruby](#)。

申请安全凭证

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

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

申请安全凭证的具体步骤如下：

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

签名版本 v3 签名过程

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

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

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

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

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

*****。用户想查看广州云服务器名为“未命名”的主机状态，只返回一条数据。则请求可能为：

```
curl -X POST https://cvm.tencentcloudapi.com \
-H "Authorization: TC3-HMAC-SHA256 Credential=AKID*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=10b1a37a7301a02ca19a647ad722d5e43b4b3cff309d421d85b46093f6ab6c4f" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Host: cvm.tencentcloudapi.com" \
-H "X-TC-Action: DescribeInstances" \
-H "X-TC-Timestamp: 1551113065" \
-H "X-TC-Version: 2017-03-12" \
-H "X-TC-Region: ap-guangzhou" \
-d '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
```

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

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

```
SignedHeaders + '\n' +
HashedRequestPayload
```

字段名称	解释
HTTPRequestMethod	HTTP 请求方法 (GET、POST)。此示例取值为 POST。
CanonicalURI	URI 参数, API 3.0 固定为正斜杠 (/)。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串, 对于 POST 请求, 固定为空字符串 "", 对于 GET 请求, 则为 URL 中问号 (?) 后面的字符串内容, 例如: Limit=10&Offset=0。 注意: CanonicalQueryString 需要参考 RFC3986 进行 URLEncode 编码 (特殊字符编码后需大写), 字符集 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回值，也可能会提供打印友好的十六进制格式的回值，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.DataMapper;

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

    public static byte[] hmac256(byte[] key, String msg) throws Exception {
        Mac mac = Mac.getInstance("HmacSHA256");
        SecretKeySpec secretKeySpec = new SecretKeySpec(key, mac.getAlgorithm());
```

```
mac.init(secretKeySpec);
return mac.doFinal(msg.getBytes(UTF8));
}

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

public static void main(String[] args) throws Exception {
    String service = "cvm";
    String host = "cvm.tencentcloudapi.com";
    String region = "ap-guangzhou";
    String action = "DescribeInstances";
    String version = "2017-03-12";
    String algorithm = "TC3-HMAC-SHA256";
    String timestamp = "1551113065";
    //String timestamp = String.valueOf(System.currentTimeMillis() / 1000);
    SimpleDateFormat sdf = new SimpleDateFormat("yyyy-MM-dd");
    // 注意时区，否则容易出错
    sdf.setTimeZone(TimeZone.getTimeZone("UTC"));
    String date = sdf.format(new Date(Long.valueOf(timestamp + "000")));

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

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

    // ***** 步骤 2: 拼接待签名字符串 *****
    String credentialScope = date + "/" + service + "/" + "tc3_request";
    String hashedCanonicalRequest = sha256Hex(canonicalRequest);
```

```
String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashed
CanonicalRequest;
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)).t
oLowerCase();
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.utcnow().strftime("%Y-%m-%d")
params = {"Limit": 1, "Filters": [{"Values": [u"未命名"], "Name": "instance-name"}]}

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

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

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

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

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

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

Golang

```
package main

import (
"crypto/hmac"
```

```
"crypto/sha256"
"encoding/hex"
"fmt"
"os"
"strings"
"time"
)

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

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

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
    host := "cvm.tencentcloudapi.com"
    algorithm := "TC3-HMAC-SHA256"
    service := "cvm"
    version := "2017-03-12"
    action := "DescribeInstances"
    region := "ap-guangzhou"
    //var timestamp int64 = time.Now().Unix()
    var timestamp int64 = 1551113065

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

```
canonicalRequest := fmt.Sprintf("%s\n%s\n%s\n%s\n%s\n%s",
    httpRequestMethod,
    canonicalURI,
    canonicalQueryString,
    canonicalHeaders,
    signedHeaders,
    hashedRequestPayload)
fmt.Println(canonicalRequest)

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

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

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

curl := fmt.Sprintf(`curl -X POST https://%s\
-H "Authorization: %s"\
-H "Content-Type: application/json; charset=utf-8"\
-H "Host: %s" -H "X-TC-Action: %s"\
-H "X-TC-Timestamp: %d"\
-H "X-TC-Version: %s"\
-H "X-TC-Region: %s"`)
```

```
-d '%s`, host, authorization, host, action, timestamp, version, region, payload)
fmt.Println(curl)
}
```

PHP

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

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

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

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

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

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

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

Ruby

```
# -*- coding: UTF-8 -*-
# require ruby>=2.3.0
```

```
require 'digest'
require 'json'
require 'time'
require 'openssl'

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

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

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

```
hashed_request_payload,
].join("\n")

puts canonical_request

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

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

# ***** 步骤 4: 拼接 Authorization *****
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{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 + '"'\n+ ' -H "X-TC-Action: ' + action + '"'\n+ ' -H "X-TC-Timestamp: ' + timestamp.toString() + '"'\n+ ' -H "X-TC-Version: ' + version + '"'\n+ ' -H "X-TC-Region: ' + region + '"'\n+ " -d '" + payload + '"'\n\nconsole.log(curlcmd)\n}\n\nmain()
```

C++

```
#include <algorithm>\n#include <cstdlib>\n#include <iostream>\n#include <iomanip>\n#include <sstream>\n#include <string>\n#include <stdio.h>\n#include <time.h>\n#include <openssl/sha.h>\n#include <openssl/hmac.h>\n\nusing namespace std;\n\nstring get_data(int64_t &timestamp)\n{\n    string utcDate;\n    char buff[20] = {0};\n    // time_t timenow;\n    struct tm sttime;\n    sttime = *gmtime(&timestamp);\n    strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);\n    utcDate = string(buff);\n    return utcDate;\n}\n\nstring int2str(int64_t n)\n{\n    std::stringstream ss;\n    ss << n;\n    return ss.str();\n}
```

```
}

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

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

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

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

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

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

return (ss.str());
}

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

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

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

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

    // ***** 步骤 1: 拼接规范请求串 *****
```

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

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

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

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

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

C

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

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

void sha256_hex(const char* str, char* result)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str, strlen(str));
    SHA256_Final(hash, &sha256);
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
```

```
snprintf(buf, sizeof(buf), "%02x", hash[i]);
strcat(result, buf);
}
}

void hmac_sha256(const char* key, int key_len,
const char* input, int input_len,
unsigned char* output, unsigned int* output_len)
{
HMAC_CTX *h;
#if OPENSSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX hmac;
HMAC_CTX_init(&hmac);
h = &hmac;
#else
h = HMAC_CTX_new();
#endif

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

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

void hex_encode(const char* input, int input_len, char* output)
{
static const char* 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\", \"\\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,
&output_len);

char signature[128] = {0};
hex_encode(k_hmac_sha_sign, output_len, signature);
printf("%s\n", signature);
```

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

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

其他语言

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

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

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

签名方法

最近更新时间：2026-07-06 15:58:36

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

首次接触，建议使用 [API Explorer](#) 中的“签名串生成”功能，选择签名版本为“API 3.0 签名 v1”，可以生成签名过程进行验证，并提供了部分编程语言的签名示例，也可直接生成 SDK 代码。推荐使用腾讯云 API 配套的 8 种常见的编程语言 SDK，已经封装了签名和请求过程，均已开源，支持 [Python](#)、[Java](#)、[PHP](#)、[Go](#)、[NodeJS](#)、[.NET](#)、[C++](#)、[Ruby](#)。

推荐使用 API Explorer

<> 点击调试

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

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

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

1. 申请安全凭证

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

安全凭证包括 SecretId 和 SecretKey：

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

申请安全凭证的具体步骤如下：

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

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

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

假设用户的 SecretId 和 SecretKey 分别是：

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

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

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

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

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

2.1. 对参数排序

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

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

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

2.2. 拼接请求字符串

此步骤生成请求字符串。

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

"DescribeInstances"，因此格式化后就为 Action=DescribeInstances。

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

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

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

1. 请求方法: 支持 POST 和 GET 方式，这里使用 GET 请求，注意方法为全大写。
2. 请求主机: 查看实例列表(DescribeInstances)的请求域名为: cvm.tencentcloudapi.com。实际的请求域名根据接口所属模块的不同而不同，详见各接口说明。
3. 请求路径: 当前版本云API的请求路径固定为 /。
4. 请求字符串: 即上一步生成的请求字符串。

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

示例的拼接结果为：

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

2.4. 生成签名串

此步骤生成签名串。

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

具体代码如下，以 PHP 语言为例：

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

最终得到的签名串为：

```
7RAM2xfNMO9EiVTNmPg06MRnCvQ=
```

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

3. 签名串编码

生成的签名串并不能直接作为请求参数，需要对其进行 URL 编码。

如上一步生成的签名串为 7RAM2xfNMO9EiVTNmPg06MRnCvQ=，最终得到的签名串请求参数（Signature）为：7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D，它将用于生成最终的请求 URL。

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

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

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

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

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

5. 签名演示

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

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

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

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

- [Signature Demo](#)

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

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKID*****&Signature=7RAM2xfNMO9EiVTNmPg06MRnCvQ%3D&Timestamp=1465185768&Version=2017-03-12。`

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

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

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

Java

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

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

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

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

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

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

```
}  
  
}
```

Python

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

```
# -*- coding: utf8 -*-  
import base64  
import hashlib  
import hmac  
import os  
import time  
  
import requests  
  
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****  
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")  
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 *****  
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")  
  
def get_string_to_sign(method, endpoint, params):  
    s = method + endpoint + "?"  
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))  
    return s + query_str  
  
def sign_str(key, s, method):  
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()  
    return base64.b64encode(hmac_str)  
  
if __name__ == '__main__':  
    endpoint = "cvm.tencentcloudapi.com"  
    data = {  
        'Action': 'DescribeInstances',  
        'InstanceIds.0': 'ins-09dx96dg',  
        'Limit': 20,  
        'Nonce': 11886,  
        'Offset': 0,  
        'Region': 'ap-guangzhou',  
        'SecretId': secret_id,  
        'Timestamp': 1465185768, # int(time.time())  
        'Version': '2017-03-12'  
    }  
}
```

```
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
# resp = requests.get("https://" + endpoint, params=data)
# print(resp.url)
```

Golang

```
package main

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

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

    var buf bytes.Buffer
    buf.WriteString("GET")
    buf.WriteString("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($param);

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

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

Ruby

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

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

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

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

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

DotNet

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

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

  public static string MakeSignPlainText(SortedDictionary<string, string> requestParams, string requestMethod, string requestHost, string requestPath)
```

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

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

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

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

    param.Add("SecretId", SECRET_ID);
    param.Add("Region", region);
}
```

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

NodeJS

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

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

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

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

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

function main(){
  // 密钥参数
  // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKID*****
```

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

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

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

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

// 2. 拼接签名原字符串
const reqMethod = "GET";
const path = "/";
strSign = formatSignString(reqMethod, endpoint, path, strParam)
// console.log(strSign)

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

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

```
}  
main()
```

返回结果

最近更新时间：2026-07-06 15:58:36

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：2026-07-06 15:58:37

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。

企业安全组相关接口

创建企业安全组规则

最近更新时间：2026-07-06 15:58:29

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

规则组编辑时添加规则（规则组管理）

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：CreateSecurityGroupRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	否	String	规则组Id 示例值：RG-SEF32
Rules.N	否	Array of SecurityGroupRule	规则列表

3. 输出参数

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

4. 示例

示例1 添加规则

往已存在的规则组添加规则

输入示例

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

<公共请求参数>

```
{
  "GroupId": "fwmrg-jvw5ulal",
  "Rules": [
    {
      "IpVersion": "ipv4",
      "SourceContent": "lhins-qwgbsmg1",
      "SourceType": "instance",
      "DestContent": "0.0.0.0/0",
      "DestType": "net",
      "Protocol": "ANY",
      "Port": "-1/-1",
      "ServiceTemplateId": "",
      "RuleAction": "accept",
      "Description": "1221aaa",
      "OrderIndex": 1,
      "Scope": "SG,LH",
      "ProtocolPortType": 0,
      "BelongMemberId": "1300448058"
    },
    {
      "IpVersion": "ipv4",
      "SourceContent": "192.168.0.0/16",
      "SourceType": "net",
      "DestContent": "0.0.0.0/0",
      "DestType": "net",
      "Protocol": "ANY",
      "Port": "-1/-1",
      "ServiceTemplateId": "",
      "RuleAction": "accept",
      "Description": "232312",
      "OrderIndex": 2,
      "Scope": "SG",
      "ProtocolPortType": 0
    }
  ]
}
```

```
]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "9a66b372-86ef-488c-b53b-f93951c880a3"
  }
}
```

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。
LimitExceeded	超过配额限制。
OperationDenied	操作被拒绝。

删除企业安全组规则

最近更新时间：2026-07-06 15:58:29

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

删除规则（规则组管理）

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：DeleteSecurityGroupRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	否	String	规则组Id 示例值：RG-23JSAE
RuleIds.N	否	Array of String	规则列表 示例值：["100","102"]

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 删除规则（规则组管理）

输入示例

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

<公共请求参数>

```
{
  "GroupId": "fwmrq_hq19nybggr",
  "RuleIds": [
    "83f3be0b-c46f-4780-bf20-30dd97db5c52",
    "e8ce92d8-4ec1-4f29-8117-d7769521c751"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "7bb99beeb-c406-43ee-9072-916472427834"
  }
}
```

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。
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。
ResourceInUse	资源被占用。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnknownParameter	未知参数错误。

查询企业安全组规则详情

最近更新时间：2026-07-06 15:58:29

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询规则详情（规则组管理）

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：DescribeSecurityGroupRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	否	String	规则组Id 示例值：RG-234HK
RuleId	否	String	规则Id 示例值：120

3. 输出参数

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

4. 示例

示例1 查询规则详情（规则组管理）

输入示例

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

{
  "RuleId": "b36302b0-3a24-43d7-8d5a-2e24d6e68c89",
  "GroupId": "fwmrg-6wq0b6j0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "812242b4-c71a-4a14-a907-1f36e440c53e",
    "Rule": {
      "Detail": "1",
      "DstContent": "::/0",
      "DstPort": "-1/-1",
      "DstType": "net",
      "IpVersion": "ipv6",
      "Protocol": "ANY",
      "ProtocolPortType": 1,
      "Region": "",
      "RuleAction": "accept",
      "RuleId": "b36302b0-3a24-43d7-8d5a-2e24d6e68c89",
      "Scope": "SG",
      "Sequence": 1,
      "ServiceTemplateId": "ppm-r7gw5s50",
      "SrcContent": "2402:4e00:c032:8800:be66:c9b8:2454:0",
      "SrcType": "net"
    }
  }
}
```

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。
ResourceNotFound	资源不存在。
ResourceNotFound.NotFound	ResourceNotFound.NotFound

查询企业安全组规则列表

最近更新时间：2026-07-06 15:58:28

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询规则组中规则列表接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeSecurityGroupRules。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	否	String	规则组Id 示例值：rg-2jsdas
Filters.N	否	Array of CommonFilter	模糊搜索关键词
Offset	否	Integer	偏移量 示例值：0
Limit	否	Integer	数量 示例值：20

3. 输出参数

参数名称	类型	描述
Rules	Array of SecGroupRuleResp	规则列表
TotalCount	Integer	总数

参数名称	类型	描述
		示例值：20
AllTotalCount	Integer	不过滤的规则总数 示例值：22
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 查询规则组中规则列表接口

