基础平台

产品文档

目录

基础平台 5
- 操作指南 5
  - 登录 5
  - 帐号信息 8
  - 账号注销 10
  - 主机安全评分 11
  - 安全设置 12
  - 资源管理 13
  - 备案管理 21
  - 常见问题 23
    - 常见登录问题 23
- API文档 26
  - 平台产品中心（tag） 26
    - 版本（2018-08-13） 26
      - API 概览 26
      - 调用方式 27
        - 接口签名v1 27
        - 接口签名v3 34
        - 请求结构 43
        - 返回结果 44
        - 公共参数 47
      - 写接口 49
        - 标签关联资源 49
        - 创建标签 51
        - 标签解绑资源 53
        - 删除标签 55
        - 批量修改资源关联的标签 57
      - 读接口 59
        - 查询资源关联标签 59
        - 查看资源关联的标签 61
        - 查询标签列表 63
      - 数据结构 65
      - 错误码 69
  - 组织与项目（tpo） 72
    - 版本（2020-09-20） 72
      - API 概览 72
      - 调用方式 74
        - 接口签名v1 74
        - 接口签名v3 81
        - 请求结构 90
        - 返回结果 91
        - 公共参数 94
      - Project相关接口 96
        - 项目添加用户及授权 96
        - 添加项目配额 98
        - 项目添加资源 100
        - 申请项目配额 102
        - 批量导入配额 104
        - 检查接口权限 106
        - 检查配额合法性 108
        - 创建项目 110
        - 删除项目 112
        - 删除项目配额 114
        - 项目移出资源 116
        - 查询项目配额申请历史 118
        - 查询各级产品 120
        - 查询用户的项目策略列表 122
        - 查询项目成员 124
        - 查询当前租户下所有非项目成员 126
        - 查询项目所有策略列表 128
        - 查询项目配额 130
        - 查询项目资源列表 132
        - 查询项目列表 134
        - 查询具有转入转出资源权限的项目列表 136
        - 查询所有资源区域 138
        - 修改单个用户的授权策略 140
        - 修改项目名称 142
        - 修改项目配额 144
        - 项目间移动资源 146
        - 项目名是否唯一 148
        - 从项目移除成员 150
      - 数据结构 152
      - 错误码 165
  - 访问管理（cam） 168
    - 版本（2019-01-16） 168
      - API 概览 168
      - 调用方式 170
        - 接口签名v1 170
        - 接口签名v3 177
        - 请求结构 186
        - 返回结果 187
        - 公共参数 190
      - 其他接口 192
        - 绑定多个策略到角色 192
        - 绑定权限策略到角色 194
        - 绑定多个角色到策略 196
        - 创建策略 198
        - 创建角色 200
        - 删除策略 202

- 删除角色 204
- 获取角色列表 206
- 解除绑定多个策略到用户组 208
- 解除绑定策略到多个用户组 210
- 解除绑定策略到多个用户 212
- 查看策略详情 214
- 获取角色详情 217
- 获取服务角色信息 219
- 查询用户组关联的策略列表 221
- 获取角色绑定的策略列表 223
- 查询策略关联的实体列表 225
- 查询策略列表 227
- 修改角色信任策略 230
- 更新策略 232
- 用户相关接口 235
- 获取CAM密码规则 235
- 子账户所属用户组列表 237
- 根据SecretId查询Uin 239
- 更新CAM密码规则 241
- 身份提供商接口 243
- 新增oauth配置 243
- 获取用户oauth标识 246
- 刷新用户userAccessToken 248
- 更新Oauth配置信息 250
- 验证用户userAccessToken 252
- 数据结构 254
- 错误码 290
- 地域信息（region） 295
- 版本（2022-06-27） 295
- API 概览 295
- 调用方式 296
- 接口签名v1 296
- 接口签名v3 303
- 请求结构 312
- 返回结果 313
- 公共参数 316
- 公有云地域 318
- 查询可用地域列表 318
- 查询地域信息的统一入口 320
- 查询可用区列表 322
- 数据结构 324
- 错误码 329
- 账号系统（account） 331
- 版本（2019-03-25） 331
- API 概览 331
- 调用方式 332
- 接口签名v1 332
- 接口签名v3 339
- 请求结构 348
- 返回结果 349
- 公共参数 352
- 账号系统 354
- 获取多因子登录参数 354
- 数据结构 356
- 错误码 364
- 项目目录管理（org） 367
- 版本（2021-10-01） 367
- API 概览 367
- 调用方式 368
- 接口签名v1 368
- 接口签名v3 375
- 请求结构 384
- 返回结果 385
- 公共参数 388
- 组织管理相关接口 390
- 添加组织 390
- 添加用户到组织并授权 392
- 删除组织 394
- 删除组织成员 396
- 查看组织成员 398
- 查询所有非组织成员 400
- 查询组织通用策略模板 402
- 查询组织下的项目列表 404
- 查询组织下的项目配额 406
- 查询组织下项目资源列表 408
- 查询组织 410
- 修改组织 412
- 修改组织成员策略 414
- 修改组织项目 416
- 数据结构 418
- 错误码 426
- 平台（platform） 429
- 版本（2019-03-14） 429
- API 概览 429
- 调用方式 430
- 接口签名v1 430
- 接口签名v3 437
- 请求结构 446
- 返回结果 447
- 公共参数 450
- platform 452

- 获取租户端主账号 452
- 获取租户端子账号 454
- 查询租户端账号 456
- 数据结构 458
- 错误码 461
- 密钥服务（sts） 463
- 版本（2018-08-13） 463
- API 概览 463
- 调用方式 464
- 接口签名v1 464
- 接口签名v3 471
- 请求结构 480
- 返回结果 481
- 公共参数 484
- 秘钥相关接口 486
- 申请扮演角色 486
- 获取控制台临时证书 489
- 获取联合身份临时访问凭证 492
- 获取用户临时证书（第三方开发商） 495
- 数据结构 497
- 错误码 498
- TCE 平台信息（location） 501
- 版本（2019-11-28） 501
- API 概览 501
- 调用方式 502
- 接口签名v1 502
- 接口签名v3 509
- 请求结构 518
- 返回结果 519
- 公共参数 522
- 地域相关接口 524
- 查询地域信息的统一入口 524
- 查询地域列表 526
- 查询可用区列表 528
- 设置故障地域列表 530
- 数据结构 532
- 错误码 536
- 租户端消息服务（tcloudmsg） 538
- 版本（2020-12-07） 538
- API 概览 538
- 调用方式 539
- 接口签名v1 539
- 接口签名v3 546
- 请求结构 555
- 返回结果 556
- 公共参数 559
- 产品自定义通道号 561
- PublicThemeAuto 561
- 数据结构 563
- 错误码 569

操作指南
登录

登录

- 运营端新建主账号首次登录时，需要进行密码重置。
- 登录时，系统会进行登录地检测，如果是非常用登录地，则需要进行手机验证码身份验证。
- 登录后，总览页可显示上次的登录时间、IP、方法和登录地区。

- 登录后，支持配置子用户在租户端登录的会话超时时间，租户的默认超时时间被置为运营端配置的时间。

> 说明：
>
> - 配置后，所有租户会话超时时间都会被修改。
> - 租户后续仍可修改各自的超时时间。运营端仅有重置的作用。
> - 运营端tcloud_session_timeout值需限制在30～1440之间，且为整数。

注册

注册时，邮箱地址即为登录帐号名 。

使用邮箱注册
邮箱地址
邮箱地址即为登录账号名
密码
确认密码
联系手机
+86
手机号码
验证码
获取验证码
提交

帐号信息

账号信息概述

通过账号信息可以查看登录账号的基本信息，包括账号昵称、账号ID、App ID、账号权限等信息。

功能介绍

查看账号信息

可以查看登录账号的账号名称、账号ID、App ID。
账号昵称：账号昵称可以编辑，不具有唯一性。
账号ID：账号ID是账号的唯一标识 。
App ID: App ID标识资源的归属，主账号和子账号的App ID相同。

账号权限

可以查看账号的身份是主账号还是子账号，主账号拥有全部权限，子账号权限由主账号配置。

资质审核

资质审核功能根据用户审核状态进行产品购买限制，不同用户之间的审核状态互不影响。只有通过了资质审核的用户，才有权限购买云产品。
若用户的资质认证状态为尚未认证，则需先认证。

具体操作：

- 单击提交资质认证，可选个人申请或企业申请。

- 若申请类型为个人，则需输入真实姓名、联系电话、住址和身份证相关的信息；若申请类型为企业，则需输入企业全称、企业联系人、企业地址、企业联系人电话和营业执照等信息。

- 单击提交，审核通过后，该账号就具备了云资源的购买资格。

账号注销

操作场景

本文将指导您如何注销云平台账号。

注意事项

账号一旦注销，将无法恢复。为了保证您的账号财产、云资源、账号信息等不受影响，在您提交注销账号申请前，请备份好账号下的所有数据。

主机安全评分

登录租户端管理控制台，在“概览”页面，查看右侧的"主机安全评分"卡片内容，显示主机安全信息，具体详情请参见主机安全。

安全设置

安全设置概述

安全设置可以管理登录账号下所有账号相关安全信息，包括密码、手机号码、邮箱、密码规则、登录保护等。

功能介绍

查看通用信息

在通用功能下可以修改密码、手机号码和邮箱。

- 修改密码：修改时必须要输入旧密码，新密码跟确认密码必须相同。密码规则可通过主账号在用户设置中设置，密码规则包括：包含大小写英文字母、数字、英文标点符号（如. / \ _等，空格除外）中的若干种，且不得包含用户名，同时可设置密码最小长度、密码过期时间、最近不可重复的历史密码数量。通过控制台修改密码适应场景为记得旧密码，密码即将过期或密码泄露情况下；修改限制：有旧密码且能登录控制台。
- 修改手机号码：只需要输入新的手机号码和验证码即可修改；修改手机号码使用场景：旧手机号码不再使用需要更换为新手机号；修改限制：必须能够登录控制台。
- 修改邮箱：修改邮箱与修改手机号码类似，只需要输入新的邮箱号与验证码即可修改；修改邮箱使用场景：邮箱号码不再使用需要更换为新邮箱；修改限制：必须能够登录控制台。

敏感操作

敏感操作用于设置账号的保护策略，进一步提高帐号安全。当执行敏感操作时，需输入短信验证码。
开启条件：已对接企业内部短信网关。
开启登录保护： 开启后，当用户在页面执行敏感操作时，将先进行身份验证，以保障您的账户安全。
关闭登录保护：关闭后，登录不需要手机验证。
敏感操作开启适用场景：政府、金融、银行等行业对保密要求较高建议开启登录保护。

登录保护

用户可以在安全设置页面设置登录保护类型。当前有三种身份认证类型：手机号验证码，虚拟MFA设备、第三方MFA设备。
虚拟MFA设备：基于 Google TOTP 算法实现， 用户需要下载 Google Authenticator 作为客户端获取口令，在登录时输入口令完成身份认证。
第三方MFA设备：第三方厂商提供的 MFA 认证工具，用户需要先将账号绑定到第三方系统并下载第三方系统的客户端，在登录时输入客户端中的口令完成身份认证。

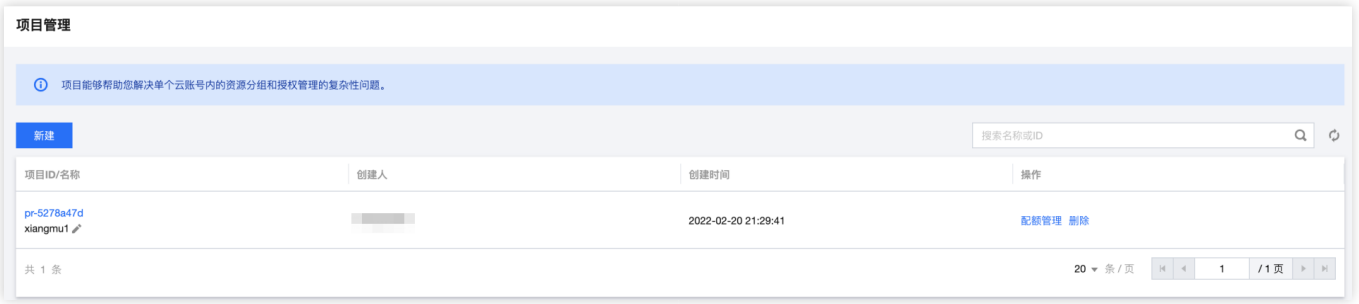
资源管理

登录租户端，单击全部云产品 > 管理与审计 > 资源管理进入资源管理控制台。

资源管理

新建项目

1. 选择项目管理，点击新建新建项目。



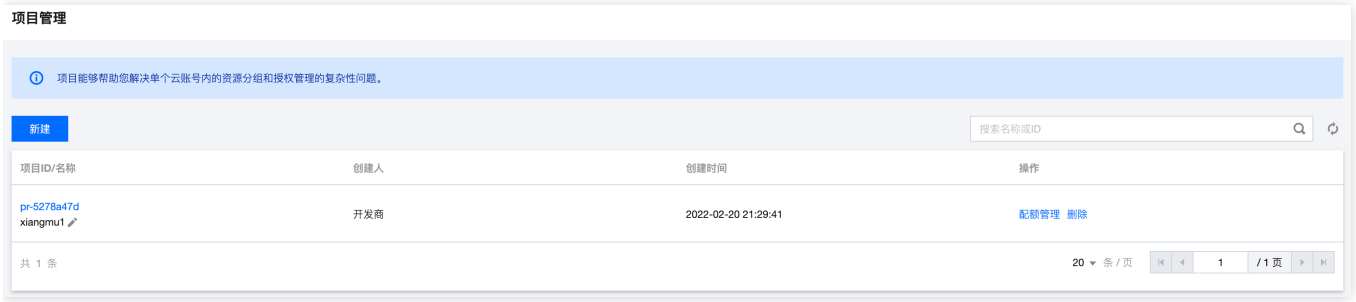
2. 在弹出的对话框中输入项目名称，单击确认。

3. 确认新建项目后可在当前页面看到基本信息，表示新建项目成功。

4. 单击项目名称右侧的编辑按钮，在弹出的对话框中修改项目名称。

项目列表

根据上述新建项目的步骤再添加一个项目，可在当前页面看到已添加所有项目的信息。

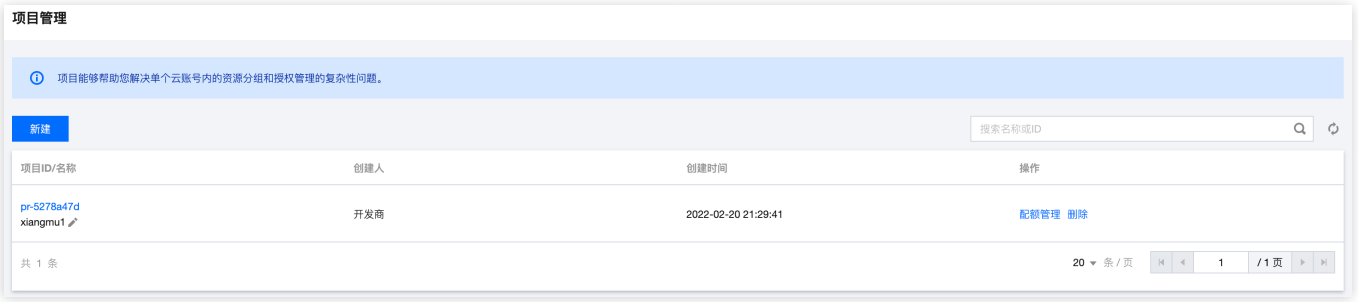


删除项目

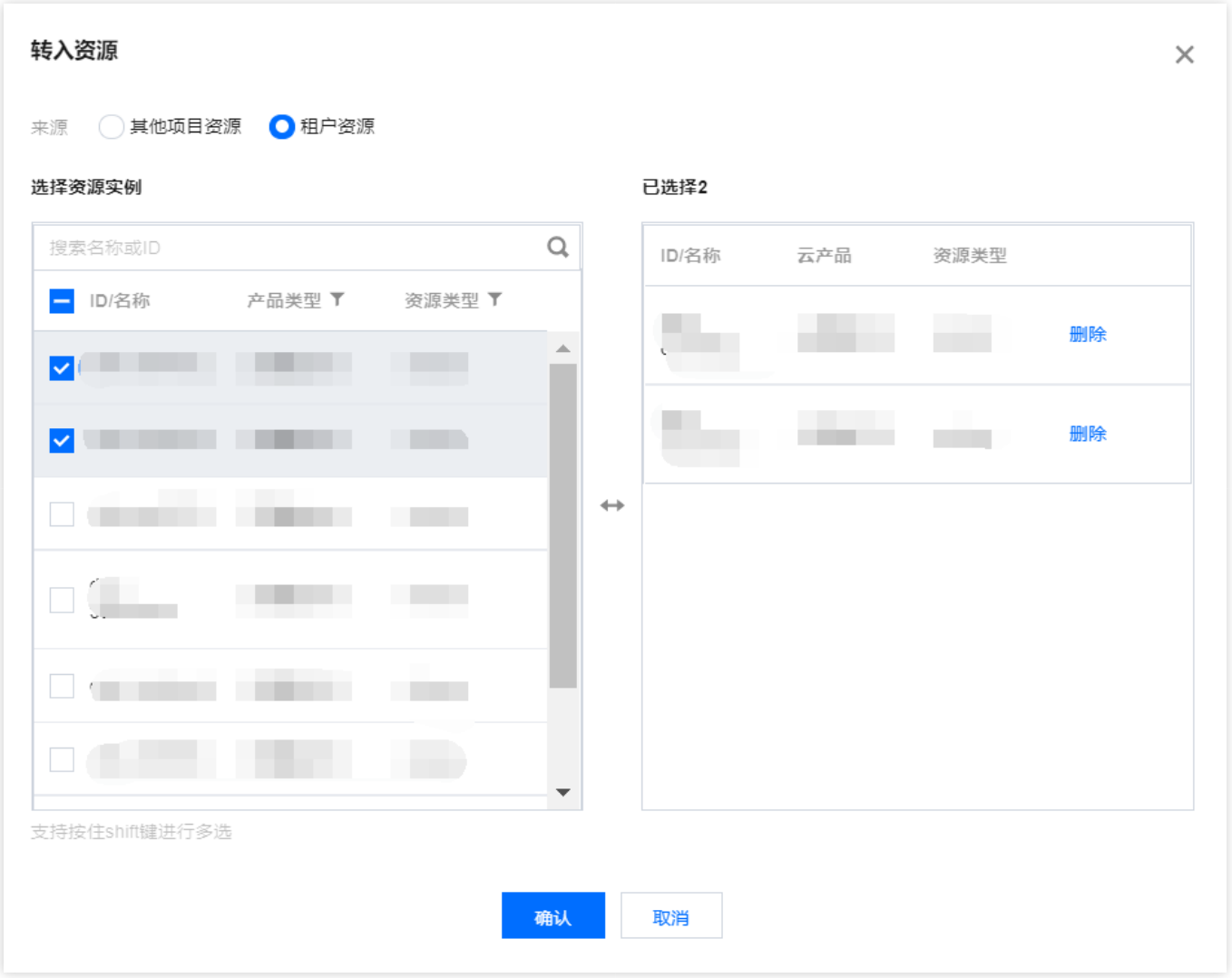
在项目列表页面，单击项目所在行的删除，在弹出的对话框中点击确认，在二次弹出的对话框里，单击确认。如果项目列表中已无之前的项目，表示删除项目成功。