输入示例

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

<公共请求参数>

```
{
  "GroupId": "fwmrg-i3pvf19a",
  "Filters": [
    {
      "Name": "RuleId",
      "Values": [
        "6eaf61e"
      ],
      "OperatorType": 9
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "AllTotalCount": 7,
    "RequestId": "d1f4711c-a542-4b7f-9443-7fd31437edd4",
    "Rules": [
      {
```

```
"BelongMember": {
  "AppId": "1300448058",
  "MemberId": "mem-tencent-e4f32cfd87465a17",
  "Nickname": "天空之蓝",
  "Uin": "100011616646"
},
"Cidr": "",
"Detail": "1",
"DnsParseCount": {
  "InvalidCount": 0,
  "ValidCount": 0
},
"Id": 7884,
"InstanceName": "",
"IpVersion": "ipv4",
"OrderIndex": 7,
"ParameterName": "",
"Port": "-1/-1",
"PrivateIp": "",
"Protocol": "ANY",
"ProtocolPortName": "",
"PublicIp": "",
"Region": "",
"RuleId": "6eaf61e3-2f5f-435c-b3d1-c39dae46a7de",
"Scope": "SG",
"ServiceTemplateId": "",
"SouCidr": "",
"SouInstanceName": "",
"SouParameterName": "",
"SouPrivateIp": "",
"SouPublicIp": "",
"SourceId": "0.0.0.0/0",
"SourceType": 0,
"Strategy": 2,
"TargetId": "0.0.0.0/0",
"TargetType": 0
}
],
"TotalCount": 1
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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。

查询规则下发状态

最近更新时间：2026-07-06 15:58:28

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询策略下发状态

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：DescribeStrategyDispatchStatus。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Product	否	String	产品 示例值：secgroup

3. 输出参数

参数名称	类型	描述
Progress	Float	进度 示例值：0.9
StartTime	String	下发开始时间 示例值：2025-10-30 10:00:00
EndTime	String	下发结束时间 示例值：2025-10-30 10:10:00
Status	Integer	下发状态，0无变动，1下发中，2下发成功，3下发失败，4更新待下发 示例值：1

参数名称	类型	描述
RuleGroupNum	Integer	下发规则组数量 示例值：3
ErrorMsg	String	下发失败错误信息 示例值：配额不足
DispatchStrategyList	Array of String	下发关联策略id列表 示例值：["fwmsid_lhdp0plbic"]
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 查询策略下发状态

输入示例

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

{
  "Product": "enterprise_sg"
}
```

输出示例

```
{
  "Response": {
    "DispatchStrategyList": [
      "fwms-kmbf8avt",
      "fwms-ms91jjvy",
      "fwms-vg9jilr5"
    ],
    "EndTime": "2026-02-27 19:52:50",
    "ErrorMsg": "",
    "Progress": 100,
    "RequestId": "77659938-d469-4313-ba83-232b66559e1e",
    "RuleGroupNum": 3,
  }
}
```

```
"StartTime": "2026-02-27 19:52:35",  
"Status": 2  
}  
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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](#)。

忽略策略问题

最近更新时间：2026-07-06 15:58:28

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

忽略策略问题

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：IgnorePolicyRisk。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
RiskId	是	String	策略问题ID 示例值：0001
MemberId	否	String	成员Id 示例值：mem-tencent-e4f32cfd87465a17

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 忽略策略问题

输入示例

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

{
  "RiskId": "ff9d17c245741c74e1019d26a60ac6b3"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "68c302a8-fe7d-4235-836d-b543f3f49ac7"
  }
}
```

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。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

修改企业安全组规则

最近更新时间：2026-07-06 15:58:27

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

修改规则（规则组管理）

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：ModifySecurityGroupRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	否	String	规则组ID 示例值：RG-2KSFE
Rule	否	SecurityGroupRule	规则

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 修改规则组内规则

输入示例

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

{
  "GroupId": "fwmrj-gjvw5ulal",
  "Rule": {
    "IpVersion": "ipv4",
    "SourceContent": "ins-3nav5p56",
    "SourceType": "instance",
    "DestContent": "0.0.0.0/0",
    "DestType": "net",
    "Protocol": "ANY",
    "Port": "-1/-1",
    "ServiceTemplateId": "",
    "RuleAction": "accept",
    "Description": "33333333",
    "OrderIndex": 1,
    "RuleId": "7394c937-85f9-4b0f-a30f-494d8369e4bd",
    "Scope": "SG",
    "ProtocolPortType": 0,
    "BelongMemberId": "1302477603"
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "dc7591e4-8fc8-417d-b355-07e7df3e8a19"
  }
}
```

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.CommonParam	MissingParameter.CommonParam
ResourceInUse	资源被占用。
ResourceNotFound.NotFound	ResourceNotFound.NotFound

通用相关接口

查询防火墙管理规则下发账号

最近更新时间：2026-07-06 15:58:20

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查看防火墙管理规则下发账号列表

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：DescribeStrategyAccounts。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Product	是	String	下发产品 secgroup // 企业安全组 示例值：secgroup
Filters.N	否	Array of CommonFilter	筛选器

3. 输出参数

参数名称	类型	描述
Accounts	Array of Account	账号列表
AccountGroups	Array of AccountGroupQuotaDetail	账号组列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得

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 查询防火墙管理规则下发账号

输入示例

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

{
  "Product": "enterprise_sg"
}
```

输出示例

```
{
  "Response": {
    "AccountGroups": [
      {
        "BottleneckUin": "100011949846",
        "DispatchRuleNum": 0,
        "GroupId": "acg-dudjfm6t",
        "GroupName": "测试1",
        "MemberCount": 3,
        "Members": [
          {
            "DispatchRuleNum": 0,
            "MemberId": "mem-1300846651-100011949846",
            "Nickname": "焦糖小蛋糕",
            "OriginRuleNum": 0,
            "RemainQuota": 0,
            "TotalQuota": 0,
            "Uin": "100011949846"
          }
        ],
      }
    ],
  }
}
```

```
"DispatchRuleNum": 0,
"MemberId": "mem-1300846651-100028354982",
"Nickname": "少年时",
"OriginRuleNum": 0,
"RemainQuota": 0,
"TotalQuota": 0,
"Uin": "100028354982"
},
{
"DispatchRuleNum": 0,
"MemberId": "mem-1300846651-100037509558",
"Nickname": "fengqqian",
"OriginRuleNum": 0,
"RemainQuota": 0,
"TotalQuota": 0,
"Uin": "100037509558"
}
],
"OriginRuleNum": 0,
"RemainQuota": 0,
"TotalQuota": 0
},
{
"BottleneckUin": "100028671395",
"DispatchRuleNum": 0,
"GroupId": "acg-ssbkqes9",
"GroupName": "生产测试组",
"MemberCount": 1,
"Members": [
{
"DispatchRuleNum": 0,
"MemberId": "mem-1300846651-100028671395",
"Nickname": "美式",
"OriginRuleNum": 0,
"RemainQuota": 0,
"TotalQuota": 0,
"Uin": "100028671395"
}
],
"OriginRuleNum": 0,
"RemainQuota": 0,
"TotalQuota": 0
},
```

```
{
  "BottleneckUin": "",
  "DispatchRuleNum": 0,
  "GroupId": "acg-na418213",
  "GroupName": "测试2",
  "MemberCount": 0,
  "Members": [],
  "OriginRuleNum": 0,
  "RemainQuota": 0,
  "TotalQuota": 0
},
{
  "BottleneckUin": "",
  "DispatchRuleNum": 0,
  "GroupId": "acg-uvrarqih",
  "GroupName": "<img src='' onerror='alert(\"cdasd\")' />",
  "MemberCount": 0,
  "Members": [],
  "OriginRuleNum": 0,
  "RemainQuota": 0,
  "TotalQuota": 0
}
],
"Accounts": [
  {
    "DispatchRuleNum": 0,
    "MemberId": "mem-1300846651-100027980407",
    "Nickname": "2468639187",
    "OriginRuleNum": 0,
    "RemainQuota": 0,
    "TotalQuota": 0,
    "Uin": "100027980407"
  },
  {
    "DispatchRuleNum": 0,
    "MemberId": "mem-1300846651-100028354982",
    "Nickname": "少年时",
    "OriginRuleNum": 0,
    "RemainQuota": 0,
    "TotalQuota": 0,
    "Uin": "100028354982"
  },
  {
```

```
"DispatchRuleNum": 0,
"MemberId": "mem-1300846651-100028671395",
"Nickname": "美式",
"OriginRuleNum": 0,
"RemainQuota": 0,
"TotalQuota": 0,
"Uin": "100028671395"
},
{
"DispatchRuleNum": 0,
"MemberId": "mem-1300846651-100037509558",
"Nickname": "fengqqian",
"OriginRuleNum": 0,
"RemainQuota": 0,
"TotalQuota": 0,
"Uin": "100037509558"
},
{
"DispatchRuleNum": 0,
"MemberId": "mem-1300846651-100011949846",
"Nickname": "焦糖小蛋糕",
"OriginRuleNum": 0,
"RemainQuota": 0,
"TotalQuota": 0,
"Uin": "100011949846"
},
{
"DispatchRuleNum": 0,
"MemberId": "mem-1300846651-100011616646",
"Nickname": "天空之蓝",
"OriginRuleNum": 17,
"RemainQuota": 1983,
"TotalQuota": 2000,
"Uin": "100011616646"
}
],
"RequestId": "4b6c5bd5-c696-4ce8-a4c0-be583552aaaa"
}
}
```

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。

多账号管理相关接口

查询集团成员列表

最近更新时间：2026-07-06 15:58:27

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询集团下所有纳管成员账号列表，支持分页、排序和多条件筛选，仅管理员可调用

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeOrganMembers。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Filters.N	否	Array of CommonFilter	搜索过滤条件列表，支持按成员 ID、账号名称、身份、纳管状态等字段筛选
Limit	否	Integer	分页大小，默认 20 示例值：20
Offset	否	Integer	分页偏移量，默认 0 示例值：0
By	否	String	排序字段，如 MemberCreateTime 示例值：MemberCreateTime
Order	否	String	排序方式：asc 升序，desc 降序 示例值：desc

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	集团成员总数 示例值：10
Members	Array of OrganMemberItem	集团成员列表
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 查询集团成员列表

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "Members": [
      {
        "AccountGroup": null,
        "AppId": "1300846651",
        "CfwInstanceId": "",
        "CfwManaged": 0,
        "CfwShareRole": "none",
        "CfwShareRoleDisplay": "none",
        "CfwSharerAppId": "",
        "MemberCreateTime": "2022-10-20 15:38:57",
        "MemberId": "mem-1300846651-100011949846",
        "Nickname": "焦糖小蛋糕",
        "NodeName": "Root",
        "PolicyAnalysisEnabled": 0,

```

```
"Role": "admin",
"RoleDisplay": "admin",
"SubAccountCount": 0
},
{
"AccountGroup": {
"GroupId": "acg-rwqqudnm",
"GroupName": "生产账号组1组"
},
"AppId": "1300448058",
"CfwInstanceId": "",
"CfwManaged": 0,
"CfwShareRole": "none",
"CfwShareRoleDisplay": "none",
"CfwSharerAppId": "",
"MemberCreateTime": "2025-12-22 20:46:59",
"MemberId": "mem-1300846651-100011616646",
"Nickname": "天空之蓝",
"NodeName": "Root",
"PolicyAnalysisEnabled": 0,
"Role": "delegatedAdmin",
"RoleDisplay": "delegatedAdmin",
"SubAccountCount": 0
},
{
"AccountGroup": null,
"AppId": "1327492904",
"CfwInstanceId": "",
"CfwManaged": 0,
"CfwShareRole": "none",
"CfwShareRoleDisplay": "none",
"CfwSharerAppId": "",
"MemberCreateTime": "2024-06-24 14:51:35",
"MemberId": "mem-1300846651-100037509558",
"Nickname": "fengqqian",
"NodeName": "Root",
"PolicyAnalysisEnabled": 0,
"Role": "delegatedAdmin",
"RoleDisplay": "delegatedAdmin",
"SubAccountCount": 0
},
{
"AccountGroup": null,
```

```
"AppId": "1314933167",
"CfwInstanceId": "",
"CfwManaged": 0,
"CfwShareRole": "none",
"CfwShareRoleDisplay": "none",
"CfwSharerAppId": "",
"MemberCreateTime": "2022-11-09 15:09:35",
"MemberId": "mem-1300846651-100028354982",
"Nickname": "少年时",
"NodeName": "Root",
"PolicyAnalysisEnabled": 0,
"Role": "delegatedAdmin",
"RoleDisplay": "delegatedAdmin",
"SubAccountCount": 0
},
{
"AccountGroup": null,
"AppId": "1315399711",
"CfwInstanceId": "",
"CfwManaged": 0,
"CfwShareRole": "none",
"CfwShareRoleDisplay": "none",
"CfwSharerAppId": "",
"MemberCreateTime": "2022-11-26 19:29:43",
"MemberId": "mem-1300846651-100028671395",
"Nickname": "美式",
"NodeName": "Root",
"PolicyAnalysisEnabled": 0,
"Role": "member",
"RoleDisplay": "member",
"SubAccountCount": 0
},
{
"AccountGroup": null,
"AppId": "1314503630",
"CfwInstanceId": "",
"CfwManaged": 0,
"CfwShareRole": "none",
"CfwShareRoleDisplay": "none",
"CfwSharerAppId": "",
"MemberCreateTime": "2022-10-20 17:04:50",
"MemberId": "mem-1300846651-100027980407",
"Nickname": "测试部门",
```

```
"NodeName": "测试的部门1号",
"PolicyAnalysisEnabled": 0,
"Role": "member",
"RoleDisplay": "member",
"SubAccountCount": 0
},
],
"RequestId": "c4feaf81-782f-43cb-9053-4a5982e84a15",
"TotalCount": 6
}
}
```

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
UnauthorizedOperation	未授权操作。

获取集团概览

最近更新时间：2026-07-06 15:58:27

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

获取集团概览信息，包括集团名称、管理员信息、成员数量等

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：DescribeOrganSummary。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。

3. 输出参数

参数名称	类型	描述
Summary	OrganSummary	集团概览
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 查询集团概览

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "RequestId": "ac3b6531-d7f0-4752-86af-962359fa7ad5",
    "Summary": {
      "AdminCount": 4,
      "AdminInfo": {
        "AppId": "1300846651",
        "MemberId": "mem-1300846651-100011949846",
        "Nickname": "焦糖小蛋糕",
        "Uin": "100011949846"
      },
      "CfwSharerCount": 1,
      "CfwUserCount": 1,
      "GroupName": "焦糖小蛋糕",
      "JoinedMemberCount": 6,
      "MemberLimit": -1,
      "MemberLimitDisplay": "无上限"
    }
  }
}
```

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
UnauthorizedOperation	未授权操作。

互联网边界规则相关接口

创建互联网边界规则

最近更新时间：2026-07-06 15:58:31

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

向已有的互联网边界ACL规则组中添加规则。需要先创建规则组，然后通过此接口添加规则。

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：CreateEdgeAclRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	否	String	规则组ID 示例值：fwmrg_abc123xyz
Rules.N	否	Array of EdgeAclRuleInfo	规则列表

3. 输出参数

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

4. 示例

示例1 创建互联网边界规则

输入示例

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

{
  "GroupId": "fwmrq_4ae6vv117c",
  "Rules": [
    {
      "OrderIndex": 1,
      "Direction": 1,
      "SourceContent": "0.0.0.0/0",
      "SourceType": "net",
      "TargetContent": "0.0.0.0/0",
      "TargetType": "net",
      "Port": "-1/-1",
      "Protocol": "TCP",
      "RuleAction": "accept",
      "Description": "1212322",
      "Scope": "all"
    },
    {
      "OrderIndex": 2,
      "Direction": 1,
      "SourceContent": "bj11",
      "SourceType": "location",
      "TargetContent": "cfwnat-e366d270",
      "TargetType": "instance",
      "Port": "-1/-1",
      "Protocol": "TCP",
      "RuleAction": "accept",
      "Description": "3232111",
      "Scope": "all",
      "BelongMemberId": "1300448058"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1a4f32a7-ae1-4049-9fe0-4ceeb07f55ce"
  }
}
```

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。
ResourceInsufficient	资源不足。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

删除互联网边界规则

最近更新时间：2026-07-06 15:58:31

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

批量删除互联网边界ACL规则。支持一次删除多条规则。

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：DeleteEdgeAclRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	否	String	规则组ID 示例值：fwmrg_abc123xyz
RuleIds.N	否	Array of String	要删除的规则ID列表 示例值：["rule_001", "rule_002"]

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 删除互联网边界规则

输入示例

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

{
  "GroupId": "fwmrg_4ae6vv117c",
  "RuleIds": [
    "7f8d9162-9c3d-4b1f-b3ed-c63e1ce583cb",
    "f5e7b8f7-ed34-489f-a7ce-2f7dc115843e"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3641dba9-c98d-41e5-b128-13a8140adf8f"
  }
}
```