资源管理

1. 在已添加的项目页面，单击项目ID/名称下方的 ID，对该项目进行资源管理。



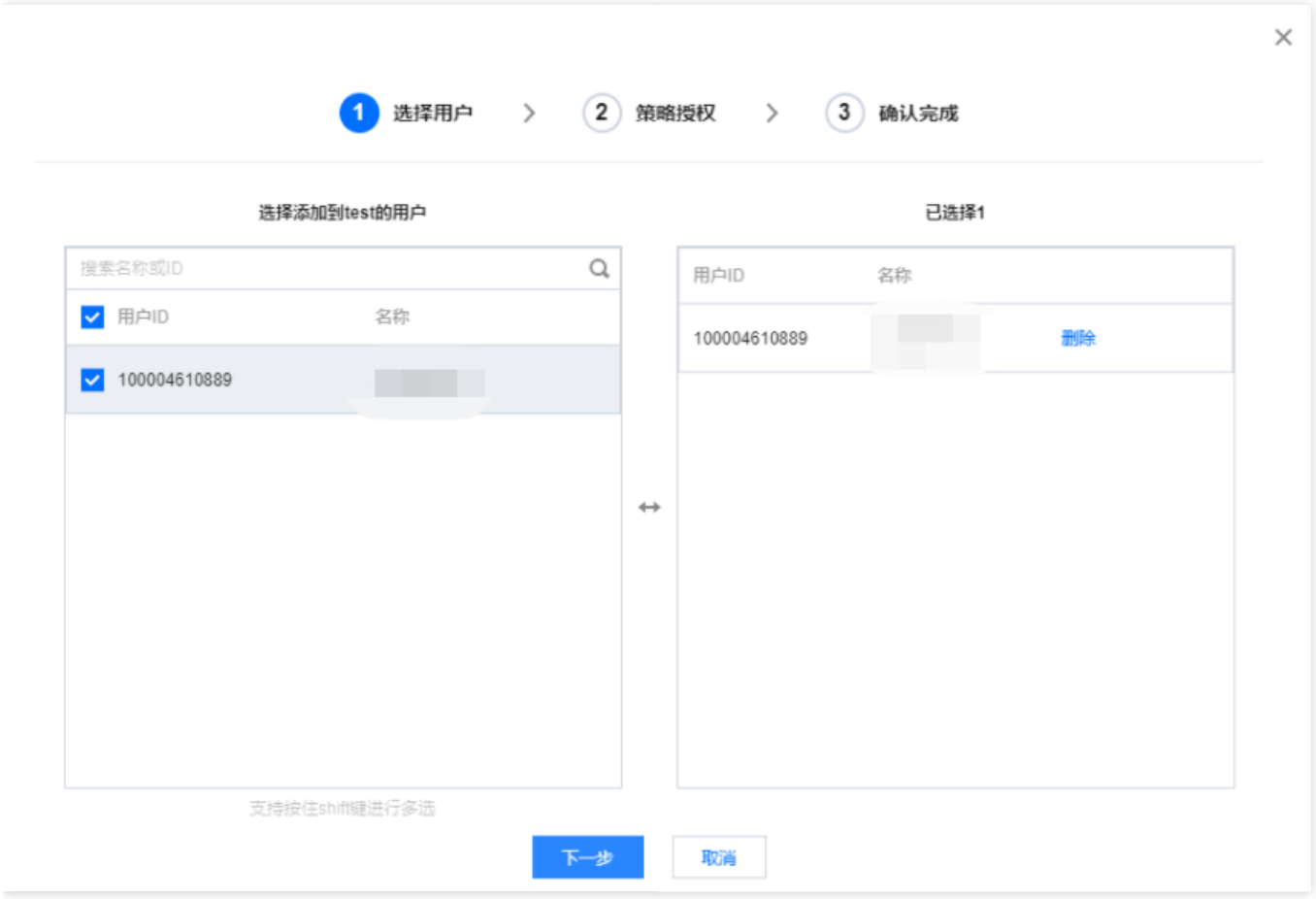
2. 将其他项目资源或租户资源转入到当前项目。
3. 点击转入资源。在弹出的对话框里，可根据业务需要选择从其他项目资源或租户资源转入到当前项目，核实后单击确认，可得到转入项目的信息。此处以转入租户资源举例。



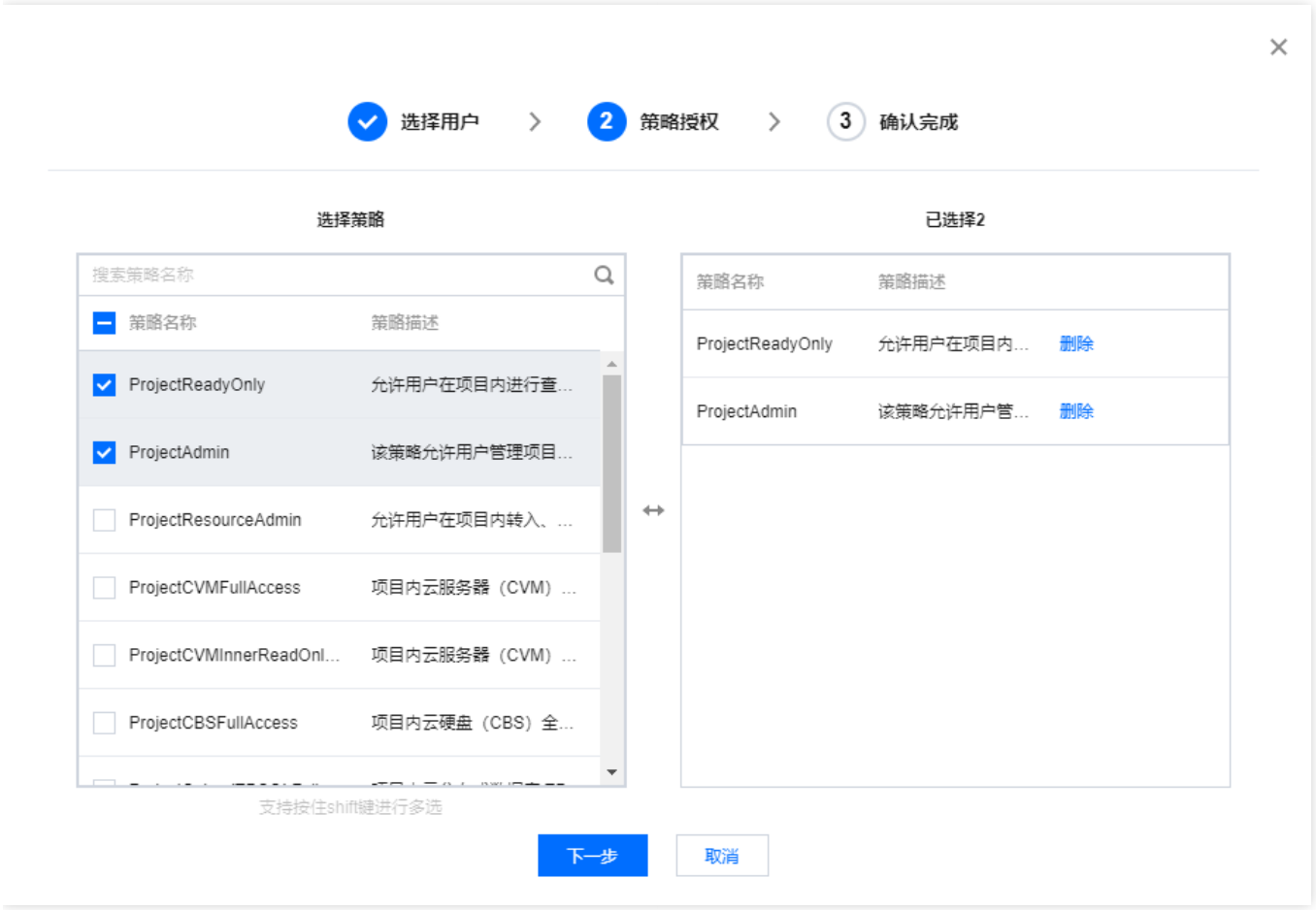
4. 新增的资源可根据实际情况点击管理对其进行管理。
5. 将当前项目的资源转移到其他项目，选中需要转出的资源后，单击转出资源，在新的弹窗选择目的项目后单击确认。如弹出转出资源成功的对话框，表示转出资源操作完成。

成员管理

1. 在【访问管理】>【用户管理】>【用户】中新建用户。
2. 在【云产品】>【管理与审计】>【资源管理】>【项目管理】页面，单击【项目ID/名称】下方的ID，对该项目进行成员管理。
3. 新增成员及授权。
   第一步：选择用户——选中列表中已有的用户添加到当前项目。



第二步：策略授权——根据所需角色的不同配置不同的授权。



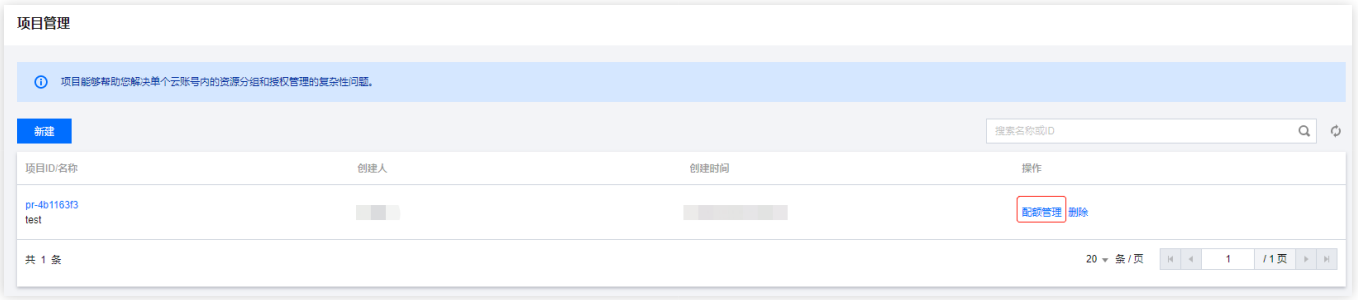
第三步：确认完成。

4. 移除。

在新增成员及授权后，如取消授权，可对已授权的成员进行移除操作。

配额管理

1. 在【项目管理】页面，对已添加的项目进行“配额管理”操作。



2. 在此页面可进行【新增配额项】、【查询】、【重置】等操作，针对配额支持下载配额模板填写相关数据后再导入配额进行配置。

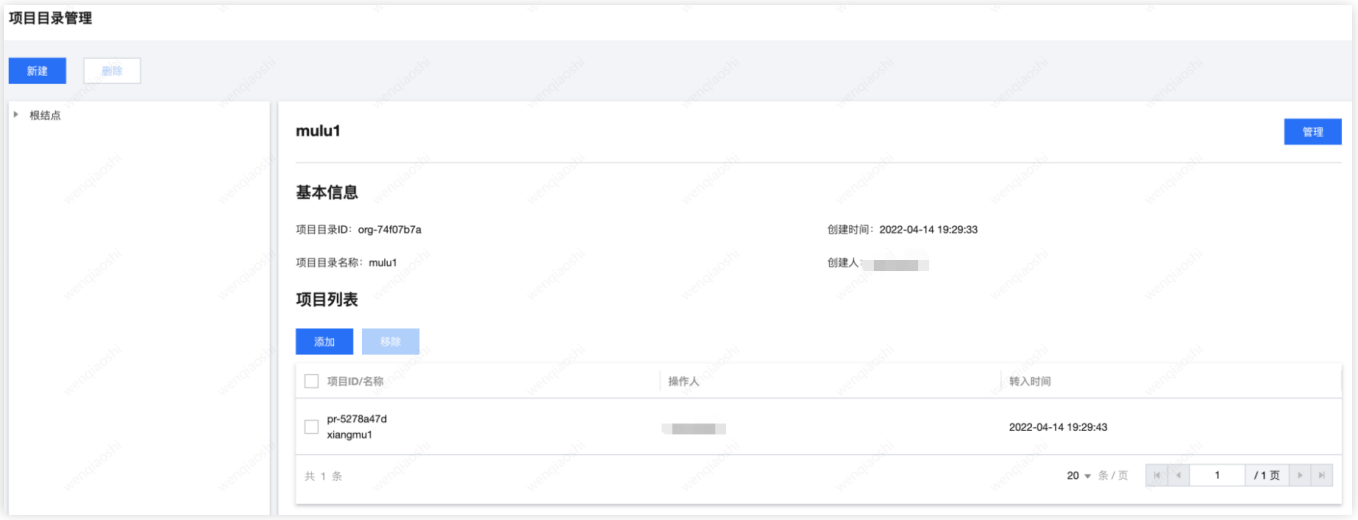
3. 在【云产品】>【管理与审计】>【资源管理】>【项目管理】页面，单击【项目ID/名称】下方的ID，可对该项目进行配额管理，进行查询和重置操作。

项目目录管理

新建项目目录

1. 单击【云产品】>【管理与审计】>【资源管理】进入资源管理控制台。

2. 选择【项目目录管理】>【新建】，创建项目目录。



3. 在弹出的窗口中选择项目目录归属，输入项目目录名称，单击【确认】。

4. 单击项目目录名称右侧的编辑，在弹出的对话框中修改项目目录名称。

资源管理

1. 单击目录页面基本信息区域右侧的【管理】，可分别查看【项目视图】、【资源视图】以及【项目配额视图】。

2. 单击右上角的 ，可下载视图列表。

备案管理

金融专区团体云用户需要通过公有云备案系统进行域名备案。您可以从金融专区复制备案码密钥，前往公有云系统验证。本文介绍金融专区生成备案码的详细操作。

限制条件

该操作仅适用于金融专区用户。

生成备案码

1. 登录金融专区控制台。
2. 在页面右上角，选择【账号】 > 【备案管理】，进入【备案管理】页面。
3. 在【备案管理】页面，单击【生成授权码】。
4. 选择【云资源类型】、【授权账号】、设置【备注】信息。

5. 单击【确定】。
6. 备案码生成后，在操作列单击【复制】，复制后即可前往公有云验证。

删除备案码

选择指定备案码，在操作列单击【删除】，确认无误后删除备案码。

常见问题
常见登录问题

常见登录问题

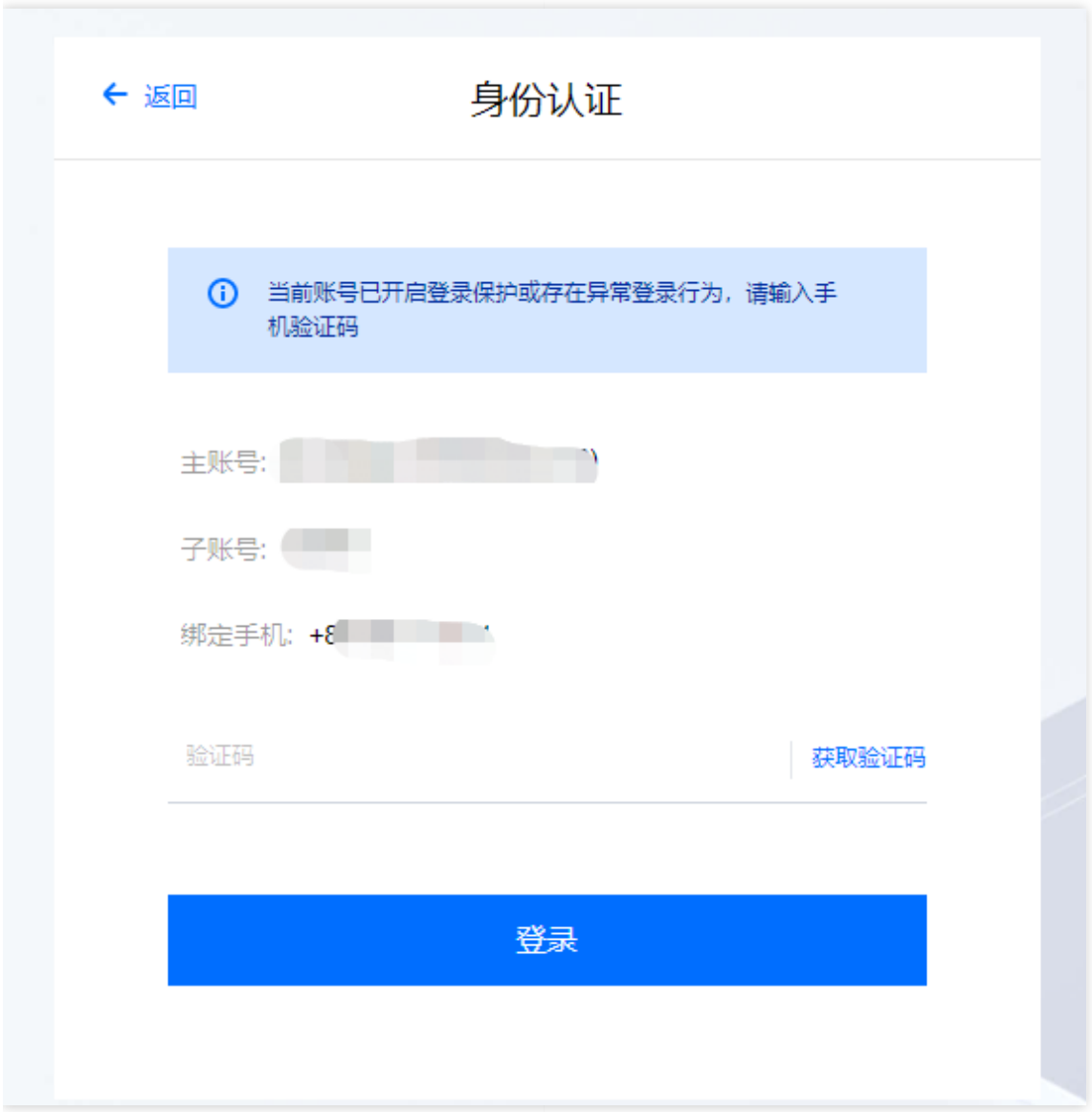
1、忘记密码，如何找回？

- 在登录页的忘记密码入口，单击忘记密码

- 找回主账号密码

2、登录保护场景下，未收到短信验证码

登录之后，若出现以下界面，说明管理员开启了登录保护。
单击获取验证码之后，若并未收到验证码，说明您的子账号关联的手机号不正确，需联系管理员，为您修改关联的手机号。


返回
身份认证
当前账号已开启登录保护或存在异常登录行为，请输入手机验证码
主账号:
子账号:
绑定手机: +8
验证码
获取验证码
登录

API文档
平台产品中心（tag）
版本（2018-08-13）
API 概览

API版本

V3

写接口

接口名称	接口功能
AddResourceTag	标签关联资源
CreateTag	创建标签
DeleteResourceTag	标签解绑资源
DeleteTag	删除标签
ModifyResourceTags	批量修改资源关联的标签

读接口

接口名称	接口功能
DescribeResourceTags	查询资源关联标签
DescribeResourceTagsByResourceIds	查看资源关联的标签
DescribeTags	查询标签列表

调用方式
接口签名v1

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐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。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
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

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' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    '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=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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.finance.cloud.tencent.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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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 http 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TCloudFinanceZone 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
- Node

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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

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

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

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

Java

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

public class CloudAPIDemo {
    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.finance.cloud.tencent.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.finance.cloud.tencent.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)).app
end("&");
        }
        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.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 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", "shjr"); // 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
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        '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)
```

接口签名v3

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：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 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，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。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

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

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- 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，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- 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）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 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
```

- 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 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- 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，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

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

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

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

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

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

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 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）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //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 = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\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 = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        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("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

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

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

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

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

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

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

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

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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。

公共参数

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

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	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。
X-TC-Timestamp	Integer	是	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。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	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。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

写接口
标签关联资源

1. 接口描述

接口请求域名： tag.api3.finance.cloud.tencent.com。

本接口用于给标签关联资源

默认接口请求频率限制：20次/秒。

接口更新时间：2024-12-06 16:56:29。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AddResourceTag
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
TagKey	是	否	String	标签键 示例值： TagKey_cvm
TagValue	是	否	String	标签值 示例值： TagValue_cvm
Resource	是	否	String	资源六段式描述 示例值： qcs::cvm:ap-region-x:uin/100004603715:instance/ins-qcmain

3. 输出参数

参数名称	类型	描述

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

4. 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
InvalidParameterValue.ResourceDescriptionError	资源描述错误
LimitExceeded.TagKey	用户创建标签键达到上限数 1000。
LimitExceeded.TagValue	单个标签键对应标签值达到上限数 1000。

创建标签

1. 接口描述

接口请求域名： tag.api3.finance.cloud.tencent.com。

本接口用于创建一对标签键和标签值

默认接口请求频率限制：20次/秒。

接口更新时间：2024-12-06 16:57:56。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateTag
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
TagKey	是	否	String	标签键 示例值： TagKey_cvm
TagValue	是	否	String	标签值 示例值： TagValue_cvm

3. 输出参数

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

4. 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
LimitExceeded.TagKey	用户创建标签键达到上限数 1000。
LimitExceeded.TagValue	单个标签键对应标签值达到上限数 1000。
InvalidParameterValue.ReservedTagKey	系统预留标签键禁止创建。
InvalidParameterValue.TagKeyCharacterIllegal	标签键包含非法字符。
InvalidParameterValue.TagKeyEmpty	标签键不能为空值。
InvalidParameterValue.TagKeyLengthExceeded	标签键长度超过限制。
InvalidParameterValue.TagValueCharacterIllegal	标签值包含非法字符。
InvalidParameterValue.TagValueLengthExceeded	标签值长度超过限制。
ResourceInUse.TagDuplicate	标签已存在。

标签解绑资源

1. 接口描述

接口请求域名： tag.api3.finance.cloud.tencent.com。

本接口用于解除标签和资源的关联关系

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-29 16:16:05。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteResourceTag
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
TagKey	是	否	String	标签键 示例值： key
Resource	是	否	String	资源六段式描述 示例值： qcs::cvm:ap-region-x:uin/100004603715:instance/qcmain

3. 输出参数

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

4. 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
ResourceNotFound.AttachedTagKeyNotFound	资源关联的标签键不存在。
InvalidParameterValue.ResourceDescriptionError	资源描述错误

删除标签

1. 接口描述

接口请求域名： tag.api3.finance.cloud.tencent.com。

本接口用于删除一对标签键和标签值

默认接口请求频率限制：20次/秒。

接口更新时间：2024-12-06 16:59:23。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteTag
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
TagKey	是	否	String	需要删除的标签键 示例值： TagKey_cvm
TagValue	是	否	String	需要删除的标签值 示例值： TagValue_cvm

3. 输出参数

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

4. 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.TagAttachedResource	已关联资源的标签无法删除。
ResourceNotFound.TagNonExist	标签不存在。

批量修改资源关联的标签

1. 接口描述

接口请求域名： tag.api3.finance.cloud.tencent.com。

本接口用于修改资源关联的所有标签

默认接口请求频率限制：200次/秒。

接口更新时间：2024-12-06 17:02:47。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyResourceTags
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Resource	是	否	String	资源的六段式描述 示例值： qcs::cvm:ap-region-x:uin/100004603715:instance/ins-qcmain
ReplaceTags	否	否	Array of Tag	需要增加或修改的标签集合。如果Resource描述的资源未关联输入的标签键，则增加关联；若已关联，则将该资源关联的键对应的标签值修改为输入值。本接口中ReplaceTags和DeleteTags二者必须存在其一，且二者不能包含相同的标签键 示例值：查看
DeleteTags	否	否	Array of TagKeyObject	需要解关联的标签集合。本接口中ReplaceTags和DeleteTags二者必须存在其一，且二者不能包含相同的标签键 示例值：查看