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。
ResourceInsufficient	资源不足。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

查询互联网边界规则列表

最近更新时间：2026-07-06 15:58:30

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询指定规则组下的互联网边界ACL规则列表。支持分页和多种过滤条件。

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：DescribeEdgeAclRules。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrg_abc123xyz
Direction	否	Integer	规则方向：0-出向，1-入向 示例值：1
Filters.N	否	Array of CommonFilter	过滤条件列表，支持按 RuleId、Direction、Protocol、RuleAction 等字段过滤
Limit	否	Integer	分页大小，默认100，最大1000 示例值：100
Offset	否	Integer	分页偏移，默认0 示例值：0
Order	否	String	排序顺序，asc:升序 desc:降序 示例值：asc
By	否	String	排序字段，支持 Sequence、RuleId 等 示例值：Sequence

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	规则总数 示例值：10
Rules	Array of EdgeAclRuleResp	规则列表
AllTotalCount	Integer	不过滤的总数 示例值：21
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 查询互联网边界规则列表

输入示例

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

{
  "GroupId": "fwmrg-ww0m4x6493",
  "Direction": 1,
  "Filters": [
    {
      "Name": "SourceContent",
      "Values": [
        "河北"
      ],
      "OperatorType": 9
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "AllTotalCount": 52,
    "RequestId": "d33c7277-5e74-4e10-b5c7-8eea9b4cc1d8",
    "Rules": [
      {
        "CityCode": 2147483647,
        "CityKey": "hb13,sx14,ln21,jl22,hlj23,js32,zj33,ah34,fj35,jx36,sd37,hn41,hb42,hn43,gd44,hn46,sc51,gz52,yn53,sx61,gs62,qh63,nmg15,gx45,xz54,nx64,xj65,bj11,tj12,sh31,cq50",
        "CityName": "河北省,山西省,辽宁省,吉林省,黑龙江省,江苏省,浙江省,安徽省,福建省,江西省,山东省,河南省,湖北省,湖南省,广东省,海南省,四川省,贵州省,云南省,陕西省,甘肃省,青海省,内蒙古自治区,广西壮族自治区,西藏自治区,宁夏回族自治区,新疆维吾尔自治区,北京市,天津市,上海市,重庆市",
        "CloudCode": "",
        "CountryCode": 1,
        "CountryKey": "zd",
        "CountryName": "全部中国地区",
        "CreateTime": "2026-04-15 17:04:00",
        "Description": "we",
        "Direction": 1,
        "DnsParseCnt": 0,
        "InstanceName": "",
        "Invalid": 0,
        "IsCloud": 0,
        "IsRegion": 1,
        "ParamTemplateId": "",
        "ParamTemplateName": "",
        "Port": "-1/-1",
        "Protocol": "HTTPS",
        "RuleAction": "accept",
        "RuleId": "4e020c5b-d7b4-46e1-93bf-c30f1a2f8275",
        "Scope": "serial",
        "Sequence": 8,
        "SourceContent": "0.0.0.0/0",
        "SourceName": "",
        "SourceType": "net",
        "TargetContent": "qq.com",
        "TargetName": "",
        "TargetType": "domain",
        "UpdateTime": "2026-04-16 16:16:42"
      },
      {
        "CityCode": 1,
```

```
"CityKey": "hb13",
"CityName": "河北省",
"CloudCode": "",
"CountryCode": 1,
"CountryKey": "",
"CountryName": "",
"CreateTime": "2026-04-15 15:59:39",
>Description": "23",
"Direction": 1,
"DnsParseCnt": 0,
"InstanceName": "",
"Invalid": 0,
"IsCloud": 0,
"IsRegion": 1,
"ParamTemplateId": "",
"ParamTemplateName": "",
"Port": "-1/-1",
"Protocol": "TCP",
"RuleAction": "accept",
"RuleId": "91092f4e-2d8a-4f33-a53b-7a6aae6f2bd5",
"Scope": "all",
"Sequence": 24,
"SourceContent": "0.0.0.0/0",
"SourceName": "",
"SourceType": "net",
"TargetContent": "0.0.0.0/0",
"TargetName": "",
"TargetType": "net",
"UpdateTime": "2026-04-16 16:16:42"
}
],
"TotalCount": 2
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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。
ResourceInsufficient	资源不足。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

修改互联网边界规则

最近更新时间：2026-07-06 15:58:30

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

修改互联网边界ACL规则。Rule 参数中必须包含 RuleId 用于指定要修改的规则。

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：ModifyEdgeAclRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	否	String	规则组ID 示例值：fwmrg_abc123xyz
Rule	否	EdgeAclRuleInfo	要修改的规则，必须包含RuleId

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 修改互联网边界规则

输入示例

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

{
  "GroupId": "fwmrj_4ae6vv117c",
  "Rule": {
    "RuleId": "fb65eefd-6a64-4c02-a720-e278d41df8bd",
    "OrderIndex": 3,
    "Direction": 1,
    "SourceContent": "0.0.0.0/0",
    "SourceType": "net",
    "TargetContent": "0.0.0.0/0",
    "TargetType": "net",
    "Port": "-1/-1",
    "Protocol": "TCP",
    "RuleAction": "drop",
    "Description": "77777",
    "Scope": "all"
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "6664b107-391d-473d-a59a-88502841fa0e"
  }
}
```

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。
ResourceInsufficient	资源不足。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

修改互联网边界规则序号

最近更新时间：2026-07-06 15:58:30

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

批量调整互联网边界ACL规则的执行顺序。Sequences 参数必须包含所有受影响的规则序号映射关系。

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：ModifyEdgeAclRuleSequence。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Direction	是	Integer	出入站方向 0=出向，1=入向 示例值：1
GroupId	否	String	规则组ID 示例值：fwmg_abc123xyz
Sequences.N	否	Array of SequenceIndex	规则序号调整列表，必须包含所有受影响的规则

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 互联网边界规则组内规则快速排序

输入示例

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

```
{
  "Direction": 1,
  "GroupId": "fwmrg_4ae6vv117c",
  "Sequences": [
    {
      "OrderIndex": 1,
      "NewOrderIndex": 2
    },
    {
      "OrderIndex": 2,
      "NewOrderIndex": 3
    },
    {
      "OrderIndex": 3,
      "NewOrderIndex": 4
    },
    {
      "OrderIndex": 4,
      "NewOrderIndex": 1
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "025c3756-bf58-408e-b806-2e68ead7f6a0"
  }
}
```

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。
ResourceInsufficient	资源不足。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

NAT边界规则相关接口

创建NAT边界规则

最近更新时间：2026-07-06 15:58:35

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

在已有规则组中添加NAT ACL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：CreateNatAclRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrg_abc123xyz
Rules.N	是	Array of NatAclRule	规则列表

3. 输出参数

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

4. 示例

示例1 添加 NAT 规则

输入示例

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

{
  "GroupId": "fwmrg_go7u11gs6v",
  "Rules": [
    {
      "SourceContent": "0.0.0.0/0",
      "SourceType": "net",
      "TargetContent": "0.0.0.0/0",
      "TargetType": "net",
      "Protocol": "UDP",
      "RuleAction": "accept",
      "OrderIndex": 2,
      "Scope": "cfwnat-e7ad7c24",
      "Direction": 1,
      "Port": "-1/-1",
      "Description": "444",
      "ParamTemplateId": "",
      "BelongMemberId": "1300448058"
    },
    {
      "SourceContent": "0.0.0.0/0",
      "SourceType": "net",
      "TargetContent": "0.0.0.0/0",
      "TargetType": "net",
      "Protocol": "ANY",
      "RuleAction": "accept",
      "OrderIndex": 3,
      "Scope": "ALL",
      "Direction": 1,
      "Port": "-1/-1",
      "Description": "555",
      "ParamTemplateId": ""
    },
    {
      "SourceContent": "0.0.0.0/0",
      "SourceType": "net",
```

```
"TargetContent": "0.0.0.0/0",
"TargetType": "net",
"Protocol": "ANY",
"RuleAction": "accept",
"OrderIndex": 1,
"Scope": "ALL",
"Direction": 0,
"Port": "-1/-1",
"Description": "222",
"ParamTemplateId": ""
}
]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3deaf1ac-6319-4eb6-a989-0b9a13054616"
  }
}
```

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。

删除NAT边界规则

最近更新时间：2026-07-06 15:58:34

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

删除NAT ACL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：DeleteNatAclRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrg_abc123xyz
RuleIds.N	是	Array of String	规则ID列表 示例值：["uuid-1","uuid-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 删除 NAT 规则

输入示例

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

{
  "GroupId": "fwmrg_wrzekq94u7",
  "RuleIds": [
    "50d186fd-6da7-4de3-acbc-8efea0272c76",
    "3196fa67-1f2f-46a1-b341-75edfa5094d3"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "32178aa0-b534-4c2f-ac11-818765fb4179"
  }
}
```

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。

查询NAT边界规则列表

最近更新时间：2026-07-06 15:58:34

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询NAT ACL规则列表

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：DescribeNatAclRules。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrq_abc123xyz
Direction	否	Integer	规则方向：0-出向，1-入向 示例值：1
Filters.N	否	Array of CommonFilter	过滤条件
Offset	否	Integer	偏移量 示例值：0
Limit	否	Integer	数量 示例值：20
Order	否	String	排序顺序，asc:升序 desc:降序 示例值：asc
By	否	String	排序字段 示例值：Sequence

3. 输出参数

参数名称	类型	描述
Rules	Array of NatAclRuleResp	规则列表
TotalCount	Integer	总数 示例值：20
AllTotalCount	Integer	不过滤的总数 示例值：21
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 NAT 规则列表

输入示例

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

{
  "GroupId": "fwmrg_9tnz3ksn7v",
  "Direction": 0,
  "Filters": [
    {
      "Name": "TargetContent",
      "Values": [
        "河北省"
      ],
      "OperatorType": 9
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "AllTotalCount": 10,
    "RequestId": "23fd42d9-7a67-4a0a-b6f1-e598451d798d",
    "Rules": [
      {
        "CityCode": 2146303999,
        "CityKey": "hb13,sx14,ln21,jl22,hlj23,js32,zj33,ah34,fj35,jx36,sd37,hn41,hb42,hn43,gd44,hn46,sc51,yn53,sx61,qh63,nmg15,gx45,xz54,nx64,xj65,bj11,tj12,sh31,cq50",
        "CityName": "河北省,山西省,辽宁省,吉林省,黑龙江省,江苏省,浙江省,安徽省,福建省,江西省,山东省,河南省,湖北省,湖南省,广东省,海南省,四川省,云南省,陕西省,青海省,内蒙古自治区,广西壮族自治区,西藏自治区,宁夏回族自治区,新疆维吾尔自治区,北京市,天津市,上海市,重庆市",
        "CountryCode": 1,
        "CreateTime": "2026-04-02 17:16:53",
        "Description": "3412",
        "Direction": 0,
        "DnsParseCnt": 0,
        "IsRegion": 1,
        "ParamTemplateId": "",
        "ParamTemplateName": "",
        "Port": "-1/-1",
        "Protocol": "ANY",
        "RuleAction": "accept",
        "RuleId": "ba37feba-4ff5-46c7-b876-77bbc456cdc7",
        "Scope": "ALL",
        "ScopeDesc": "全局规则",
        "Sequence": 8,
        "SourceContent": "43.143.218.25",
        "SourceType": "net",
        "TargetContent": "0.0.0.0/0",
        "TargetType": "net",
        "UpdateTime": "2026-04-07 18:15:05"
      }
    ],
    "TotalCount": 1
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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NAT边界规则

最近更新时间：2026-07-06 15:58:34

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

修改NAT ACL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：ModifyNatAclRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrg_abc123xyz
Rule	是	NatAclRule	规则

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 NAT 规则

输入示例

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

{
  "GroupId": "fwmrg_go7u11gs6v",
  "Rule": {
    "SourceContent": "0.0.0.0/0",
    "SourceType": "net",
    "TargetContent": "0.0.0.0/0",
    "TargetType": "net",
    "Protocol": "UDP",
    "RuleAction": "accept",
    "OrderIndex": 2,
    "Scope": "cfwnat-e366d270",
    "Direction": 1,
    "RuleId": "5712589f-c85c-4b3e-8917-bc3bb5779133",
    "Port": "-1/-1",
    "Description": "444",
    "ParamTemplateId": "",
    "BelongMemberId": "1300448058"
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1b4f862f-f14a-47e9-8d4d-9ff400989ed4"
  }
}
```

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。

NAT边界规则排序

最近更新时间：2026-07-06 15:58:33

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

调整NAT ACL规则优先级顺序

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：ModifyNatAclRuleSequence。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrg_abc123xyz
Sequences.N	是	Array of SequenceIndex	序号调整列表
Direction	是	Integer	规则方向：1-进站规则，0-出站规则 示例值：1

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 NAT 规则组内快速排序

输入示例

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

<公共请求参数>

```
{
  "GroupId": "fwmrg_wrzekq94u7",
  "Sequences": [
    {
      "OrderIndex": 1,
      "NewOrderIndex": 3
    },
    {
      "OrderIndex": 2,
      "NewOrderIndex": 1
    },
    {
      "OrderIndex": 3,
      "NewOrderIndex": 2
    }
  ],
  "Direction": 1
}
```

输出示例

```
{
  "Response": {
    "RequestId": "7d8a3017-26bb-4d73-b2e3-58ca0443bb3b"
  }
}
```

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。

VPC边界规则相关接口

创建VPC边界规则

最近更新时间：2026-07-06 15:58:33

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

在已有规则组中添加VPC ACL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：CreateVpcAclRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrg_abc123xyz
Rules.N	是	Array of VpcAclRule	规则列表

3. 输出参数

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

4. 示例

示例1 添加VPC 规则

输入示例

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

{
  "GroupId": "fwmrq_aussbmarr2",
  "Rules": [
    {
      "SourceContent": "0.0.0.0/0",
      "SourceType": "net",
      "DestContent": "0.0.0.0/0",
      "DestType": "net",
      "Protocol": "ANY",
      "RuleAction": "accept",
      "OrderIndex": 1,
      "EdgeId": "all",
      "FwGroupId": "ccn-c0qmm031",
      "Port": "-1/-1",
      "Description": "zero",
      "ParamTemplateId": "",
      "BelongMemberId": "1300448058"
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "c4b0ec1c-82a7-4781-b73e-86c0623187b6"
  }
}
```

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。

删除VPC边界规则

最近更新时间：2026-07-06 15:58:33

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

删除VPC ACL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：DeleteVpcAclRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrg_abc123xyz
RuleIds.N	是	Array of String	规则ID列表 示例值：["uuid-1","uuid-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 删除VPC 规则

输入示例

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

{
  "GroupId": "fwmrj_aussbmarr2",
  "RuleIds": [
    "e8314eac-7313-4be5-83fc-1a2e05362f1d",
    "1293a7fa-b6ee-4243-9429-8e690ce92057"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "6c4dc364-63dc-4ed1-9c17-b2dba1c9e1de"
  }
}
```

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。

查询VPC边界规则列表

最近更新时间：2026-07-06 15:58:32

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询VPC ACL规则列表

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：DescribeVpcAclRules。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrg_abc123xyz
Filters.N	否	Array of CommonFilter	过滤条件
Offset	否	Integer	偏移量 示例值：0
Limit	否	Integer	数量 示例值：20

3. 输出参数

参数名称	类型	描述
Rules	Array of VpcAclRuleResp	规则列表
TotalCount	Integer	总数

参数名称	类型	描述
		示例值：20
AllTotalCount	Integer	不过滤的总数 示例值：21
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VPC 规则列表