3. 输出参数

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

4. 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
InvalidParameterValue.ResourceDescriptionError	资源描述错误
LimitExceeded.TagKey	用户创建标签键达到上限数 1000。
LimitExceeded.TagValue	单个标签键对应标签值达到上限数 1000。
InvalidParameter.Tag	Tag参数错误。
InvalidParameterValue.DeleteTagsParamError	DeleteTags中不能包含ReplaceTags或AddTags中元素。

读接口
查询资源关联标签

1. 接口描述

接口请求域名： tag.api3.finance.cloud.tencent.com。

查询资源关联标签

默认接口请求频率限制：20次/秒。

接口更新时间：2020-05-19 06:20:33。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeResourceTags
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
CreateUin	否	否	Uint64	创建者uin 示例值： 100004611496
ResourceRegion	否	否	String	资源所在地域 示例值： ap-region-y
ServiceType	否	否	String	业务类型 示例值： ckafka
ResourcePrefix	否	否	String	资源前缀 示例值： ckafkaId
ResourceId	否	否	String	资源唯一标识 示例值： ckafka-o85jq584
Offset	否	否	Uint64	数据偏移量，默认为 0, 必须为Limit参数的整数倍 示例值： 0

参数名称	必选	允许NULL	类型	描述
Limit	否	否	Uint64	每页大小，默认为 15 示例值： 15
CosResourceId	否	否	Uint64	是否是Cos的资源id 示例值： 0

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	结果总数 示例值： 320
Offset	Uint64	数据位移偏量 示例值： 0
Limit	Uint64	每页大小 示例值： 15
Rows	Array of TagResource	资源标签 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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。

查看资源关联的标签

1. 接口描述

接口请求域名： tag.api3.finance.cloud.tencent.com。

用于查询已有资源标签键值对

默认接口请求频率限制：20次/秒。

接口更新时间：2024-12-06 17:00:18。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeResourceTagsByResourceIds
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ServiceType	是	否	String	业务类型 示例值： cvm
ResourcePrefix	是	否	String	资源前缀 示例值： instance
ResourceIds	是	否	Array of String	资源唯一标记 示例值： ins-qcmain
ResourceRegion	是	否	String	资源所在地域 示例值： ap-region-x
Offset	否	否	Uint64	数据偏移量，默认为 0, 必须为Limit参数的整数倍 示例值： 0
Limit	否	否	Uint64	每页大小，默认为 15 示例值： 15

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	结果总数 示例值： 15
Offset	Uint64	数据位移偏量 示例值： 0
Limit	Uint64	每页大小 示例值： 15
Tags	Array of TagResource	标签列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InvalidParameterValue.RegionInvalid	地域错误。
InvalidParameterValue.ResourcePrefixInvalid	资源前缀错误。
InvalidParameterValue.ServiceTypeInvalid	业务类型错误。
InvalidParameterValue.UinInvalid	Uin参数不合法。
InvalidParameterValue.ResourceIdSizeInvalid	资源数量超过上限(50)

查询标签列表

1. 接口描述

接口请求域名： tag.api3.finance.cloud.tencent.com。

用于查询已建立的标签列表。

默认接口请求频率限制：20次/秒。

接口更新时间：2024-12-06 17:01:38。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeTags
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
TagKey	否	否	String	标签键,与标签值同时存在或同时不存在，不存在时表示查询该用户所有标签 示例值： TagKey_cvm
TagValue	否	否	String	标签值,与标签键同时存在或同时不存在，不存在时表示查询该用户所有标签 示例值： TagValue_cvm
Offset	否	否	Uint64	数据偏移量，默认为 0, 必须为Limit参数的整数倍 示例值： 0
Limit	否	否	Uint64	每页大小，默认为 15 示例值： 15
CreateUin	否	否	Uint64	创建者用户 Uin，不传或为空只将 Uin 作为条件查询 示例值： 121

参数名称	必选	允许NULL	类型	描述
TagKeys	否	否	Array of String	标签键数组,与标签值同时存在或同时不存在，不存在时表示查询该用户所有标签,当与TagKey同时传递时只会本值 示例值： Key

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	结果总数 示例值： 1
Offset	Uint64	数据位移偏量 示例值： 0
Limit	Uint64	每页大小 示例值： 15
Tags	Array of TagWithDelete	标签列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InvalidParameterValue.UinInvalid	Uin参数不合法。

数据结构

ResourceIdTag

资源标签键值

被如下接口引用：DescribeResourceTagsByTagKeys、DescribeResourcesByTags

名称	必选	允许NULL	类型	描述
ResourceId	是	否	String	资源唯一标识 示例值： 50
TagKeyValues	是	否	String	标签键值对 示例值： TagValue_cvm

TagWithDelete

表示一个标签键值对以及是否允许删除

被如下接口引用：DescribeTags

名称	必选	允许NULL	类型	描述
TagKey	是	否	String	标签键 示例值： TagKey1
TagValue	是	否	String	标签值 示例值： TagValue1
CanDelete	是	否	Uint64	是否可以删除 示例值： TagKey1

ServiceTypeFilter

查询资源

被如下接口引用：DescribeResourcesBindTag

名称	必选	允许NULL	类型	描述
ServiceType	是	否	String	服务类型

名称	必选	允许NULL	类型	描述
				示例值： "cvm"
ResourcePrefix	是	否	Array of String	资源前缀 示例值： ["instance"]

TagFilter

tag过滤数组多个是与的关系

被如下接口引用：DescribeResourcesBindTag、DescribeResourcesByTags

名称	必选	允许NULL	类型	描述
TagKey	是	否	String	标签键 示例值： TagKey
TagValue	否	否	Array of String	标签值数组 多个值的话是或的关系 示例值： TagValue

TagKeyObject

标签键对象

被如下接口引用：ModifyResourceTags

名称	必选	允许NULL	类型	描述
TagKey	是	否	String	标签键 示例值： TagKey
TagValue	否	否	String	标签值 示例值： TagValue

TagResource

标签键值对以及资源ID

被如下接口引用：DescribeResourceTags、DescribeResourceTagsByResourceIds、DescribeResourcesBindTag

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
TagKey	是	否	String	标签键 示例值： TagKey
TagValue	是	否	String	标签值 示例值： TagValue
ResourceId	是	否	String	资源ID 示例值： ResourceId
TagKeyMd5	是	否	String	标签键MD5值 示例值： md5res
TagValueMd5	是	否	String	标签值MD5值 示例值： md5res
ServiceType	是	是	String	资源类型 示例值： cos

Tag

表示一个标签键值对

被如下接口引用：BatchCreateTag、ModifyResourceTags

名称	必选	允许NULL	类型	描述
TagKey	是	否	String	标签键 示例值： TagKey
TagValue	是	否	String	标签值 示例值： TagValue

TagEntry

标签条目

被如下接口引用：DescribeTagValues

名称	必选	允许NULL	类型	描述
TagKey	否	否	String	标签键 示例值： TagKey_cvm

名称	必选	允许NULL	类型	描述
TagValue	否	否	String	标签值 示例值： TagValue

错误码

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 cor
rect."
        },
        "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
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接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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
ResourceNotFound.TagNonExist	标签不存在。
InvalidParameterValue.TagFiltersLengthExceeded	tagFilters不存在或长度大于6
InvalidParameterValue.ResourcePrefixInvalid	资源前缀错误。
InvalidParameterValue.TagValueCharacterIllegal	标签值包含非法字符。

错误码	说明
InvalidParameter.Tag	Tag参数错误。
ResourceNotFound.AttachedTagKeyNotFound	资源关联的标签键不存在。
FailedOperation.TagAttachedResource	已关联资源的标签无法删除。
InvalidParameterValue.TagKeyCharacterIllegal	标签键包含非法字符。
InvalidParameterValue.TagKeyEmpty	标签键不能为空值。
LimitExceeded.TagValue	单个标签键对应标签值达到上限数 1000。
InvalidParameterValue.ServiceTypeInvalid	业务类型错误。
InvalidParameterValue.TagValueLengthExceeded	标签值长度超过限制。
InvalidParameterValue.RegionInvalid	地域错误。
InvalidParameterValue.UinInvalid	Uin参数不合法。
InvalidParameterValue.ReservedTagKey	系统预留标签键禁止创建。
InvalidParameterValue.ResourceDescriptionError	资源描述错误
ResourceInUse.TagDuplicate	标签已存在。
InvalidParameterValue.DeleteTagsParamError	DeleteTags中不能包含ReplaceTags或AddTags中元素。
InvalidParameterValue.TagKeyLengthExceeded	标签键长度超过限制。
InvalidParameterValue.TagListEmpty	tagList参数为空
InvalidParameterValue.ResourceIdSizeInvalid	资源数量超过上限(50)
InvalidParameterValue.TagFilters	TagFilters参数有误
LimitExceeded.TagKey	用户创建标签键达到上限数 1000。

组织与项目（tpo）
版本（2020-09-20）
API 概览

API版本

V3

Project相关接口

接口名称	接口功能
AddProjectMemberPolicy	项目添加用户及授权
AddProjectQuota	添加项目配额
AddProjectResource	项目添加资源
ApplyProjectQuota	申请项目配额
BatchAddProjectQuota	批量导入配额
CheckActionPermission	检查接口权限
CheckProjectQuotas	检查配额合法性
CreateProject	创建项目
DeleteProject	删除项目
DeleteProjectQuota	删除项目配额
DeleteProjectResource	项目移出资源
DescribeApplyProjectQuotaHistory	查询项目配额申请历史
DescribeProductTree	查询各级产品
DescribeProjectMemberPolicies	查询用户的项目策略列表
DescribeProjectMembers	查询项目成员
DescribeProjectNonMembers	查询当前租户下所有非项目成员

接口名称	接口功能
DescribeProjectPolicies	查询项目所有策略列表
DescribeProjectQuotas	查询项目配额
DescribeProjectResources	查询项目资源列表
DescribeProjects	查询项目列表
DescribeResourceAdminProjects	查询具有转入转出资源权限的项目列表
DescribeResourceRegions	查询所有资源区域
ModifyProjectMemberPolicy	修改单个用户的授权策略
ModifyProjectName	修改项目名称
ModifyProjectQuota	修改项目配额
MoveProjectResource	项目间移动资源
ProjectNameExists	项目名是否唯一
RemoveProjectMember	从项目移除成员

调用方式
接口签名v1

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐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。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
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

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' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    '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=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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.finance.cloud.tencent.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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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 http 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TCloudFinanceZone 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
- Node

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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

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

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

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

Java

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

public class CloudAPIDemo {
    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.finance.cloud.tencent.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.finance.cloud.tencent.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)).app
end("&");
        }
        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.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 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", "shjr"); // 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
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        '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)
```

接口签名v3

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：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 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，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。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

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

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- 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，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- 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）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 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
```

- 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 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- 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，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

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

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

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

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

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

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 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）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //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 = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\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 = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        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("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

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

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

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

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

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

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

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

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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。

公共参数

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

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	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。
X-TC-Timestamp	Integer	是	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。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	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。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

Project相关接口
项目添加用户及授权

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

项目添加用户及授权

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AddProjectMemberPolicy
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
Uins	是	否	Array of Uint64	用户ID列表 示例值： [1000,1001]
PolicyNames	是	否	Array of String	待添加的策略名 示例值： ["policygen-20141112201913","policygen-20141112201914"]

3. 输出参数

参数名称	类型	描述
SuccessfulUins	Array of UinPolicy	成功列表 示例值：查看

参数名称	类型	描述
FailedUins	Array of UinPolicy	失败列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
InternalError.DatabaseError	内部数据库错误
ResourceNotFound.ProjectNotFoundError	项目不存在
UnauthorizedOperation.AuthFailedError	鉴权失败

添加项目配额

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

添加项目配额

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AddProjectQuota
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
ProductCode	是	否	String	一级产品定义 示例值： p_cvm
SubProductCode	否	否	String	二级产品定义 示例值： sp_cvm_sh1
BillingItemCode	否	否	String	三级产品定义 示例值： v_cvm_cpu
SubBillingItemCode	否	否	String	四级产品定义 示例值： sv_cbs_memspace_common
ProductQuota	否	否	Uint64	一级产品配额 示例值： 100

参数名称	必选	允许NULL	类型	描述
SubProductQuota	否	否	Uint64	二级产品配额 示例值： 0
BillingItemQuota	否	否	Uint64	三级产品配额 示例值： 100
SubBillingItemQuota	否	否	Uint64	四级产品配额 示例值： 0

3. 输出参数

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败
InvalidParameter.InvalidProjectQuota	无效的配额

项目添加资源

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

向项目里添加资源

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AddProjectResource
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
ResourceList	是	否	Array of TransferResource	添加的资源列表 示例值：查看

3. 输出参数

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败
InvalidParameter.UnsupportedProductCodeError	不支持的ProductCode
ResourceNotFound.ProjectResourceNotFound	项目内资源不存在

申请项目配额

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

申请项目配额

默认接口请求频率限制：20次/秒。

接口更新时间：2023-05-12 16:52:18。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ApplyProjectQuota
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
ProductCode	是	否	String	产品code 示例值： p_tct
ProductName	是	否	String	产品名称 示例值： tct
QuotaKey	是	否	String	配额key 示例值： p_cvm###
QuotaName	是	否	String	配额名称 示例值： 云数据库Redis
QuotaValue	是	否	Uint64	配额数 示例值： 100

参数名称	必选	允许NULL	类型	描述
QuotaUsed	是	否	Uint64	已用配额数 示例值： 100
ApplyQuotaValue	是	否	Uint64	申请的配额数 示例值： 100
Reason	是	否	String	申请理由 示例值： 申请100CVM

3. 输出参数

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

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.EmptyParameter	有参数为空
InvalidParameter.InvalidParameter	参数不对
InvalidParameter.InvalidProjectQuota	无效的配额
InternalError.InvalidHTTPCode	无效httpcode
InternalError.ThirdPartServiceBadRequest	服务调用失败
InvalidParameter.UnsupportedProductCodeError	不支持的ProductCode
InvalidParameter.UsedQuotaNotEnough	新的配额要大于已使用的配额
ResourceNotFound.ProjectNotFoundError	项目不存在
UnauthorizedOperation.AuthFailedError	鉴权失败
InternalError.DatabaseError	内部数据库错误

批量导入配额

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

批量导入配额

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-21 16:55:24。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：BatchAddProjectQuota
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
QuotaSet	是	否	Array of AddQuota	待添加的配额 示例值：查看

3. 输出参数

参数名称	类型	描述
AddList	Array of AddQuota	添加列表 示例值：查看
AddSuccess	Bool	是否添加成功，传1表示成功，传0表示不成功 示例值： true

参数名称	类型	描述
UpdateList	Array of AddQuota	更新列表 示例值：查看
UpdateSuccess	Bool	是否更新成功，传1表示成功，传0表示不成功 示例值： true
ErrorList	Array of AddQuota	错误列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败

检查接口权限

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

检查接口权限

默认接口请求频率限制：20次/秒。

接口更新时间：2023-05-12 16:52:28。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CheckActionPermission
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
CheckAction	是	否	String	需要检查的接口 示例值： ""
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11

3. 输出参数

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

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.EmptyParameter	有参数为空
InvalidParameter.InvalidParameter	参数不对
InvalidParameter.InvalidProjectQuota	无效的配额
InternalError.InvalidHTTPCode	无效httpcode
InternalError.ThirdPartServiceBadRequest	服务调用失败
InvalidParameter.UnsupportedProductCodeError	不支持的ProductCode
InvalidParameter.UsedQuotaNotEnough	新的配额要大于已使用的配额
ResourceNotFound.ProjectNotFoundError	项目不存在
UnauthorizedOperation.AuthFailedError	鉴权失败
InternalError.DatabaseError	内部数据库错误

检查配额合法性

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

检查配额合法性，转换配额名称到编码

默认接口请求频率限制：20次/秒。

接口更新时间：2023-03-07 11:20:32。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CheckProjectQuotas
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
QuotaSet	否	否	Array of QuotaNames	配额名 示例值：查看

3. 输出参数

参数名称	类型	描述
QuotaSet	Array of QuotaChecking	配额列表 示例值：查看

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败

创建项目

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

创建项目

默认接口请求频率限制：20次/秒。

接口更新时间：2023-07-24 14:40:24。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateProject
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectName	是	否	String	项目名 示例值： pr1
Organization	否	否	String	归属组织 示例值： org1
ProjectDescription	否	否	String	项目描述 示例值： prdesc

3. 输出参数

参数名称	类型	描述
ProjectId	String	项目ID 示例值： pr-dcd34c11
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ProjectNameTooLong	项目名过长
InvalidParameter.EmptyParameter	有参数为空
InternalError.ComponentFail	调用其它组件接口失败

删除项目

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

删除项目

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteProject
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11

3. 输出参数

参数名称	类型	描述
ProjectId	String	项目ID 示例值： pr-dcd34c11
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.DatabaseError	内部数据库错误
ResourceNotFound.ProjectNotFoundError	项目不存在
FailedOperation.ProjectResourceNotEmpty	项目内资源不为空
FailedOperation.ProjectQuotaNotEmpty	项目配额不为空

删除项目配额

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

删除项目配额

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteProjectQuota
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
ResourceList	是	否	Array of DeleteQuota	资源列表 示例值：查看

3. 输出参数

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败

项目移出资源

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

将资源移出项目

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteProjectResource
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
ResourceList	是	否	Array of TransferResource	资源列表 示例值：查看

3. 输出参数

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败
InvalidParameter.UnsupportedProductCodeError	不支持的ProductCode
FailedOperation.ProjectCountError	资源的项目数过多
ResourceNotFound.ProjectResourceNotFound	项目内资源不存在

查询项目配额申请历史

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询项目配额申请历史

默认接口请求频率限制：20次/秒。

接口更新时间：2023-05-12 16:52:11。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeApplyProjectQuotaHistory
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
PageNumber	否	否	Uint64	页码 示例值： 100
PageSize	否	否	Uint64	每页个数 示例值： 100

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数 示例值： 100

参数名称	类型	描述
ApplyProjectQuotaHistorySet	Array of ApplyProjectQuotaHistory	申请配额历史列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.EmptyParameter	有参数为空
InvalidParameter.InvalidParameter	参数不对
InvalidParameter.InvalidProjectQuota	无效的配额
InternalError.InvalidHTTPCode	无效httpcode
InternalError.ThirdPartServiceBadRequest	服务调用失败
InvalidParameter.UnsupportedProductCodeError	不支持的ProductCode
InvalidParameter.UsedQuotaNotEnough	新的配额要大于已使用的配额
ResourceNotFound.ProjectNotFoundError	项目不存在
UnauthorizedOperation.AuthFailedError	鉴权失败
InternalError.DatabaseError	内部数据库错误

查询各级产品

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询各级产品

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeProductTree
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProductCode	否	否	String	一级产品 示例值： p_cvm
SubProductCode	否	否	String	二级产品 示例值： sp_cvm_vself2
BillingItemCode	否	否	String	三级产品 示例值： v_eip_hour
SubBillingItemCode	否	否	String	四级产品 示例值： sv_eip_hour
AccessTypes	否	否	Array of String	资源获取方式，billing表示从计费同步资源ID，local表示直接从垂直产品同步资源ID 示例值： ["local"]
Status	否	否	String	配额项状态，为1表示仅展示有效配额，为0表示展示所有配额 示例值： 1

3. 输出参数

参数名称	类型	描述
ProductSet	Array of Product	一级产品列表 示例值：查看
SubProductSet	Array of Product	二级产品列表 示例值：查看
BillingItemSet	Array of Product	三级产品列表 示例值：查看
SubBillingItemSet	Array of Product	四级产品列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode

查询用户的项目策略列表

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询用户的项目策略列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeProjectMemberPolicies
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
AccountUin	是	否	Uint64	用户Uin 示例值： 100
Filter	否	否	PolicyFilter	过滤参数 示例值：查看

3. 输出参数

参数名称	类型	描述
OwnedPolicies	Array of Policy	已添加策略 示例值：查看

参数名称	类型	描述
Policies	Array of Policy	未添加策略 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败

查询项目成员

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询项目成员

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeProjectMembers
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
PageNumber	否	否	Uint64	页码 示例值： 100
PageSize	否	否	Uint64	每页数量 示例值： 100
Filter	否	否	ProjectMemberFilter	搜索条件 示例值：查看

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数 示例值： 100
MemberSet	Array of ProjectMember	成员列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.DatabaseError	内部数据库错误
ResourceNotFound.ProjectNotFoundError	项目不存在
UnauthorizedOperation.AuthFailedError	鉴权失败

查询当前租户下所有非项目成员

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询当前租户下所有非项目成员

默认接口请求频率限制：20次/秒。

接口更新时间：2024-10-14 15:04:32。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeProjectNonMembers
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
PageNumber	否	否	Uint64	页码 示例值： 100
PageSize	否	否	Uint64	每页数量 示例值： 100
Filter	否	否	ProjectMemberFilter	搜索参数 示例值：查看

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数 示例值： 100
MemberSet	Array of ProjectMember	成员列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败
ResourceNotFound.ProjectNotFoundError	项目不存在

查询项目所有策略列表

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询项目所有策略列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeProjectPolicies
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
PageNumber	否	否	Uint64	页码 示例值： 100
PageSize	否	否	Uint64	每页数量 示例值： 100
Filter	否	否	PolicyFilter	模糊查询 示例值：查看

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数

参数名称	类型	描述
		示例值： 100
PolicySet	Array of Policy	策略列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.DatabaseError	内部数据库错误