输入示例

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

<公共请求参数>

```
{
  "GroupId": "fwmrg-yltolsxma7",
  "Filters": [
    {
      "Name": "RuleId",
      "Values": [
        "2ceddffe"
      ],
      "OperatorType": 9
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "AllTotalCount": 3,
    "RequestId": "f7d0cafa-6f06-49f5-862f-bf68897a762f",
    "Rules": [
      {
```

```
"CreateTime": "2026-04-13 20:34:47",
"Description": "1212",
"DestContent": "::/0",
"DestType": "net",
"DnsParseCnt": 0,
"EdgeId": "ALL",
"FwGroupId": "ALL",
"IpVersion": "ipv6",
"ParamTemplateId": "",
"Port": "-1/-1",
"Protocol": "ANY",
"RuleAction": "accept",
"RuleId": "2ceddffe-a328-4bdf-af5f-a02821fab173",
"Sequence": 3,
"SourceContent": "::/0",
"SourceType": "net",
"UpdateTime": "2026-04-15 11:05:06"
}
],
"TotalCount": 1
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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VPC边界规则

最近更新时间：2026-07-06 15:58:32

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

修改VPC ACL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：ModifyVpcAclRule。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrg_abc123xyz
Rule	是	VpcAclRule	规则

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VPC 规则

输入示例

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

{
  "GroupId": "fwmrg_aussbmarr2",
  "Rule": {
    "SourceContent": "10.10.0.0/16",
    "SourceType": "net",
    "DestContent": "0.0.0.0/0",
    "DestType": "net",
    "Protocol": "ANY",
    "RuleAction": "drop",
    "OrderIndex": 2,
    "EdgeId": "all",
    "FwGroupId": "ccn-c0qmm031",
    "RuleId": "1293a7fa-b6ee-4243-9429-8e690ce92057",
    "Port": "-1/-1",
    "Description": "two",
    "ParamTemplateId": "",
    "BelongMemberId": "1300448058"
  }
}
```

输出示例

```
{
  "Response": {
    "RequestId": "0c3001bf-d642-4dae-8062-b5cd606238fd"
  }
}
```

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。

VPC边界规则排序

最近更新时间：2026-07-06 15:58:32

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

调整VPC ACL规则优先级顺序

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：ModifyVpcAclRuleSequence。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	是	String	规则组ID 示例值：fwmrg_abc123xyz
Sequences.N	是	Array of SequenceIndex	序号调整列表

3. 输出参数

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

4. 示例

示例1 VPC 规则组内规则快速排序

输入示例

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

<公共请求参数>

```
{
  "GroupId": "fwmrg_aussbmarr2",
  "Sequences": [
    {
      "OrderIndex": 1,
      "NewOrderIndex": 3
    },
    {
      "OrderIndex": 2,
      "NewOrderIndex": 1
    },
    {
      "OrderIndex": 3,
      "NewOrderIndex": 2
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3034a6c8-1bc4-4e4e-b69b-a11b1468b79e"
  }
}
```

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。

策略分析相关接口

查询账号+产品维度风险统计

最近更新时间：2026-07-06 15:58:26

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询账号+产品维度风险统计，按账号分组返回各产品的体检策略数、待整改风险数、整改率、最近体检时间等信息，支持按账号名称/ID搜索以及仅看待整改、仅超时未体检筛选

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：DescribePolicyRiskAccountProductStats。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Limit	否	Integer	分页大小，按账号分页，默认20，最大100 示例值：20
Offset	否	Integer	分页偏移，默认0 示例值：0
Filters.N	否	Array of CommonFilter	筛选条件列表。支持的筛选字段：AccountName（账号名称模糊搜索）、AccountId（账号Uin精确搜索）、OnlyUntreated（仅看待整改，值为1时生效）、OnlyOverdue（仅超时未体检，值为1时生效）

3. 输出参数

参数名称	类型	描述
AccountStats	Array of AccountStatsGroup	按账号分组的风险统计列表 注意：此字段可能返回 null，表示取不到有效值。

参数名称	类型	描述
TotalCount	Integer	满足条件的账号总数 示例值：10
OverdueAccountCount	Integer	超时未体检的账号数 示例值：2
OverdueProductCount	Integer	超时未体检的产品数 示例值：2
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 查询仅待整改

输入示例

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

<公共请求参数>

```
{
  "Filters": [
    {
      "Name": "OnlyUntreated",
      "Values": [
        "1"
      ],
      "OperatorType": 7
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "AccountStats": [
```

```
{
  "Member": {
    "AppId": "1300448058",
    "MemberId": "mem-1300846651-100011616646",
    "Nickname": "天空之蓝",
    "Uin": "100011616646"
  },
  "ProductStats": [
    {
      "IgnoredRiskCount": 55,
      "LastCheckTime": "2026-03-26 15:29:52",
      "PolicyCount": 1151,
      "Product": "sg",
      "ProductName": "安全组",
      "RectifyRate": "0%",
      "TotalRiskCount": 152,
      "TreatedRiskCount": 0,
      "UntreatedRiskCount": 97
    },
    {
      "IgnoredRiskCount": 0,
      "LastCheckTime": "2026-03-09 19:42:49",
      "PolicyCount": 34,
      "Product": "enterprise_sg",
      "ProductName": "企业安全组",
      "RectifyRate": "0%",
      "TotalRiskCount": 4,
      "TreatedRiskCount": 0,
      "UntreatedRiskCount": 4
    }
  ],
  "UntreatedRiskCount": 101
},
"RequestId": "d4f449f9-921c-43ad-a97a-edb27a2d685a",
"TotalCount": 1
}
```

示例2 查询仅超时未体检

输入示例

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

<公共请求参数>

```
{
  "Filters": [
    {
      "Name": "OnlyOverdue",
      "Values": [
        "1"
      ],
      "OperatorType": 7
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "AccountStats": [
      {
        "Member": {
          "AppId": "1314503630",
          "MemberId": "mem-1300846651-100027980407",
          "Nickname": "测试部门",
          "Uin": "100027980407"
        },
        "ProductStats": [],
        "UntreatedRiskCount": 0
      },
      {
        "Member": {
          "AppId": "1300846651",
          "MemberId": "mem-1300846651-100011949846",
          "Nickname": "焦糖小蛋糕",
          "Uin": "100011949846"
        },
        "ProductStats": [],

```

```
"UntreatedRiskCount": 0
},
{
  "Member": {
    "AppId": "1327492904",
    "MemberId": "mem-1300846651-100037509558",
    "Nickname": "fengqqian",
    "Uin": "100037509558"
  },
  "ProductStats": [],
  "UntreatedRiskCount": 0
},
{
  "Member": {
    "AppId": "1315399711",
    "MemberId": "mem-1300846651-100028671395",
    "Nickname": "美式",
    "Uin": "100028671395"
  },
  "ProductStats": [],
  "UntreatedRiskCount": 0
},
{
  "Member": {
    "AppId": "1314933167",
    "MemberId": "mem-1300846651-100028354982",
    "Nickname": "少年时",
    "Uin": "100028354982"
  },
  "ProductStats": [],
  "UntreatedRiskCount": 0
}
],
"RequestId": "82758a29-81f8-466e-a8cd-ea60343d23fe",
"TotalCount": 5
}
}
```

示例3 查询账号+产品维度风险统计

输入示例

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

<公共请求参数>

```
{
  "Limit": 20,
  "Offset": 0
}
```

输出示例

```
{
  "Response": {
    "AccountStats": [
      {
        "Member": {
          "AppId": "1300448058",
          "Uin": "100000009710",
          "Nickname": "cfw-test",
          "MemberId": "member-001"
        },
        "UntreatedRiskCount": 20,
        "ProductStats": [
          {
            "Product": "enterprise_sg",
            "ProductName": "企业安全组",
            "PolicyCount": 200,
            "UntreatedRiskCount": 20,
            "TotalRiskCount": 30,
            "TreatedRiskCount": 5,
            "IgnoredRiskCount": 5,
            "RectifyRate": "0%",
            "LastCheckTime": "2025-10-22 10:00:00"
          },
          {
            "Product": "sg",
            "ProductName": "安全组",
            "PolicyCount": 100,
            "UntreatedRiskCount": 0,

```

```
"TotalRiskCount": 0,
"TreatedRiskCount": 0,
"IgnoredRiskCount": 0,
"RectifyRate": "无需整改",
>LastCheckTime": "2025-10-22 10:00:00"
}
],
{
  "Member": {
    "AppId": "1300846651",
    "Uin": "100000009711",
    "Nickname": "cfw-test2",
    "MemberId": "member-002"
  },
  "UntreatedRiskCount": 1,
  "ProductStats": [
    {
      "Product": "sg",
      "ProductName": "安全组",
      "PolicyCount": 20,
      "UntreatedRiskCount": 1,
      "TotalRiskCount": 1,
      "TreatedRiskCount": 0,
      "IgnoredRiskCount": 0,
      "RectifyRate": "0%",
      "LastCheckTime": "2025-10-22 10:00:00"
    }
  ]
},
"TotalCount": 2,
"RequestId": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
}
```

示例4 根据Uin过滤

输入示例

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

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

<公共请求参数>

```
{
  "Filters": [
    {
      "Name": "Uin",
      "Values": [
        "100011616646"
      ],
      "OperatorType": 9
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "AccountStats": [
      {
        "Member": {
          "AppId": "1300448058",
          "MemberId": "mem-1300846651-100011616646",
          "Nickname": "天空之蓝",
          "Uin": "100011616646"
        },
        "ProductStats": [
          {
            "IgnoredRiskCount": 55,
            "LastCheckTime": "2026-03-26 15:29:52",
            "PolicyCount": 1151,
            "Product": "sg",
            "ProductName": "安全组",
            "RectifyRate": "0%",
            "TotalRiskCount": 152,
            "TreatedRiskCount": 0,
            "UntreatedRiskCount": 97
          },
          {
            "IgnoredRiskCount": 0,
```

```
"LastCheckTime": "2026-03-09 19:42:49",
"PolicyCount": 34,
"Product": "enterprise_sg",
"ProductName": "企业安全组",
"RectifyRate": "0%",
"TotalRiskCount": 4,
"TreatedRiskCount": 0,
"UntreatedRiskCount": 4
}
],
"UntreatedRiskCount": 101
}
],
"RequestId": "de07ff51-a85e-4970-883a-0b73f788eb06",
"TotalCount": 1
}
}
```

示例5 根据账号昵称筛选

输入示例

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

<公共请求参数>

```
{
  "Filters": [
    {
      "Name": "Nickname",
      "Values": [
        "蓝"
      ],
      "OperatorType": 9
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "AccountStats": [
      {
        "Member": {
          "AppId": "1300448058",
          "MemberId": "mem-1300846651-100011616646",
          "Nickname": "天空之蓝",
          "Uin": "100011616646"
        },
        "ProductStats": [
          {
            "IgnoredRiskCount": 55,
            "LastCheckTime": "2026-03-26 15:29:52",
            "PolicyCount": 1151,
            "Product": "sg",
            "ProductName": "安全组",
            "RectifyRate": "0%",
            "TotalRiskCount": 152,
            "TreatedRiskCount": 0,
            "UntreatedRiskCount": 97
          },
          {
            "IgnoredRiskCount": 0,
            "LastCheckTime": "2026-03-09 19:42:49",
            "PolicyCount": 34,
            "Product": "enterprise_sg",
            "ProductName": "企业安全组",
            "RectifyRate": "0%",
            "TotalRiskCount": 4,
            "TreatedRiskCount": 0,
            "UntreatedRiskCount": 4
          }
        ],
        "UntreatedRiskCount": 101
      }
    ],
    "RequestId": "09c2396d-5a7f-4d0b-b54d-eea11f0729f1",
    "TotalCount": 1
  }
}
```

示例6 默认请求

输入示例

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

{}
```

输出示例

```
{
  "Response": {
    "AccountStats": [
      {
        "Member": {
          "AppId": "1300448058",
          "MemberId": "mem-1300846651-100011616646",
          "Nickname": "天空之蓝",
          "Uin": "100011616646"
        },
        "ProductStats": [
          {
            "IgnoredRiskCount": 55,
            "IsOverdue": true,
            "LastCheckTime": "2026-03-26 15:29:52",
            "PolicyCount": 1151,
            "Product": "sg",
            "ProductName": "安全组",
            "RectifyRate": "0%",
            "SubcategoryIds": [
              "risk_port_ssh_22",
              "inbound_accept_any",
              "exact_duplicate_rules",
              "merge_rules",
              "risk_port_rdp_3389",
              "risk_port_20_21",
              "overridden_rules"
            ],
            "RiskCount": 55,
            "RiskLevel": "中",
            "RiskType": "端口",
            "Subcategory": "ssh_22",
            "SubcategoryName": "ssh_22",
            "SubcategoryType": "端口",
            "Uin": "100011616646"
          }
        ]
      }
    ]
  }
}
```

```
"TotalRiskCount": 152,
"TreatedRiskCount": 0,
"UntreatedRiskCount": 97
},
{
  "IgnoredRiskCount": 0,
  "IsOverdue": true,
  "LastCheckTime": "2026-03-27 11:47:27",
  "PolicyCount": 46,
  "Product": "enterprise_sg",
  "ProductName": "企业安全组",
  "RectifyRate": "无需整改",
  "SubcategoryIds": [],
  "TotalRiskCount": 24,
  "TreatedRiskCount": 24,
  "UntreatedRiskCount": 0
},
],
"RectifyRate": "14%",
"UntreatedRiskCount": 97
},
{
  "Member": {
    "AppId": "1300846651",
    "MemberId": "mem-1300846651-100011949846",
    "Nickname": "焦糖小蛋糕",
    "Uin": "100011949846"
  },
  "ProductStats": [],
  "RectifyRate": "无需整改",
  "UntreatedRiskCount": 0
},
{
  "Member": {
    "AppId": "1327492904",
    "MemberId": "mem-1300846651-100037509558",
    "Nickname": "fengqqian",
    "Uin": "100037509558"
  },
  "ProductStats": [],
  "RectifyRate": "无需整改",
  "UntreatedRiskCount": 0
},
},
```

```
{
  "Member": {
    "AppId": "1315399711",
    "MemberId": "mem-1300846651-100028671395",
    "Nickname": "美式",
    "Uin": "100028671395"
  },
  "ProductStats": [],
  "RectifyRate": "无需整改",
  "UntreatedRiskCount": 0
},
{
  "Member": {
    "AppId": "1314933167",
    "MemberId": "mem-1300846651-100028354982",
    "Nickname": "少年时",
    "Uin": "100028354982"
  },
  "ProductStats": [],
  "RectifyRate": "无需整改",
  "UntreatedRiskCount": 0
},
{
  "Member": {
    "AppId": "1314503630",
    "MemberId": "mem-1300846651-100027980407",
    "Nickname": "测试部门",
    "Uin": "100027980407"
  },
  "ProductStats": [],
  "RectifyRate": "无需整改",
  "UntreatedRiskCount": 0
}
],
"OverdueAccountCount": 6,
"OverdueProductCount": 2,
"RequestId": "6f57db6e-44a6-4d70-9fae-4b9eef649b3b",
"TotalCount": 6
}
```

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](#)。

查询风险分类统计数据

最近更新时间：2026-07-06 15:58:25

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询策略体检风险分类统计数据,包含各类风险的规则数量、处置状态、整改率等信息

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：DescribeRiskCategoryStats。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Limit	否	Integer	分页大小,默认20 示例值：20
Offset	否	Integer	分页偏移量,默认0 示例值：0
Product	否	String	产品类型 示例值：EnterpriseSG
Filters.N	否	Array of CommonFilter	筛选器
By	否	String	排序字段："RuleCount", "TreatedCount", "IgnoredCount", "UntreatedCount", "DisposalRate" 示例值：UntreatedCount
Order	否	String	顺序 示例值：asc
MemberId	否	String	成员Id 示例值：mem-1300846651-100011616646

3. 输出参数

参数名称	类型	描述
Total	Integer	风险分类总数 示例值：8
Data	Array of RiskCategoryItem	风险分类统计列表
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 查询策略体检风险分类统计数据

输入示例

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

{
  "Product": "enterprise_sg",
  "MemberId": "mem-1300846651-100011616646"
}
```