查询项目配额

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询项目配额

默认接口请求频率限制：20次/秒。

接口更新时间：2024-10-10 20:03:50。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeProjectQuotas
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
PageNumber	否	否	Uint64	页码 示例值： 100
PageSize	否	否	Uint64	每页数量 示例值： 100
Filter	否	否	ProjectQuotaFilter	搜索条件 示例值：查看
Status	否	否	String	配额项状态，为1表示仅展示有效配额，为0表示展示所有配额 示例值： 1

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数 示例值： 100
QuotaSet	Array of ProjectQuota	配额列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败

查询项目资源列表

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询项目资源列表

默认接口请求频率限制：20次/秒。

接口更新时间：2023-10-13 11:18:22。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeProjectResources
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
PageNumber	否	否	Uint64	页码 示例值： 100
PageSize	否	否	Uint64	每页数量 示例值： 100
Filter	否	否	ProjectResourceFilter	搜索条件 示例值：查看

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数 示例值： 100
ResourceSet	Array of ProjectResource	资源列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败

查询项目列表

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询项目列表

默认接口请求频率限制：20次/秒。

接口更新时间：2023-07-24 14:42:18。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeProjects
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PageNumber	否	否	Uint64	页码 示例值： 100
PageSize	否	否	Uint64	每页数量 示例值： 100
Filter	否	否	DescribeProjectsFilter	过滤条件 示例值：查看
MemberUin	否	否	String	成员UIN 示例值： 100

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数 示例值： 100
ProjectSet	Array of Project	项目列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.DatabaseError	内部数据库错误

查询具有转入转出资源权限的项目列表

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询具有转入转出资源权限的项目列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeResourceAdminProjects
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数 示例值： 100
ProjectSet	Array of ResourceAdminProject	项目列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败

查询所有资源区域

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

查询所有资源区域

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeResourceRegions
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
RegionSet	Array of Region	地域列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode

修改单个用户的授权策略

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

修改单个用户的授权策略

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyProjectMemberPolicy
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
AccountUin	是	否	Uint64	用户UID 示例值： 100
PolicyNames	是	否	Array of String	待添加的策略名列表 示例值： ["policygen-20141112201913","policygen-20141112201914"]

3. 输出参数

参数名称	类型	描述
PolicyNames	Array of String	策略名称列表 示例值： ["policygen-20141112201913","policygen-20141112201914"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败

修改项目名称

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

修改项目名称

默认接口请求频率限制：20次/秒。

接口更新时间：2023-07-24 14:39:49。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyProjectName
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目id 示例值： pr-dcd34c11
ProjectName	是	否	String	项目名称 示例值： pr1
ProjectDescription	否	否	String	项目描述 示例值： prdesc

3. 输出参数

参数名称	类型	描述
ProjectId	String	项目id 示例值： pr-dcd34c11
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
ResourceNotFound.ProjectNotFoundError	项目不存在
InvalidParameter.ProjectNameTooLong	项目名过长
InternalError.ComponentFail	调用其它组件接口失败
InvalidParameter.EmptyParameter	有参数为空
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败

修改项目配额

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

修改项目配额

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyProjectQuota
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
ProductCode	是	否	String	一级产品定义 示例值： p_cvm
QuotaKey	是	否	String	配额键值 示例值： p_cvm###
QuotaValue	是	否	String	配额数 示例值： 100

3. 输出参数

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
InvalidParameter.EmptyParameter	有参数为空
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
InvalidParameter.InvalidParameter	参数不对
InvalidParameter.UsedQuotaNotEnough	新的配额要大于已使用的配额
UnauthorizedOperation.AuthFailedError	鉴权失败

项目间移动资源

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

无

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：MoveProjectResource
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OldProjectId	是	否	String	原项目 示例值： pr-dcd34c11
NewProjectId	是	否	String	新项目 示例值： pr-dcd34c11
ResourceList	是	否	Array of TransferResource	资源 示例值：查看

3. 输出参数

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败
InvalidParameter.UnsupportedProductCodeError	不支持的ProductCode
ResourceNotFound.ProjectResourceNotFound	项目内资源不存在
FailedOperation.ProjectCountError	资源的项目数过多
FailedOperation.ProjectResourceNotEmpty	项目内资源不为空

项目名是否唯一

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

项目名是否唯一

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-09 16:15:26。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ProjectNameExists
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectName	是	否	String	项目名 示例值： pr1

3. 输出参数

参数名称	类型	描述
Exist	Bool	是否存在 示例值： true
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.DatabaseError	内部数据库错误

从项目移除成员

1. 接口描述

接口请求域名： tpo.api3.finance.cloud.tencent.com。

从项目移除成员

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-18 17:30:37。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RemoveProjectMember
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
Uins	是	否	Array of Uint64	待移除的用户UID列表 示例值： [1000,1001]

3. 输出参数

参数名称	类型	描述
Uins	Array of Uint64	移除的用户uin列表 示例值： [1000,1001]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.DatabaseError	内部数据库错误
UnauthorizedOperation.AuthFailedError	鉴权失败

数据结构

AddQuota

添加的配额

被如下接口引用：BatchAddProjectQuota

名称	必选	允许NULL	类型	描述
ProductCode	是	否	String	一级产品编码 示例值： p_cvm
SubProductCode	否	否	String	二级产品编码 示例值： sp_cvm_sh1
BillingItemCode	否	否	String	三级产品编码 示例值： v_cvm_cpu
SubBillingItemCode	否	否	String	四级产品编码 示例值： sv_cbs_memspace_common
QuotaValue	是	否	Int64	配额值 示例值： 100
QuotaKey	否	否	String	配额键 示例值： p_cvm###

Project

项目

被如下接口引用：DescribeProjects

名称	必选	允许NULL	类型	描述
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
ProjectName	是	否	String	项目名称 示例值： pr1
Organization	是	否	String	组织 示例值： org1

名称	必选	允许NULL	类型	描述
Creator	是	否	String	创建者 示例值： admin
CreateTime	是	否	Datetime	创建时间 示例值： 2014-08-03 12:00:00
CreatorUin	是	否	Uint64	创建者Uin 示例值： 1000
OrgId	是	否	String	组织id 示例值： org-231
OrgName	是	否	String	组织名称 示例值： org1
OrgOperationTime	是	否	String	项目转入项目目录时间 示例值： 2014-08-03 12:00:00
OrgOperator	是	否	String	项目转入项目目录操作人 示例值： admin
ProjectDescription	否	否	String	项目描述 示例值： desc
Uin	否	否	String	用户ID 示例值： 100
AppId	否	否	Uint64	AppId 示例值： 100

ProjectResourceFilter

资源列表搜索条件

被如下接口引用：DescribeProjectResources

名称	必选	允许NULL	类型	描述
ResourceId	否	否	String	资源ID 示例值： ins-abzttwhl
RegionId	否	否	Uint64	地域 示例值： 5000001

名称	必选	允许NULL	类型	描述
ProductCode	否	否	String	产品Code 示例值： p_cvm
ResourceType	否	否	String	资源类型（暂不支持） 示例值： cvm
Keyword	否	否	String	资源ID或资源名称模糊查询 示例值： ins-dsad12
Product	否	否	String	产品Code 示例值： p_cvm
ServiceType	否	否	String	服务类型 示例值： cvm

DescribeProjectsFilter

查询项目列表搜索条件

被如下接口引用：DescribeProjects

名称	必选	允许NULL	类型	描述
Keyword	否	否	String	按项目ID或者项目名模糊搜索 示例值： ""
Type	否	否	Uint64	匹配类型 示例值： 1

ProjectQuota

项目配额

被如下接口引用：DescribeProjectQuotas

名称	必选	允许NULL	类型	描述
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
ProductName	是	否	String	产品名 示例值： cvm

名称	必选	允许NULL	类型	描述
ProductCode	是	否	String	产品Code 示例值： p_cvm
SubProductCode	是	是	String	二级产品 示例值： sp_cvm_sh1
BillingItemCode	是	是	String	三级产品 示例值： v_cvm_cpu
SubBillingItemCode	是	是	String	四级产品 示例值： sv_cbs_memspace_common
QuotaKey	是	否	String	配额键 示例值： p_cvm###
QuotaName	是	是	String	配额项名称 示例值： 云数据库Redis
QuotaValue	是	否	String	总配额 示例值： 100
QuotaLeft	是	否	Int64	剩余配额 示例值： 100
QuotaUsed	是	否	Int64	已使用配额 示例值： 100
CreateTime	是	否	String	创建时间 示例值： 2014-08-03 12:00:00
UpdateTime	是	否	String	更新时间 示例值： 2014-08-03 12:00:00
BillingItemName	是	否	String	产品项 示例值： 存储空间
SubBillingItemName	是	否	String	产品细项 示例值： SSD云硬盘-存储空间
SubProductName	是	否	String	子产品名 示例值： SSD云硬盘
Unit	是	否	String	单位 示例值： 1

DeleteQuota

删除Quota

被如下接口引用：DeleteProjectQuota

名称	必选	允许NULL	类型	描述
ProductCode	是	否	String	产品Code 示例值： p_cvm
QuotaKey	是	否	String	配额键 示例值： p_cvm###

ProjectQuotaFilter

配额查询条件

被如下接口引用：DescribeProjectQuotas

名称	必选	允许NULL	类型	描述
ProductCode	是	否	String	一级产品 示例值： p_cvm
SubProductCode	否	否	String	二级产品 示例值： sp_cvm_sh1
BillingItemCode	否	否	String	三级产品 示例值： v_cvm_cpu
SubBillingItemCode	否	否	String	四级产品 示例值： sv_cbs_memspace_common
Keyword	否	否	String	配额项名称模糊查询 示例值： cvm

ProjectResource

项目资源

被如下接口引用：DescribeProjectResources、DescribeResources

名称	必选	允许NULL	类型	描述
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11

名称	必选	允许NULL	类型	描述
ResourceId	是	否	String	资源ID 示例值： ins-abzttwhl
ResourceName	是	否	String	资源名称 示例值： cvm
ProductCode	是	否	String	产品Code 示例值： p_cvm
ProductName	是	否	String	产品名称 示例值： cvm
ProductGroupName	是	否	String	产品大类名称 示例值： cvm
RegionId	是	否	Int64	地域ID 示例值： 5000001
ProjectName	是	否	String	项目名称 示例值： pr1
RegionEnName	是	否	String	地域英文名 示例值： region1
RegionName	是	否	String	地域名称 示例值： region1
ResourceType	是	否	String	资源类型 示例值： cvm
ServiceType	是	否	String	产品对应的serviceType 示例值： cvm
Product	否	否	String	产品名称 示例值： cvm
Uin	否	否	Uint64	用户ID 示例值： 100