输出示例

```
{
  "Response": {
    "Data": [
      {
        "CategoryId": "deviate_baseline",
        "CategoryName": "安全基线偏离规则",
        "Description": "检测到入向全放行规则，范围过大",
        "DisposalRate": 0,
        "HasRisk": 1,
        "IgnoredCount": 0,
        "RemediationStatus": "Incomplete",

```

```
"RiskLevel": 2,
"RiskLevelName": "高危",
"RuleCount": 1,
"SubcategoryId": "inbound_accept_any",
"SubcategoryName": "入向全放行规则",
"Suggestion": "检测到入向全放行规则，范围过大。建议根据业务实际需求调整规则，缩小覆盖范围以提升规则精度。",
"TreatedCount": 0,
"UntreatedCount": 1
},
{
"CategoryId": "invalid_rule",
"CategoryName": "无效规则",
>Description": "规则关联的资产实例、地址模板、资源标签等模板已被部分或全部删除",
"DisposalRate": 0,
"HasRisk": 1,
"IgnoredCount": 0,
"RemediationStatus": "Incomplete",
"RiskLevel": 2,
"RiskLevelName": "高危",
"RuleCount": 1,
"SubcategoryId": "loss_of_effect_rules",
"SubcategoryName": "失效规则",
"Suggestion": "检测到规则关联的资产实例、地址模板、或资源标签等模板已被部分删除。建议直接删除该无效规则。",
"TreatedCount": 0,
"UntreatedCount": 1
},
{
"CategoryId": "invalid_rule",
"CategoryName": "无效规则",
>Description": "规则已被优先级更高的规则覆盖",
"DisposalRate": 0,
"HasRisk": 1,
"IgnoredCount": 0,
"RemediationStatus": "Incomplete",
"RiskLevel": 0,
"RiskLevelName": "低危",
"RuleCount": 2,
"SubcategoryId": "overridden_rules",
"SubcategoryName": "被高优覆盖规则",
"Suggestion": "检测到规则已被优先级更高的规则覆盖。建议删除被覆盖的无效规则。",
"TreatedCount": 0,
```

```
"UntreatedCount": 2
}
],
"RequestId": "11a97c77-9c0e-468f-b6a2-02b38c121322",
"Total": 2
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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。

查询策略风险列表

最近更新时间：2026-07-06 15:58:24

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询用户所有规则的策略问题

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：DescribeRiskList。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Limit	是	Integer	条数限制 示例值：100
Offset	是	Integer	偏移量 示例值：0
Product	否	String	产品类型 示例值：EnterpriseSG
Filters.N	否	Array of CommonFilter	筛选条件
Order	否	String	排序方式 示例值：desc
By	否	String	排序字段 示例值：FoundTime
MemberId	否	String	成员Id 示例值：mem-1300846651-100011616646

3. 输出参数

参数名称	类型	描述
PolicyRiskLst	Array of PolicyRisk	策略问题列表
Total	Integer	策略问题数 示例值：0
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 查询用户所有规则的策略问题

输入示例

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

{
  "Limit": 10,
  "Offset": 0,
  "Product": "enterprise_sg",
  "MemberId": "mem-1300846651-100011616646"
}
```

输出示例

```
{
  "Response": {
    "PolicyRiskLst": [
      {
        "Direction": "Egress",
        "DisposalTime": "",
        "FoundTime": "2026-03-04 17:36:12",
        "Id": "06c15d97ec325e8e5fe6f10883478360",
        "Product": "enterprise_sg",

```

```
"RiskCategory": "deviate_baseline",
"RiskFeature": "inbound_accept_any",
"RiskLevel": 2,
"RiskReason": "108275允许进站任意访问，范围过大",
"RiskSubCategory": "inbound_accept_any",
"RuleCount": 1,
"RuleType": "ACCEPT",
"SgId": [
  ""
],
"SgRuleId": [
  "108275"
],
>Status": 0,
"Suggestion": "检测到入向全放行规则，范围过大。建议根据业务实际需求调整规则，缩小覆盖范围以提升规则精度。",
},
{
  "Direction": "Egress",
  "DisposalTime": "",
  "FoundTime": "2026-03-02 22:22:33",
  "Id": "a10b189d0974f4f1b4f04fc98187fc4d",
  "Product": "enterprise_sg",
  "RiskCategory": "invalid_rule",
  "RiskFeature": "loss_of_effect_rules",
  "RiskLevel": 2,
  "RiskReason": "",
  "RiskSubCategory": "loss_of_effect_rules",
  "RuleCount": 1,
  "RuleType": "ACCEPT",
  "SgId": [
    ""
  ],
  "SgRuleId": [
    "164108"
  ],
  "Status": 0,
  "Suggestion": "检测到规则关联的资产实例、地址模板、或资源标签模板已被部分删除。建议直接删除该无效规则。",
},
{
  "Direction": "Egress",
  "DisposalTime": "",
  "FoundTime": "2026-03-04 17:58:46",
```

```
"Id": "c987cf88c6c0de2702f06530a0070564",
"Product": "enterprise_sg",
"RiskCategory": "invalid_rule",
"RiskFeature": "overridden_rules_by:108275",
"RiskLevel": 0,
"RiskReason": "规则（允许 ::/0 -> 2400:ee00:101c:5701:0:9d35:c8f9:d41f:ANY (ANY) ）覆盖了1条低优先级规则",
"RiskSubCategory": "overridden_rules",
"RuleCount": 2,
"RuleType": "ACCEPT",
"SgId": [
  ""
],
"SgRuleId": [
  "108275",
  "108276"
],
"Status": 0,
"Suggestion": "检测到规则已被优先级更高的规则覆盖。建议删除被覆盖的无效规则。"
}
],
"RequestId": "ba338963-595a-4d8b-b630-2b646cb97be1",
"Total": 3
}
}
```

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。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

实时分析风险详情

最近更新时间：2026-07-06 15:58:25

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

获取实时分析风险详情

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：DescribeRiskAnalysisDetails。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Id	否	String	风险ID 示例值：c1w3
SearchType	否	String	查询类型，analyze实时数据分析，task定时分析结果 示例值：analyze
MemberId	否	String	成员Id 示例值：mem-tencent-e4f32cfd87465a17

3. 输出参数

参数名称	类型	描述
EnterpriseSecurityGroupRule	Array of AnalysisSgRuleInfoResp	风险企业安全组规则列表
SecurityGroupPolicy	Array of SecurityGroupRiskPolicy	风险安全组规则列表

参数名称	类型	描述
Status	Integer	实时分析状态，1分析执行中请轮询，0分析已结束 示例值：1
EnterpriseSecurityGroupRuleIPV6	Array of AnalysisSgRuleInfoResp	风险企业安全组IPV6规则列表
RequestId	String	唯一请求 ID，由服务端生成，每次请求都会返回（若请求因其他原因未能抵达服务端，则该次请求不会获得 RequestId）。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取实时分析风险详情

输入示例

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

{
  "Id": "398d5185cb7c4b15e44640a38f78d681",
  "SearchType": "analyze"
}
```

输出示例

```
{
  "Response": {
    "EnterpriseSecurityGroupRuleIPV6": [],
    "RequestId": "0831ba3a-9c6d-4b1b-98e2-b92c2499e934"
  }
}
```

示例2 查看指定成员的

输入示例

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

<公共请求参数>

```
{
  "Id": "77c4bea5ce631684079e8d6420d535ed",
  "SearchType": "analyze",
  "MemberId": "mem-tencent-e4f32cfd87465a17"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "14e70aa5-d247-47a2-ba74-d67f5d05956b",
    "SecurityGroupPolicy": [
      {
        "Action": "ACCEPT",
        "AddressTemplate": {
          "AddressGroupId": "",
          "AddressId": ""
        },
        "CidrBlock": "0.0.0.0/0",
        "Direction": "Ingress",
        "Ipv6CidrBlock": "",
        "ModifyTime": "2026-03-27 17:23:58",
        "PolicyDescription": "cdasccsd",
        "PolicyIndex": 0,
        "Port": "ALL",
        "Protocol": "ALL",
        "Region": "ap-shanghai-fsi",
        "SecurityGroupId": "",
        "ServiceTemplate": {
          "ServiceGroupId": "",
          "ServiceId": ""
        },
        "Version": "3"
      },
    ],
  },
}
```

```
"Action": "ACCEPT",
"AddressTemplate": {
"AddressGroupId": "",
"AddressId": ""
},
"CidrBlock": "",
"Direction": "Ingress",
"Ipv6CidrBlock": "::/0",
"ModifyTime": "2026-03-16 17:54:26",
"PolicyDescription": "",
"PolicyIndex": 1,
"Port": "ALL",
"Protocol": "ALL",
"Region": "ap-shanghai-fsi",
"SecurityGroupId": "",
"ServiceTemplate": {
"ServiceGroupId": "",
"ServiceId": ""
},
"Version": "3"
}
]
}
}
```

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](#)。

取消忽略策略风险

最近更新时间：2026-07-06 15:58:26

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

取消忽略策略风险

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：CancellgnorePolicyRisk。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
RiskId	是	String	风险ID 示例值：1
MemberId	否	String	成员Id 示例值：mem-tencent-e4f32cfd87465a17

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 取消忽略风险

输入示例

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

{
  "RiskId": "fb7a6f3ccbcf1de31cb8ac9abb0e5b23"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "4d48e988-fdd5-472b-97ae-c41e63928257"
  }
}
```

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。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

创建风险分析任务

最近更新时间：2026-07-06 15:58:26

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

创建策略风险分析任务

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateAnalyzePolicyTask。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Products.N	是	Array of String	产品类型 示例值：["EnterpriseSG","SG"]
MemberIdSet.N	否	Array of String	成员Id 列表 示例值：["mem-1300846651-100011616646"]

3. 输出参数

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

4. 示例

示例1 创建策略分析任务

输入示例

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

{
  "Products": [
    "sg"
  ],
  "MemberIdSet": [
    "mem-1300846651-100011616646"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "552fd3c4-9fb2-4bdb-8fac-dac5fccc1b98"
  }
}
```

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。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

规则组相关接口

创建NAT边界规则组

最近更新时间：2026-07-06 15:58:22

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

创建NAT ACL规则组（NAT边界防火墙规则组管理）

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：CreateNatAclRuleGroup。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupName	是	String	规则组名称 示例值："NAT规则组-测试"
Product	是	String	产品类型，固定为 cfw_nat_acl 示例值：cfw_nat_acl
Rules.N	是	Array of NatAclRule	规则列表

3. 输出参数

参数名称	类型	描述
GroupId	String	创建的规则组ID 示例值：fwmrg_abc123xyz
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NAT 规则组

输入示例

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

<公共请求参数>

```
{
  "GroupName": "one",
  "Product": "cfw_nat_acl",
  "Rules": [
    {
      "SourceContent": "0.0.0.0/0",
      "SourceType": "net",
      "TargetContent": "0.0.0.0/0",
      "TargetType": "net",
      "Protocol": "TCP",
      "RuleAction": "accept",
      "OrderIndex": 1,
      "Scope": "ap-shanghai",
      "Direction": 1,
      "Port": "-1/-1",
      "Description": "zero",
      "ParamTemplateId": ""
    },
    {
      "SourceContent": "0.0.0.0/0",
      "SourceType": "net",
      "TargetContent": "0.0.0.0/0",
      "TargetType": "net",
      "Protocol": "UDP",
      "RuleAction": "accept",
      "OrderIndex": 2,
      "Scope": "cfwnat-e7ad7c24",
      "Direction": 1,
      "Port": "-1/-1",
      "Description": "222",
      "ParamTemplateId": ""
    }
  ]
}
```

```
"BelongMemberId": "1300448058"
},
{
  "SourceContent": "0.0.0.0/0",
  "SourceType": "net",
  "TargetContent": "0.0.0.0/0",
  "TargetType": "net",
  "Protocol": "ANY",
  "RuleAction": "accept",
  "OrderIndex": 3,
  "Scope": "ALL",
  "Direction": 1,
  "Port": "-1/-1",
  "Description": "333",
  "ParamTemplateId": ""
}
]
}
```

输出示例

```
{
  "Response": {
    "GroupId": "fwmrg_wrzekq94u7",
    "RequestId": "62cbbbe0-bac7-4e18-a85c-94a7784dfd04"
  }
}
```

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。

创建互联网边界规则组

最近更新时间：2026-07-06 15:58:22

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

创建互联网边界ACL规则组，支持同时创建多条规则。Product 必须为 cfw_edge_acl。规则支持 IP、域名、参数模板、实例、标签等多种源/目标类型。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateEdgeAclRuleGroup。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupName	否	String	规则组名称，长度1-50字符 示例值：Edge边界规则组-测试
Product	否	String	产品类型，固定为 cfw_edge_acl 示例值：cfw_edge_acl
Rules.N	否	Array of EdgeAclRuleInfo	规则列表

3. 输出参数

参数名称	类型	描述
GroupId	String	创建的规则组ID 示例值：fwmrg_abc123xyz
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 创建互联网边界规则组

输入示例

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

{
  "GroupName": "edge",
  "Product": "cfw_edge_acl",
  "Rules": [
    {
      "OrderIndex": 1,
      "Direction": 1,
      "SourceContent": "0.0.0.0/0",
      "SourceType": "net",
      "TargetContent": "0.0.0.0/0",
      "TargetType": "net",
      "Port": "-1/-1",
      "Protocol": "TCP",
      "RuleAction": "accept",
      "Description": "1212",
      "Scope": "all"
    },
    {
      "OrderIndex": 2,
      "Direction": 1,
      "SourceContent": "bj11",
      "SourceType": "location",
      "TargetContent": "cfwnat-e366d270",
      "TargetType": "instance",
      "Port": "-1/-1",
      "Protocol": "TCP",
      "RuleAction": "accept",
      "Description": "3232",
      "Scope": "all",
      "BelongMemberId": "1300448058"
    }
  ]
}
```

```
]
}
```

输出示例

```
{
  "Response": {
    "GroupId": "fwmrj_4ae6vv117c",
    "RequestId": "fcb93770-4c26-4b35-9ded-8daec5b99a19"
  }
}
```

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。
ResourceInsufficient	资源不足。
UnauthorizedOperation	未授权操作。
UnsupportedOperation	操作不支持。

创建VPC边界规则组

最近更新时间：2026-07-06 15:58:21

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

创建VPC ACL规则组（VPC间防火墙规则组管理）

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：CreateVpcAclRuleGroup。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupName	是	String	规则组名称 示例值："VPC规则组-测试"
Product	是	String	产品类型，固定为 cfw_vpc_acl 示例值：cfw_vpc_acl
Rules.N	是	Array of VpcAclRule	规则列表

3. 输出参数

参数名称	类型	描述
GroupId	String	创建的规则组ID 示例值：fwmrg_abc123xyz
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 vpc 规则组

输入示例

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

<公共请求参数>

```
{
  "GroupName": "222",
  "Product": "cfw_vpc_acl",
  "Rules": [
    {
      "SourceContent": "0.0.0.0/0",
      "SourceType": "net",
      "DestContent": "0.0.0.0/0",
      "DestType": "net",
      "Protocol": "ANY",
      "RuleAction": "accept",
      "OrderIndex": 1,
      "EdgeId": "all",
      "FwGroupId": "ccn-c0qmm031",
      "Port": "-1/-1",
      "Description": "one",
      "ParamTemplateId": "",
      "BelongMemberId": "1300448058"
    },
    {
      "SourceContent": "0.0.0.0/0",
      "SourceType": "net",
      "DestContent": "0.0.0.0/0",
      "DestType": "net",
      "Protocol": "ANY",
      "RuleAction": "accept",
      "OrderIndex": 2,
      "EdgeId": "all",
      "FwGroupId": "cfwg-e37fa79c",
      "Port": "-1/-1",
      "Description": "2222",
      "ParamTemplateId": "",
      "BelongMemberId": "1300448058"
    }
  ]
}
```

```
}  
]  
}
```

输出示例

```
{  
  "Response": {  
    "GroupId": "fwmrg_0i4eie7e50",  
    "RequestId": "0573ab58-ad2c-4bf4-9868-43ba0e066db0"  
  }  
}
```

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。

创建企业安全组规则组

最近更新时间：2026-07-06 15:58:22

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

创建规则组（规则组管理）

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： CreateSecurityGroupRuleGroup。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupName	否	String	规则组ID 示例值："RG-13KAE"
Product	否	String	产品类型 示例值：0
Rules.N	否	Array of SecurityGroupRule	规则列表

3. 输出参数

参数名称	类型	描述
GroupId	String	创建的规则组ID 示例值：fwmrg_abc123xyz
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 创建规则组

输入示例

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

<公共请求参数>

```
{
  "GroupName": "归属测试组1",
  "Product": "enterprise_sg",
  "Rules": [
    {
      "IpVersion": "ipv4",
      "SourceContent": "subnet-imemp5hz",
      "SourceType": "instance",
      "DestContent": "0.0.0.0/0",
      "DestType": "net",
      "Protocol": "ANY",
      "Port": "-1/-1",
      "ServiceTemplateId": "",
      "RuleAction": "accept",
      "Description": "1221",
      "OrderIndex": 1,
      "Scope": "SG",
      "ProtocolPortType": 0,
      "BelongMemberId": "1302477603"
    },
    {
      "IpVersion": "ipv4",
      "SourceContent": "192.168.0.0/16",
      "SourceType": "net",
      "DestContent": "0.0.0.0/0",
      "DestType": "net",
      "Protocol": "ANY",
      "Port": "-1/-1",
      "ServiceTemplateId": "",
      "RuleAction": "accept",
      "Description": "33333",
    }
  ]
}
```

```
"OrderIndex": 2,
"Scope": "SG",
"ProtocolPortType": 0
}
]
}
```

输出示例

```
{
  "Response": {
    "GroupId": "fwmg-jvw5ulal",
    "RequestId": "1af65155-debe-49ca-b9d3-cac479b51cc7"
  }
}
```

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。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
UnknownParameter	未知参数错误。

删除规则组

最近更新时间：2026-07-06 15:58:21

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

删除规则组

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：DeleteRuleGroup。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupIds.N	否	Array of String	规则组Id列表 示例值：["RG-234JSES","RG-234JSES"]

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 删除规则组

输入示例

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

<公共请求参数>

```
{
  "GroupIds": [
    "fwmrg_7lonr9wtf2",
    "fwmrg-ibmst8kv"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "46ff3321-e595-4bd8-a5a1-5d09f7c7a004"
  }
}
```

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。
ResourceInUse	资源被占用。

修改规则组信息

最近更新时间：2026-07-06 15:58:21

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

修改规则组信息（规则组管理）

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：ModifyRuleGroup。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
GroupId	否	String	规则组名称 示例值：RG-8UJS2
GroupName	否	String	规则组名称 示例值：重保规则组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 修改规则组信息

输入示例

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

```
{
  "GroupId": "fwmrj_3zo766tr76",
  "GroupName": "gltest3"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "776b199e-2954-4d63-b36d-3a12ca06e59d"
  }
}
```

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。

规则管理相关接口

创建下发规则

最近更新时间：2026-07-06 15:58:24

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

创建策略

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：CreateStrategy。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Product	是	String	产品类型 示例值：1
ReceiveAccount.N	否	Array of String	下发账号 示例值：["10000000698","10000000697"]
PreStrategy.N	否	Array of StrategyReq	前区规则组
PostStrategy.N	否	Array of StrategyReq	后区规则组
ReceiveGroup.N	否	Array of String	下发账号组 示例值：["10000000698","10000000697"]

3. 输出参数

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

4. 示例

示例1 创建下发策略

输入示例

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

{
  "Product": "enterprise_sg",
  "ReceiveAccount": [
    "100011616646"
  ],
  "PreStrategy": [
    {
      "GroupId": "fwmrg-sygzqi8x",
      "Sequence": 1
    },
    {
      "GroupId": "fwmrg-5gxybcnr",
      "Sequence": 2
    }
  ],
  "PostStrategy": [
    {
      "GroupId": "fwmrg-6wq0b6j0",
      "Sequence": 1
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "0517848f-e4bf-419e-a26a-724289d0b9ea"
  }
}
```

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。

删除下发规则

最近更新时间：2026-07-06 15:58:24

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

删除策略

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：DeleteStrategy。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
StrategyIds.N	否	Array of String	策略Id列表 示例值：["fwmsid_tqupwhxyt1","fwmsid_296bree3jd"]

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 删除下发策略

输入示例

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

<公共请求参数>

```
{
  "StrategyIds": [
    "fwms-44n76l18",
    "fwms-zz23s84d",
    "fwms-nwwa678r"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "683c469a-bfa7-42f1-8169-ee3da5d13f31"
  }
}
```

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。

查询规则列表

最近更新时间：2026-07-06 15:58:24

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询策略列表

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：DescribeStrategies。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Product	否	String	产品类型 示例值：secgroup
ExecArea	否	String	执行区域：pre是前区，post 是后区 示例值：pre
Filters.N	否	Array of CommonFilter	筛选条件
Limit	否	Integer	条数限制 示例值：100
Offset	否	Integer	偏移量 示例值：0

3. 输出参数

参数名称	类型	描述
TotalCount	Integer	总数

参数名称	类型	描述
		示例值：102
Strategies	Array of StrategyResp	策略列表
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 查询策略列表

输入示例

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

{
  "Product": "enterprise_sg",
  "ExecArea": "pre",
  "Filters": [
    {
      "Name": "GroupName",
      "OperatorType": 9,
      "Values": [
        "类型安全组"
      ]
    }
  ],
  "Limit": 10,
  "Offset": 0
}
```

输出示例

```
{
  "Response": {
    "RequestId": "6139ea4e-d67e-4bc6-a122-d71a6a339dac",
```

```
"Strategies": [
  {
    "CreateBy": "100011616646",
    "CreateName": "天空之蓝",
    "CreateTime": "2026-02-27 19:06:59",
    "ErrMsg": "",
    "ErrorType": "",
    "ExecArea": "pre",
    "GroupId": "fwmrg-6wq0b6j0",
    "GroupName": "不同类型安全组下发",
    "ReceiveAccount": [
      {
        "Nickname": "天空之蓝",
        "Uin": "100011616646"
      }
    ],
    "RuleCount": 7,
    "RuleStatus": 1,
    "Sequence": 2,
    "StrategyId": "fwms-ms91jjvy",
    "UpdateBy": "100011616646",
    "UpdateName": "天空之蓝",
    "UpdateTime": "2026-02-27 19:52:50"
  },
  {
    "CreateBy": "100011616646",
    "CreateName": "天空之蓝",
    "CreateTime": "2026-02-27 19:22:44",
    "ErrMsg": "",
    "ErrorType": "",
    "ExecArea": "pre",
    "GroupId": "fwmrg-6wq0b6j0",
    "GroupName": "不同类型安全组下发",
    "ReceiveAccount": [
      {
        "Nickname": "天空之蓝",
        "Uin": "100011616646"
      }
    ],
    "RuleCount": 7,
    "RuleStatus": 1,
    "Sequence": 3,
    "StrategyId": "fwms-vg9ji1r5",
```

```
"UpdateBy": "100011616646",
"UpdateName": "天空之蓝",
"UpdateTime": "2026-02-27 19:52:50"
}
],
"TotalCount": 2
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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。
InternalServerError.Unknown	InternalServerError.Unknown

错误码	描述
InvalidParameter	参数错误。
MissingParameter	缺少参数错误。
MissingParameter.CommonParam	MissingParameter.CommonParam
ResourceNotFound	资源不存在。
ResourceNotFound.NotFound	ResourceNotFound.NotFound
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。

查询下发规则信息

最近更新时间：2026-07-06 15:58:23

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

查询策略详情

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeStrategy。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
StrategyId	否	String	策略Id 示例值：SID-133

3. 输出参数

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

4. 示例

示例1 查询策略详情

输入示例

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

<公共请求参数>

```
{
  "StrategyId": "fwms-kmbf8avt"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "209ce0d3-c47f-49b9-936c-c35aa2966910",
    "Strategy": {
      "CreateBy": "100011616646",
      "CreateName": "天空之蓝",
      "CreateTime": "2026-02-27 19:22:44",
      "ErrMsg": "",
      "ErrorType": "",
      "ExecArea": "pre",
      "GroupId": "fwmrg-5gxybcnr",
      "GroupName": "gl测试构造风险",
      "ReceiveAccount": [
        {
          "Nickname": "天空之蓝",
          "Uin": "100011616646"
        }
      ],
      "RuleCount": 11,
      "RuleStatus": 1,
      "Sequence": 1,
      "StrategyId": "fwms-kmbf8avt",
      "UpdateBy": "100011616646",
      "UpdateName": "天空之蓝",
      "UpdateTime": "2026-02-27 19:52:50"
    }
  }
}
```

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。
ResourceNotFound	资源不存在。
ResourceNotFound.NotFound	ResourceNotFound.NotFound

下发规则

最近更新时间：2026-07-06 15:58:23

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

下发策略

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：DispatchStrategy。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Status	否	Integer	1:下发, 2:中止 示例值：1
Product	否	String	产品 示例值：secgroup

3. 输出参数

参数名称	类型	描述
Status	Integer	返回状态 示例值：0
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 下发策略

输入示例

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

{
  "Status": 1,
  "Product": "enterprise_sg"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "ef9bf807-fa58-425e-b660-6a35462f281f"
  }
}
```

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。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
OperationDenied	操作被拒绝。
ResourceNotFound	资源不存在。

编辑下发规则

最近更新时间：2026-07-06 15:58:23

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

修改策略信息

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：ModifyStrategy。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
StrategyId	否	String	策略Id 示例值：SID-211
ReceiveAccount.N	否	Array of String	下发规则接收账号 示例值：["1000000067","100000045"]
Sequence	否	Integer	优先级 示例值：1
GroupId	否	String	规则组Id 示例值：RG-2334J
ReceiveGroup.N	否	Array of String	下发规则接收账号组 示例值：["1000000067","100000045"]

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 修改策略

输入示例

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

{
  "StrategyId": "fwms-wsnee2iu",
  "ReceiveAccount": [
    "100011616646"
  ],
  "Sequence": 1,
  "GroupId": "fwmrg-ubr1sci4",
  "ReceiveGroup": [
    "acg-ssbkqes9"
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "650cf5bf-8f86-4a21-bfc0-8fc587d415be"
  }
}
```

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。
ResourceNotFound	资源不存在。
ResourceNotFound.NotFound	ResourceNotFound.NotFound

下发规则快速排序

最近更新时间：2026-07-06 15:58:22

1. 接口描述

接口请求域名：fwm.tencentcloudapi.com。

快速排序修改策略优先级

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：ModifyStrategySequence。
Version	是	String	公共参数 ，本接口取值：2025-06-11。
Region	否	String	公共参数 ，此参数为可选参数。
Sequences.N	否	Array of SequenceIndex	优先级列表
ExecArea	否	String	执行区域 示例值：post
Product	否	String	产品类型 示例值：secgroup

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 策略快速排序

输入示例

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

{
  "Sequences": [
    {
      "OrderIndex": 3,
      "NewOrderIndex": 1
    },
    {
      "OrderIndex": 1,
      "NewOrderIndex": 3
    }
  ],
  "ExecArea": "pre",
  "Product": "enterprise_sg"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "c05626c7-1fab-4394-842a-393cfa56fa2c"
  }
}
```

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。

数据结构

最近更新时间：2026-07-06 15:58:35

Account

账号基本信息，主要有 Uin 和 AppId

被如下接口引用：DescribeStrategyAccounts。

名称	类型	描述
AppId	String	租户appid 示例值：12500000
Uin	String	租户uin 示例值：10000001
RemainQuota	Integer	剩余可用额度 示例值：1000
Nickname	String	租户名称 示例值：天空之蓝
DispatchRuleNum	Integer	下发规则数 示例值：100
OriginRuleNum	Integer	产品已有规则数 示例值：100
TotalQuota	Integer	总额度 示例值：10000
MemberId	String	成员Id 示例值：mem-1300846651-100011616646

AccountGroupInfo

账户组信息

被如下接口引用：DescribeOrganMembers。

名称	类型	必选	描述
GroupId	String	否	账户组ID 示例值：acg-23ras
GroupName	String	否	账户组名称 示例值：生产环境组

AccountGroupQuotaDetail

账号组配额详情

被如下接口引用：DescribeStrategyAccounts。

名称	类型	描述
GroupId	String	账号组Id 示例值：acg-2fafs
GroupName	String	账号组名称 示例值：生产组
MemberCount	Integer	账号组成员数 示例值：2
RemainQuota	Integer	取组内 RemainQuota 最小成员的值 示例值：10
TotalQuota	Integer	同上成员的 TotalQuota 示例值：2
DispatchRuleNum	Integer	同上成员的 DispatchRuleNum 示例值：6
OriginRuleNum	Integer	同上成员的 OriginRuleNum 示例值：1
BottleneckUin	String	配额最少的成员 Uin 示例值：1342343243
Members	Array of Account	成员列表

AccountProductDetailStats

单个产品的风险统计详情

被如下接口引用：DescribePolicyRiskAccountProductStats。

名称	类型	描述
Product	String	产品类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：enterprise_sg
ProductName	String	产品名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：企业安全组
PolicyCount	Integer	体检策略数 注意：此字段可能返回 null，表示取不到有效值。 示例值：200

名称	类型	描述
UntreatedRiskCount	Integer	待整改风险数 注意：此字段可能返回 null，表示取不到有效值。 示例值：20
TotalRiskCount	Integer	总风险数 注意：此字段可能返回 null，表示取不到有效值。 示例值：15
TreatedRiskCount	Integer	已处置数 注意：此字段可能返回 null，表示取不到有效值。 示例值：3
IgnoredRiskCount	Integer	已忽略数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
RectifyRate	String	整改率，如 50%，无需整改时为 无需整改 注意：此字段可能返回 null，表示取不到有效值。 示例值：0%
LastCheckTime	String	最近一次体检时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2025-10-22 10:00:00
SubcategoryIds	Array of String	子类 ID 列表 示例值：["sub_001", "sub_003", "sub_007"]
IsOverdue	Boolean	是否超时未体检 示例值：true

AccountStatsGroup

按账号分组的风险统计数据

被如下接口引用：DescribePolicyRiskAccountProductStats。

名称	类型	描述
Member	MemberInfo	成员账号信息 注意：此字段可能返回 null，表示取不到有效值。
UntreatedRiskCount	Integer	该账号下所有产品待整改风险数汇总 注意：此字段可能返回 null，表示取不到有效值。 示例值：20
ProductStats	Array of AccountProductDetailStats	该账号下各产品维度的风险统计列表 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
RectifyRate	String	整改率 示例值：16.0%

AddressTemplateSpecification

安全组地址模版

被如下接口引用：DescribeRiskAnalysisDetails。

名称	类型	必选	描述
AddressId	String	否	IP地址ID，例如：ipm-2uw6ujo6。 示例值：ipm-2uw6ujo6
AddressGroupId	String	否	IP地址组ID，例如：ipmg-2uw6ujo6。 示例值：ipm-2uw6ujo6

AnalysisSgRuleInfoResp

企业安全组规则列表信息

被如下接口引用：DescribeRiskAnalysisDetails。

名称	类型	描述
Id	Integer	规则id 等同RuleUuid 示例值：1
RuleId	String	规则Id 示例值：id-xxxx
OrderIndex	Integer	排序 示例值：1
CfwOrderIndex	Integer	云防排序 示例值：1
SourceId	String	源规则内容 示例值：1.1.1.1
SourceType	Integer	源规则类型 取值范围 0/1/2/3/4/5/6/7/8/9/100 0表示ip(net), 1表示VPC实例(instance) 2表示子网实例(instance) 3表示CVM实例(instance) 4表示CLB实例(instance) 5表示ENI实例(instance) 6表示数据库实例(instance)

名称	类型	描述
		7表示模板(template) 8表示标签(tag) 9表示地域(region) 100表示资产分组(resourcegroup) 示例值: 0
TargetId	String	目的规则内容 示例值: 1.1.1.1
TargetType	Integer	目的规则类型 取值范围 0/1/2/3/4/5/6/7/8/9/100 0表示ip(net), 1表示VPC实例(instance) 2表示子网实例(instance) 3表示CVM实例(instance) 4表示CLB实例(instance) 5表示ENI实例(instance) 6表示数据库实例(instance) 7表示模板(template) 8表示标签(tag) 9表示地域(region) 100表示资产分组(resourcegroup) 示例值: 0
Protocol	String	协议名称 取值范围:TCP/ANY/ICMP/UDP ANY:表示所有 示例值: TCP
Port	String	端口 示例值: 3306
Strategy	Integer	规则策略 取值范围:1/2 1:阻断 2:放行 示例值: 1
Detail	String	描述 示例值: 规则描述
Region	String	地域 示例值: ap-guangzhou
ServiceTemplateId	String	服务模板id 示例值: tpm-3c140219
SoulInstanceName	String	源资产名称 示例值: tpm-3c140219

名称	类型	描述
SouPublicIp	String	源资产公网ip 示例值: 1.1.1.1
SouPrivateIp	String	源资产内网ip 示例值: 1.1.1.1
SouCidr	String	源资产网段信息 示例值: 10.0.0.0/32
SouParameterName	String	源模板名称 示例值: 模板名称
InstanceName	String	目的资产名称 示例值: ins-3c140219
PublicIp	String	目的资产公网ip 示例值: 1.1.1.1
PrivateIp	String	目的资产内网ip 示例值: 1.1.1.1
Cidr	String	目的资产网段信息 示例值: 10.0.0.0/32
ParameterName	String	目的模板名称 示例值: 模板名称
ProtocolPortName	String	端口模板名称 示例值: 模板名称
DnsParseCount	SgDnsParseCount	域名解析的IP统计
Scope	String	规则生效范围 示例值: SG
RulePartition	Integer	分区: 1防火墙管理最前分区 2是云防规则 3防火墙管理最后分区 示例值: 1
GroupId	String	规则组Id 示例值: fwmsg-xxxxxxxxxx
GroupName	String	规则组名称 示例值: 下发规则组
GroupRuleId	String	规则组内规则id 示例值: 539a6900-855b-484c-9c99-ac0c37028ddf
StrategyId	String	策略Id 示例值: fwmsid-xxxxxxxxxx

名称	类型	描述
IpVersion	String	ip类型 示例值: ipv4
BelongMember	MemberInfo	成员信息 注意: 此字段可能返回 null, 表示取不到有效值。

CommonFilter

通用筛选条件

被如下接口引用: DescribeEdgeAclRules, DescribeNatAclRules, DescribeOrganMembers, DescribePolicyRiskAccountProductStats, DescribeRiskCategoryStats, DescribeRiskList, DescribeSecurityGroupRules, DescribeStrategies, DescribeStrategyAccounts, DescribeVpcAclRules。

名称	类型	必选	描述
Name	String	否	筛选字段名。支持: SecurityGroupId、FwGroupId、Ip (IP地址模糊搜索)、InstanceName (实例名称模糊搜索)、VpcId (VPC ID精确搜索) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: SecurityGroupId
Values	Array of String	否	筛选值列表 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: []
OperatorType	Integer	否	操作类型。1=等于, 7=in, 9=模糊匹配 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 7

EdgeAclRuleInfo

互联网边界规则

被如下接口引用: CreateEdgeAclRule, CreateEdgeAclRuleGroup, ModifyEdgeAclRule。

名称	类型	必选	描述
RuleId	String	否	规则ID, 修改规则时必填 示例值: rule_001
OrderIndex	Integer	否	规则执行顺序, 数字越小优先级越高, 创建规则组时必须从1开始严格递增 示例值: 1
Direction	Integer	否	规则方向: 0-出站, 1-进站 示例值: 1
SourceContent	String	否	源地址内容, 根据 SourceType 不同有不同的格式: ip 时为 IP/CIDR, domain 时为域名, template 时为模板ID, instance 时为实例ID列表

名称	类型	必选	描述
			(逗号分隔), tag 时为标签键值对 (格式: key:value) 示例值: 192.168.1.0/24
SourceType	String	否	源地址类型: ip-IP地址, domain-域名, template-参数模板, instance-实例, tag-标签 示例值: ip
TargetContent	String	否	目标地址内容, 格式同 SourceContent 示例值: 10.0.0.0/8
TargetType	String	否	目标地址类型: ip-IP地址, domain-域名, template-参数模板, instance-实例, tag-标签 示例值: ip
Port	String	否	端口, 支持单端口、端口范围和逗号分隔的多端口, 如: 80、1-65535、80,443,8080 示例值: 80,443
Protocol	String	否	协议类型: TCP、UDP、ICMP、ANY 示例值: TCP
RuleAction	String	否	规则动作: accept-放行, drop-阻断, log-观察 示例值: accept
Description	String	否	规则描述, 长度0-256字符 示例值: 允许内网访问
Scope	String	否	生效范围: serial, 串行; side, 旁路; all, 全局 示例值: all
BelongMemberId	String	否	规则归属的成员账号ID (多账号场景下使用)。当 SourceType 或 TargetType 为 instance 或 tag 时, 此参数必填, 用于指定实例/标签所属的成员账号 示例值: 100023718505
ParamTemplateId	String	否	参数模板 示例值: pp-23432

EdgeAclRuleResp

互联网边界规则响应结构

被如下接口引用: DescribeEdgeAclRules。

名称	类型	描述
RuleId	String	规则ID 示例值: rule_001
Sequence	Integer	规则执行顺序 示例值: 1

名称	类型	描述
Direction	Integer	规则方向：0-出站，1-进站 示例值：1
SourceContent	String	源地址内容 示例值：192.168.1.0/24
SourceType	String	源地址类型 示例值：ip
SourceName	String	源地址名称（当类型为模板/实例/标签时返回对应名称） 注意：此字段可能返回 null，表示取不到有效值。 示例值：内网模板
TargetContent	String	目标地址内容 示例值：10.0.0.0/8
TargetType	String	目标地址类型 示例值：ip
TargetName	String	目标地址名称（当类型为模板/实例/标签时返回对应名称） 注意：此字段可能返回 null，表示取不到有效值。 示例值：生产环境服务器
Port	String	目标端口 示例值：80,443
Protocol	String	协议类型 示例值：TCP
RuleAction	String	规则动作：accept-放行，drop-阻断，log-观察 示例值：accept
Description	String	规则描述 示例值：允许内网访问
Scope	String	规则生效范围 示例值：0
CountryName	String	地域名称1（正则匹配时使用） 注意：此字段可能返回 null，表示取不到有效值。 示例值：中国
CityName	String	地域名称2（正则匹配时使用） 注意：此字段可能返回 null，表示取不到有效值。 示例值：北京
ParamTemplateId	String	参数模板ID（当类型为模板时返回） 注意：此字段可能返回 null，表示取不到有效值。 示例值：tpl_001
ParamTemplateName	String	参数模板名称（当类型为模板时返回） 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
		示例值：内网地址模板
Invalid	Integer	规则是否失效：0-有效，1-失效 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
BelongMember	MemberInfo	规则归属的成员账号 注意：此字段可能返回 null，表示取不到有效值。
CountryCode	Integer	国家Id 示例值：0
CityCode	Integer	城市Id 示例值：0
IsRegion	Integer	0为正常规则,1为地域规则 示例值：0
CloudCode	String	云厂商code 示例值：0
IsCloud	Integer	0为正常规则,1为云厂商规则 示例值：0
InstanceName	String	实例名称 示例值：内部接口实例
CountryKey	String	地区简称 示例值：zd
CityKey	String	省份、城市简称 示例值：hn43
CreateTime	String	规则创建时间 示例值：2025-07-24 12:20:21
UpdateTime	String	规则最近更新时间 示例值：2025-07-24 12:20:21
DnsParseCnt	Integer	域名数 示例值：0

MemberInfo

成员信息

被如下接口引用：DescribeEdgeAclRules, DescribeNatAclRules, DescribeOrganSummary, DescribePolicyRiskAccountProductStats, DescribeRiskAnalysisDetails, DescribeSecurityGroupRules, DescribeStrategies, DescribeStrategy, DescribeVpcAclRules。

名称	类型	必选	描述
AppId	String	否	成员AppId 示例值：16898822
Uin	String	否	成员Uin 示例值：12121212
Nickname	String	否	成员昵称 示例值：天空之蓝
MemberId	String	否	成员Id 示例值：1212-3232

NatAclRule

NAT边界规则

被如下接口引用：CreateNatAclRule, CreateNatAclRuleGroup, ModifyNatAclRule。

名称	类型	必选	描述
SourceContent	String	是	源地址内容 示例值：10.0.0.0/24
SourceType	String	是	源类型：ip/url/template/instance/tag 示例值：ip
TargetContent	String	是	目的地址内容 示例值：0.0.0.0/0
TargetType	String	是	目的类型：ip/url/template/instance/tag 示例值：ip
Protocol	String	是	协议：TCP/UDP/ICMP/ANY/HTTP/HTTPS/DNS/FTP等 示例值：TCP
RuleAction	String	是	动作：accept/drop/log 示例值：accept
OrderIndex	Integer	是	优先级（从1开始） 示例值：1
Scope	String	是	规则生效范围：ALL-全局生效，ap-xxx-地域生效，cfwnat-xxx-NAT 防火墙实例生效 示例值：ALL
Direction	Integer	否	规则方向：1-入站规则，0-出站规则 示例值：1
RuleId	String	否	规则ID（修改时必填） 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	必选	描述
			示例值：uuid-xxx
Port	String	否	端口（ICMP协议时空） 注意：此字段可能返回 null，表示取不到有效值。 示例值：80
Description	String	否	规则描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：允许HTTP流量
ParamTemplateId	String	否	端口模板ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：ppm-xxx
BelongMemberId	String	否	规则归属的成员账号ID（当Scope为cfwnat-xxx或Source Type/Dest Type为instance/tag时必填） 注意：此字段可能返回 null，表示取不到有效值。 示例值：1300448058

NatAcIRuleResp

NAT边界规则响应结构

被如下接口引用：DescribeNatAcIRules。

名称	类型	描述
RuleId	String	规则ID 示例值：uuid-xxx
Sequence	Integer	优先级 示例值：1
Direction	Integer	规则方向：0-出向，1-入向 示例值：1
SourceContent	String	源地址内容 示例值：10.0.0.0/24
SourceType	String	源类型 示例值：ip
SourceName	String	源资产名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：ins-xxx
TargetContent	String	目的地址内容 示例值：0.0.0.0/0
TargetType	String	目的类型 示例值：ip

名称	类型	描述
TargetName	String	目的资产名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：ins-xxx
Port	String	端口 注意：此字段可能返回 null，表示取不到有效值。 示例值：80
Protocol	String	协议 示例值：TCP
RuleAction	String	动作：accept/drop/log 示例值：accept
Description	String	规则描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：允许HTTP流量
Scope	String	规则生效范围 示例值：ALL
ScopeDesc	String	规则生效范围描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：全部实例
FwInslId	String	防火墙实例ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：cfwnat-xxx
CountryName	String	国家名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：中国
CityName	String	城市名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：北京
CountryCode	Integer	国家代码 注意：此字段可能返回 null，表示取不到有效值。 示例值：86
CityCode	Integer	城市代码 注意：此字段可能返回 null，表示取不到有效值。 示例值：131
CountryKey	String	国家键值 注意：此字段可能返回 null，表示取不到有效值。 示例值：CN
CityKey	String	城市键值 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
		示例值：BJ
IsRegion	Integer	是否地域规则：0-否，1-是 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
CloudCode	String	云厂商代码 注意：此字段可能返回 null，表示取不到有效值。 示例值：tencent
IsCloud	Integer	是否云厂商规则：0-否，1-是 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
ParamTemplateId	String	端口模板ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：ppm-xxx
ParamTemplateName	String	端口模板名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：端口模板-HTTP
Invalid	Integer	规则是否失效：0-有效，1-失效 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
BelongMember	MemberInfo	规则归属的成员账号 注意：此字段可能返回 null，表示取不到有效值。
CreateTime	String	创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2025-01-01 12:00:00
UpdateTime	String	更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2025-01-01 12:00:00
DnsParseCnt	Integer	域名数 示例值：0

OrganMemberItem

集团成员信息

被如下接口引用：DescribeOrganMembers。

名称	类型	描述
MemberId	String	成员 ID 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
		示例值：mem-1300448058-100027980407
AppId	String	成员账号 AppId 注意：此字段可能返回 null，表示取不到有效值。 示例值：1302385041
Uin	String	账号Uin 示例值：100027980407
Nickname	String	账号名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：子公司A
SubAccountCount	Integer	子账号数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：5
NodeName	String	所属组织架构节点名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：技术部
Role	String	成员身份：admin-管理员，delegatedAdmin-委派管理员，member-普通成员 注意：此字段可能返回 null，表示取不到有效值。 示例值：member
RoleDisplay	String	成员身份显示名称（前端展示用） 注意：此字段可能返回 null，表示取不到有效值。 示例值：普通成员
AccountGroup	AccountGroupInfo	所属账户组 注意：此字段可能返回 null，表示取不到有效值。
CfwManaged	Integer	云防火墙纳管状态：0-未纳管，1-已纳管 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
CfwShareRole	String	云防火墙共享角色：sharer-共享者，user-使用者，none-未设置 注意：此字段可能返回 null，表示取不到有效值。 示例值：sharer
CfwShareRoleDisplay	String	云防火墙共享角色显示名称（前端展示用） 注意：此字段可能返回 null，表示取不到有效值。 示例值：共享者
CfwSharerAppId	String	云防火墙共享者 AppId，成员角色为使用者时有值 注意：此字段可能返回 null，表示取不到有效值。 示例值：1300448058
CfwInstanceId	String	云防火墙计费实例 ID，非空表示已购买云防火墙 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
		示例值: cfwins-abc12345
PolicyAnalysisEnabled	Integer	策略分析权限: 0-关闭, 1-开启 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
MemberCreateTime	String	成员加入集团时间 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2025-01-15 10:30:00
JoinType	String	账号加入方式 示例值: invite

OrganSummary

集团概览

被如下接口引用: DescribeOrganSummary。

名称	类型	描述
GroupName	String	集团名称 示例值: 腾讯云集团
AdminInfo	MemberInfo	管理员账号信息
AdminCount	Integer	管理员/委派管理员数量 示例值: 2
JoinedMemberCount	Integer	已接入成员数 示例值: 10
MemberLimit	Integer	接入成员上限 (-1表示无上限) 示例值: 100
MemberLimitDisplay	String	接入成员上限显示 示例值: 100
CfwSharerCount	Integer	规格共享者数量 示例值: 3
CfwUserCount	Integer	规格使用者数量 示例值: 5
Departments	Array of String	部门名称列表 示例值: ["root"]
ManagedMemberCount	Integer	纳管账号数 示例值: 1

名称	类型	描述
ManagedProductCount	Integer	纳管产品数 示例值：1
CfwManageCount	Integer	纳管账号数 示例值：2

PolicyRisk

策略风险

被如下接口引用：DescribeRiskList。

名称	类型	必选	描述
Id	String	是	问题对应的唯一uuid 示例值：s1w45
RiskCategory	String	否	风险大类 示例值：无效规则
RiskSubCategory	String	否	风险子类 示例值：未命中规则
RuleType	String	否	规则分类 示例值：拒绝规则
RiskLevel	Integer	否	风险等级，0：低风险 1：中风险 2：高风险 示例值：2
Product	String	否	安全组 示例值：所属产品
SgRuleId	Array of String	否	风险包含的企业安全组规则ID 示例值：["fwm-04556"]
RuleCount	Integer	否	风险包含安全组ID内的问题规则数 示例值：2
SgId	Array of String	否	风险包含的安全组ID 示例值：["sg-888"]
RiskFeature	String	否	风险特征 示例值：无效规则
Suggestion	String	否	处置建议 示例值：基于业务实际需求，可考虑删除规则
Status	Integer	否	处置状态，0：未处理，1：已处理，2：忽略

名称	类型	必选	描述
			示例值：0
FoundTime	String	否	发现时间 示例值：2025-06-30 12:00:00
DisposalTime	String	否	处置时间 示例值：2025-06-30 13:00:00
Region	String	否	安全组地域 示例值：ap-guangzhou
Direction	String	否	Ingress进站，Egress出站 示例值：Egress
RiskReason	String	否	风险原因 示例值：源IP、目的IP都是1.1.1.1，2.2.2.2

ReceiveAccount

账号信息

被如下接口引用：DescribeStrategies, DescribeStrategy。

名称	类型	必选	描述
Uin	String	否	租户 uin 示例值：100000001
Nickname	String	否	租户名称 示例值：腾讯云
ReceiverType	Integer	否	0=账号uin, 1=账号组 示例值：1
Members	Array of MemberInfo	否	只有ReceiverType 是 1 时 才返回账号列表

RiskCategoryItem

风险分类统计项

被如下接口引用：DescribeRiskCategoryStats。

名称	类型	描述
CategoryId	String	风险大类ID 示例值：invalid_rule
CategoryName	String	风险大类名称 示例值：无效规则

名称	类型	描述
SubcategoryId	String	风险子类ID 示例值: loss_of_effect_rules
SubcategoryName	String	风险子类名称 示例值: 失效规则
RiskLevel	Integer	风险等级(0-低危,1-中危,2-高危) 示例值: 2
RiskLevelName	String	风险等级名称(低/中/高) 示例值: 高
Description	String	风险描述 示例值: 规则关联的资产实例、地址模板、资源标签等模板已被部分或全部删除
Suggestion	String	处置建议 示例值: 建议直接删除该无效规则
RuleCount	Integer	该类风险的规则数量 示例值: 15
TreatedCount	Integer	已处置数量 示例值: 8
IgnoredCount	Integer	已忽略数量 示例值: 2
UntreatedCount	Integer	待整改数量 示例值: 5
DisposalRate	Integer	整改率(百分比字符串) 示例值: 66
HasRisk	Integer	是否有未处理风险 -1: 未体检 0: 无风险 1: 有风险 示例值: 0
RemediationStatus	String	整改状态: Completed: 已整改完成 (整改率 100%) Incomplete: 未整改完成 (整改率 < 100%) -: 未体检/无数据 示例值: Completed

SecGroupRuleResp

企业安全组规则列表信息

被如下接口引用: DescribeSecurityGroupRules。

名称	类型	描述
OrderIndex	Integer	排序 示例值：1
RuleId	String	主键id 示例值：uuid
IpVersion	String	ip类型 示例值：ipv4
SourceId	String	源规则内容 示例值：1.1.1.1
SourceType	Integer	源规则类型 取值范围 0/1/2/3/4/5/6/7/8/9/100 0表示ip(net), 1表示VPC实例(instance) 2表示子网实例(instance) 3表示CVM实例(instance) 4表示CLB实例(instance) 5表示ENI实例(instance) 6表示数据库实例(instance) 7表示模板(template) 8表示标签(tag) 9表示地域(region) 100表示资产分组(resourcegroup) 示例值：0
TargetId	String	目的规则内容 示例值：1.1.1.1
TargetType	Integer	目的规则类型 取值范围 0/1/2/3/4/5/6/7/8/9/100 0表示ip(net), 1表示VPC实例(instance) 2表示子网实例(instance) 3表示CVM实例(instance) 4表示CLB实例(instance) 5表示ENI实例(instance) 6表示数据库实例(instance) 7表示模板(template) 8表示标签(tag) 9表示地域(region) 100表示资产分组(resourcegroup) 示例值：0
Protocol	String	协议名称 取值范围:TCP/ANY/ICMP/UDP ANY:表示所有 示例值：TCP

名称	类型	描述
Port	String	端口 示例值：80
Strategy	Integer	策略 示例值：1
Detail	String	描述 示例值：放通TCP 80
Region	String	地域 示例值：ap-guangzhou
ServiceTemplateId	String	服务模板id 示例值：tpm-3c140219
SouInstanceName	String	源资产名称 示例值：ins-3c140219
SouPublicIp	String	源资产公网ip 示例值：1.1.1.1
SouPrivateIp	String	源资产内网ip 示例值：1.1.1.1
SouCidr	String	源资产网段信息 示例值：1.1.1.1/10
SouParameterName	String	源模板名称 示例值：模板名称
InstanceName	String	目的资产名称 示例值：ins-3c140219
PublicIp	String	目的资产公网ip 示例值：1.1.1.1
PrivateIp	String	目的资产内网ip 示例值：1.1.1.1
Cidr	String	目的资产网段信息 示例值：1.1.1.1/10
ParameterName	String	目的模板名称 示例值：模板名称
ProtocolPortName	String	端口模板名称 示例值：模板名称
Id	Integer	规则id 等同RuleUuid 示例值：1
DnsParseCount	SgDnsParseCount	域名解析的IP统计

名称	类型	描述
Scope	String	规则生效范围 示例值：SG
IsNew	Integer	规则最新一次是否有改动 取值范围：0/1 0:否 1:是 示例值：1
BelongMember	MemberInfo	规则归属的成员账号（当FwGroupId为cfwg-xxx或SourceType/DestType为instance/tag时必须填） 注意：此字段可能返回 null，表示取不到有效值。

SecurityGroupRiskPolicy

安全组风险规则详情

被如下接口引用：DescribeRiskAnalysisDetails。

名称	类型	必选	描述
PolicyIndex	Integer	否	安全组规则索引号 示例值：0
Protocol	String	否	协议, 取值: TCP,UDP,ICMP,ICMPv6,ALL。 示例值：TCP
Port	String	否	端口(all, 离散port, range)。 示例值：22
ServiceTemplate	ServiceTemplateSpecification	否	端口ID或者协议端口组ID。 ServiceTemplate和Protocol+Port互斥。
CidrBlock	String	否	网段或IP(互斥)。 示例值：182.2.2.1
Ipv6CidrBlock	String	否	网段或IPv6(互斥)。 示例值：182.2.2.1
SecurityGroupId	String	否	安全组实例ID，例如：sg-ohuuioma。 示例值：sg-ohuuioma
AddressTemplate	AddressTemplateSpecification	否	IP地址ID或者ID地址组ID。
Action	String	否	动作：ACCEPT 或 DROP。 示例值：ACCEPT
PolicyDescription	String	否	安全组规则描述。 示例值：TEST
Version	String	否	安全组规则当前版本 示例值：10

名称	类型	必选	描述
Direction	String	否	规则方向, Egress出站规则, Ingress入站规则 示例值: Egress
ModifyTime	String	否	安全组最近修改时间。 示例值: 2025-07-01 00:00:00
Region	String	否	安全组所在地域 示例值: ap-guangzhou

SecurityGroupRule

规则数据结构描述

被如下接口引用: CreateSecurityGroupRule, CreateSecurityGroupRuleGroup, ModifySecurityGroupRule。

名称	类型	必选	描述
IpVersion	String	否	ip类型 示例值: ipv4
SourceContent	String	否	源地址 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "0.0.0.0/0"
SourceType	String	否	源类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "net"
DestContent	String	否	目的地址 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "0.0.0.0/0"
DestType	String	否	目的类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "net"
Protocol	String	否	协议 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ALL
Port	String	否	端口 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: -1/-1
ServiceTemplateId	String	否	模板 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "ippm-12dsa"

名称	类型	必选	描述
RuleAction	String	否	动作 注意：此字段可能返回 null，表示取不到有效值。 示例值："DROP"
Description	String	否	描述 注意：此字段可能返回 null，表示取不到有效值。 示例值："connect"
OrderIndex	Integer	否	优先级 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RuleId	String	否	rule id 注意：此字段可能返回 null，表示取不到有效值。 示例值：12222
Scope	String	否	生效范围 注意：此字段可能返回 null，表示取不到有效值。 示例值："SG"
ProtocolPortType	Integer	否	端口类型 示例值：0
BelongMemberId	String	否	规则归属的成员账号ID（当FwGroupId为cfwg-xxx或Source Type/DestType为instance/tag时必须填） 示例值：1300448058

SequenceIndex

规则序号调整结构

被如下接口引用：ModifyEdgeAclRuleSequence, ModifyNatAclRuleSequence, ModifyStrategySequence, ModifyVpcAclRuleSequence。

名称	类型	必选	描述
OrderIndex	Integer	否	原规则序号 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
NewOrderIndex	Integer	否	新规则序号 注意：此字段可能返回 null，表示取不到有效值。 示例值：2

ServiceTemplateSpecification

安全组服务模版

被如下接口引用：DescribeRiskAnalysisDetails。

名称	类型	必选	描述
ServiceId	String	否	协议端口ID，例如：ppm-f5n1f8da。 示例值：ppm-f5n1f8da
ServiceGroupId	String	否	协议端口组ID，例如：ppmg-f5n1f8da 示例值：ppmg-f5n1f8da

SgDnsParseCount

企业安全组域名解析的IP统计

被如下接口引用：DescribeRiskAnalysisDetails, DescribeSecurityGroupRules。

名称	类型	描述
ValidCount	Integer	有效下发的IP个数，离散数据 示例值：10
InvalidCount	Integer	未下发的IP个数，离散数据 示例值：10

SgRuleResp

规则列表响应数据结构

被如下接口引用：DescribeSecurityGroupRule。

名称	类型	描述
RuleId	String	规则Id 示例值: fwm_newsg_1300448058_443a085a07b6b578326217e765ecc579_17610187015422
Sequence	Integer	优先级 示例值: 2
Region	String	区域 示例值: ap-guangzhou
IpVersion	String	ip类型 示例值: ipv6
SrcContent	String	源内容 示例值: 2001:db8::/32
SrcType	String	源类型 示例值: 0
DstContent	String	目的内容 示例值: 2001:db8::/32
DstType	String	目的类型 示例值: 0
Protocol	String	协议 示例值: TCP
ProtocolPortType	Integer	协议端口参数模板 示例值: 0
ServiceTemplateId	String	协议端口参数模板id 示例值: ippm-13nsa
DstPort	String	端口段,支持单端口,多端口和端口段 示例值: 3306
RuleAction	String	策略, 1阻断, 2放行 示例值: ACCEPT
Detail	String	描述 示例值: Test IPv6 rule
RuleSource	String	规则来源, 0为用户控制台添加 示例值: 0
Scope	String	生效范围, 字节位, 1:SG 企业安全组, 2:LH 轻量服务器 示例值: SG

StrategyReq

创建策略的策略数据结构

被如下接口引用: CreateStrategy。

名称	类型	必选	描述
GroupId	String	否	规则组Id 示例值: "GR-1234"
Sequence	Integer	否	优先级 示例值: 1

StrategyResp

查询策略时策略列表参数

被如下接口引用: DescribeStrategies, DescribeStrategy。

名称	类型	描述
StrategyId	String	策略Id 示例值: SID-2JSS
GroupId	String	规则组Id 示例值: RG-10KJk
GroupName	String	规则组名称 示例值: 重保规则组1
RuleCount	Integer	规则数 示例值: 2
RuleStatus	Integer	策略状态 示例值: 1
ReceiveAccount	Array of ReceiveAccount	下发账号
Sequence	Integer	优先级 示例值: 1
ErrMsg	String	下发失败原因 示例值: 配额不足
ErrorType	String	下发失败原因类型 示例值: SG_QUOTA_LIMIT
UpdateTime	String	更新时间 示例值: 2025-10-14 11:23:32
CreateBy	String	创建人 示例值: cfw-test@123
UpdateBy	String	更新人 示例值: cfw-test@123
ExecArea	String	执行区域 示例值: post
CreateName	String	创建人名称 示例值: 天空之蓝
UpdateName	String	更新人名称 示例值: 天空之蓝
CreateTime	String	创建时间 示例值: 2025-10-14 11:23:32

VpcAclRule

VPC边界规则

被如下接口引用：CreateVpcAclRule, CreateVpcAclRuleGroup, ModifyVpcAclRule。

名称	类型	必选	描述
SourceContent	String	是	源地址内容 示例值：10.0.0.0/24
SourceType	String	是	源类型：ip/url/template/instance/tag 示例值：ip
DestContent	String	是	目的地址内容 示例值：0.0.0.0/0
DestType	String	是	目的类型：ip/url/template/instance/tag 示例值：ip
Protocol	String	是	协议：TCP/UDP/ICMP/ANY/HTTP/HTTPS/DNS/FTP等 示例值：TCP
RuleAction	String	是	动作：accept/drop/log 示例值：accept
OrderIndex	Integer	是	优先级（从1开始） 示例值：1
Edgeld	String	是	边界防火墙ID：ALL表示全局，CFWS-xxx表示指定边界 示例值：ALL
FwGroupId	String	是	防火墙实例ID（规则生效范围）：ALL-全局生效，ccn-xxx-云联网实例，cfwg-xxx-防火墙组实例 示例值：ALL
RuleId	String	否	规则ID（修改时必填） 示例值：uuid-xxx
Port	String	否	端口（ICMP协议时空） 示例值：80
Description	String	否	规则描述 示例值：允许HTTP流量
ParamTemplateId	String	否	端口模板ID 示例值：ppm-xxx
BelongMemberId	String	否	规则归属的成员账号ID（当FwGroupId为cfwg-xxx或SourceType/DestType为instance/tag时必填） 示例值：1300448058

VpcAclRuleResp

VPC边界规则响应结构

被如下接口引用：DescribeVpcAclRules。

名称	类型	描述
RuleId	String	规则ID 示例值: uuid-xxx
Sequence	Integer	优先级 示例值: 1
IpVersion	String	IP版本: ipv4或ipv6 示例值: ipv4
SourceContent	String	源地址内容 示例值: 10.0.0.0/24
SourceType	String	源类型 示例值: ip
SourceName	String	源资产名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ins-xxx
DestContent	String	目的地址内容 示例值: 0.0.0.0/0
DestType	String	目的类型 示例值: ip
DestName	String	目的资产名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ins-xxx
Protocol	String	协议 示例值: TCP
Port	String	端口 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 80
ParamTemplateId	String	端口模板ID 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ppm-xxx
ParamTemplateName	String	端口模板名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 端口模板-HTTP
RuleAction	String	动作: accept/drop/log 示例值: accept
Description	String	规则描述 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 允许HTTP流量

名称	类型	描述
Edgeld	String	边界防火墙ID 示例值：ALL
FwGroupId	String	防火墙实例ID 示例值：ALL
Invalid	Integer	规则是否失效：0-有效，1-失效 示例值：0
BelongMember	MemberInfo	规则归属的成员账号 注意：此字段可能返回 null，表示取不到有效值。
CreateTime	String	规则创建时间 示例值：2026-03-04 16:45:45
UpdateTime	String	规则修改时间 示例值：2026-03-04 16:45:45
DnsParseCnt	Integer	域名数 示例值：0
FwGroupName	String	防火墙组名称 示例值：ccn-klt3mzot

错误码

最近更新时间：2026-07-06 15:58:37

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。

错误码	说明
AuthFailure.UnauthorizedOperation	请求未授权。请参考 CAM 文档对鉴权的说明。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误（包括参数格式、类型等错误）。
InvalidParameterValue	参数取值错误。
InvalidRequest	请求 body 的 multipart 格式错误。
IpInBlacklist	IP 地址在黑名单中。
IpNotInWhitelist	IP 地址不在白名单中。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数。
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。

错误码	说明
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。
InternalError.Unknown	InternalError.Unknown
MissingParameter.CommonParam	MissingParameter.CommonParam
OperationDenied	操作被拒绝。
ResourceNotFound.NotFound	ResourceNotFound.NotFound