QuotaNames

产品名称

被如下接口引用：CheckProjectQuotas

名称	必选	允许NULL	类型	描述
ProductName	是	否	String	一级产品名 示例值： cvm
SubProductName	否	否	String	二级产品名 示例值： SSD云硬盘
BillingItemName	否	否	String	三级产品名 示例值： 存储空间
SubBillingItemName	否	否	String	四级产品名 示例值： SSD云硬盘-存储空间

ProjectMemberFilter

查询

被如下接口引用：DescribeProjectMembers、DescribeProjectNonMembers

名称	必选	允许NULL	类型	描述
Keyword	否	否	String	按用户ID或用户名模糊查询 示例值： ""

Products

产品

被如下接口引用：DescribeProducts

名称	必选	允许NULL	类型	描述
ProductName	是	否	String	Product名称 示例值： cvm
ResourceTypes	是	否	Array of Resource	资源类型 示例值：查看

TransferResource

资源转移

被如下接口引用：AddProjectResource、DeleteProjectResource、MoveProjectResource

名称	必选	允许NULL	类型	描述
ProductCode	是	否	String	产品code 示例值： p_cvm
RegionId	是	否	String	地域ID 示例值： 500001
ResourceId	是	否	String	资源ID 示例值： ins-abzttwhl
Uin	否	否	String	用户Uin 示例值： 1000
Region	否	否	String	地域 示例值： region1

UinPolicy

uin策略

被如下接口引用：AddProjectMemberPolicy

名称	必选	允许NULL	类型	描述
Uin	是	否	Uint64	账户ID 示例值： 1000
PolicyName	是	否	String	策略名 示例值： policygen-20141112201913
Detail	否	是	String	失败详情 示例值：

Resource

资源类型

被如下接口引用：DescribeProducts

名称	必选	允许NULL	类型	描述
ProductCode	是	否	String	产品Code

名称	必选	允许NULL	类型	描述
				示例值： p_cvm
ResourceType	是	否	String	资源类型 示例值： volume
ServiceType	是	否	String	服务类型，对应cam serivceType 示例值： cvm

QuotaChecking

配额校验结果

被如下接口引用：CheckProjectQuotas

名称	必选	允许NULL	类型	描述
BillingItemCode	是	否	String	三级产品 示例值： v_cvm_cpu
BillingItemName	是	否	String	产品项 示例值： 存储空间
Error	是	否	String	错误内容 示例值： operation error
Exists	是	否	Bool	是否已有配额，传true表示有配额，传false表示没有 示例值： true
ProductCode	是	否	String	产品code 示例值： p_cvm
ProductName	是	否	String	产品名称 示例值： cvm
QuotaKey	是	否	String	配额键 示例值： p_cvm###
QuotaLeft	是	否	String	剩余配额 示例值： 100
QuotaUsed	是	否	String	已使用配额 示例值： 100

名称	必选	允许NULL	类型	描述
QuotaValue	是	否	String	配额值 示例值： 100
SubBillingItemCode	是	否	String	四级产品 示例值： sv_cbs_memspace_common
SubBillingItemName	是	否	String	产品细项 示例值： SSD云硬盘-存储空间
SubProductCode	是	否	String	二级产品 示例值： sp_cvm_sh1
SubProductName	是	否	String	子产品名 示例值： SSD云硬盘
Success	是	否	Bool	是否合法，传1表示合法，传0表示不合法 示例值： true

PolicyFilter

策略

被如下接口引用：DescribeProjectMemberPolicies、DescribeProjectPolicies

名称	必选	允许NULL	类型	描述
Keyword	否	否	String	按策略名模糊查询 示例值： 1

ApplyProjectQuotaHistory

申请项目配额历史

被如下接口引用：DescribeApplyProjectQuotaHistory

名称	必选	允许NULL	类型	描述
PaperID	是	否	Uint64	审批单id 示例值： 1
ApplicantUin	是	否	String	申请人uin 示例值： 1000

名称	必选	允许NULL	类型	描述
Applicant	是	否	String	申请人 示例值： admin
CTime	是	否	String	申请时间 示例值： 2014-10-09T14:43:49+08:00
QuotaKey	是	否	String	配额key 示例值： p_cvm###
QuotaName	是	否	String	配额名称 示例值： 云服务器CVM
QuotaValue	是	否	Uint64	配额数 示例值： 1000
ApplyQuotaValue	是	否	Uint64	申请的配额数 示例值： 100
Status	是	否	Uint64	审批状态 示例值： 14

Policy

策略

被如下接口引用：DescribeProjectMemberPolicies、DescribeProjectMembers、DescribeProjectNonMembers、DescribeProjectPolicies

名称	必选	允许NULL	类型	描述
PolicyName	是	否	String	策略名 示例值： policygen-20141112201913
Description	是	否	String	策略描述 示例值： desc
PolicyId	否	否	Uint64	策略ID 示例值： 1000

Region

地域

被如下接口引用：DescribeResourceRegions

名称	必选	允许NULL	类型	描述
RegionId	是	否	Uint64	地域ID 示例值： 5000001
RegionName	是	否	String	地域名称 示例值： region1

ResourceAdminProject

项目

被如下接口引用：DescribeResourceAdminProjects

名称	必选	允许NULL	类型	描述
ProjectId	是	否	String	项目ID 示例值： pr-dcd34c11
ProjectName	是	否	String	项目名称 示例值： pro1
Organization	是	否	String	项目名 示例值： org1

ProjectMember

项目成员

被如下接口引用：DescribeProjectMembers、DescribeProjectNonMembers

名称	必选	允许NULL	类型	描述
Uin	是	否	Uint64	用户Uin 示例值： 1000
Uid	是	否	Uint64	用户Uid 示例值： 1000
Name	是	否	String	用户名 示例值： member1

名称	必选	允许NULL	类型	描述
Policies	是	是	Array of Policy	具有的项目策略 示例值：查看

Product

产品

被如下接口引用：DescribeProductTree

名称	必选	允许NULL	类型	描述
ProductCode	是	否	String	产品Code 示例值： p_cvm
ProductName	是	否	String	产品名 示例值： cvm
Unit	是	否	String	单位 示例值： 1

错误码

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 cor
rect."
    },
    "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
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接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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
InvalidParameter.InvalidProjectQuota	无效的配额
InvalidParameter.EmptyParameter	有参数为空
InternalError.ComponentFail	调用其它组件接口失败
InternalError.InvalidHTTPCode	无效httpcode

错误码	说明
InvalidParameter.UsedQuotaNotEnough	新的配额要大于已使用的配额
InvalidParameter.InvalidParameter	参数不对
InvalidParameter.UnsupportedProductCodeError	不支持的ProductCode
UnauthorizedOperation.AuthFailedError	鉴权失败
FailedOperation.ProjectQuotaNotEmpty	项目配额不为空
ResourceNotFound.ProjectNotFoundError	项目不存在
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.DatabaseError	内部数据库错误
FailedOperation.ProjectCountError	资源的项目数过多
InvalidParameter.ProjectNameTooLong	项目名过长
ResourceNotFound.ProjectResourceNotFound	项目内资源不存在
FailedOperation.ProjectResourceNotEmpty	项目内资源不为空

访问管理（cam）
版本（2019-01-16）
API 概览

API版本

V3

其他接口

接口名称	接口功能
AttachRolePolicies	绑定多个策略到角色
AttachRolePolicy	绑定权限策略到角色
AttachRolesPolicy	绑定多个角色到策略
CreatePolicy	创建策略
CreateRole	创建角色
DeletePolicy	删除策略
DeleteRole	删除角色
DescribeRoleList	获取角色列表
DetachGroupPolicies	解除绑定多个策略到用户组
DetachGroupsPolicy	解除绑定策略到多个用户组
DetachUsersPolicy	解除绑定策略到多个用户
GetPolicy	查看策略详情
GetRole	获取角色详情
GetServiceRoleInfo	获取服务角色信息
ListAttachedGroupPolicies	查询用户组关联的策略列表
ListAttachedRolePolicies	获取角色绑定的策略列表

接口名称	接口功能
ListEntitiesForPolicy	查询策略关联的实体列表
ListPolicies	查询策略列表
UpdateAssumeRolePolicy	修改角色信任策略
UpdatePolicy	更新策略

用户相关接口

接口名称	接口功能
GetPasswordRules	获取CAM密码规则
GetSubsGroup	子账户所属用户组列表
GetUinBySecretId	根据SecretId查询Uin
UpdatePasswordRules	更新CAM密码规则

身份提供商接口

接口名称	接口功能
CreateOauthProvider	新增oauth配置
GetUserAccessToken	获取用户oauth标识
RefreshUserToken	刷新用户userAccessToken
UpdateOauthProvider	更新Oauth配置信息
VerifyUserAccessToken	验证用户userAccessToken

调用方式
接口签名v1

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐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。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
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

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' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    '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=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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.finance.cloud.tencent.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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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 http 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TCloudFinanceZone 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
- Node

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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

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

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

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

Java

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

public class CloudAPIDemo {
    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.finance.cloud.tencent.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.finance.cloud.tencent.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)).app
end("&");
        }
        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.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 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", "shjr"); // 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
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        '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)
```

接口签名v3

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：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 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，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。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

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

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- 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，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- 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）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 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
```

- 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 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- 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，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

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

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

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

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

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

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 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）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //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 = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\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 = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        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("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

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

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

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

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

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

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

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

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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。

公共参数

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

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	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。
X-TC-Timestamp	Integer	是	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。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	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。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

其他接口
绑定多个策略到角色

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

绑定多个策略到角色

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AttachRolePolicies
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	Uint64	角色ID(与角色名称必传一项) 示例值： 4611686018427387904
RoleName	否	否	String	角色名称(与角色ID必传一项) 示例值： QCS_RoleName
PolicyId	否	否	Array of Uint64	策略ID list(与策略名 list必传一项) 示例值： [1001,1002]
PolicyName	否	否	Array of String	策略名 list(与策略ID list必传一项) 示例值： ["policygen-1","policygen-2"]

3. 输出参数

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

4. 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.PolicyFull	用户策略数超过上限。
InvalidParameter.OperatePoliciesOverLimit	一次操作策略数过多。
InvalidParameter.PasswordLengthTooShort	密码太短。
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdNotExist	策略ID不存在。

绑定权限策略到角色

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（AttachRolePolicy）用于绑定策略到角色。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AttachRolePolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyId	否	否	Uint64	策略ID，入参PolicyId与PolicyName二选一 示例值： 1
AttachRoleId	否	否	String	角色ID，用于指定角色，入 参 AttachRoleId 与 AttachRoleName 二选一 示例值： 4611686018427397905
AttachRoleName	否	否	String	角色名称，用于指定角色，入 参 AttachRoleId 与 AttachRoleName 二选一 示例值： QCS_RoleName
PolicyName	否	否	String	策略名，入参PolicyId与PolicyName二选一 示例值： policygen-20141112201913

3. 输出参数

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

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.PolicyIdNotExist	策略ID不存在。
InternalError.SystemError	内部错误。
InvalidParameter.RoleNotExist	角色不存在。
InvalidParameter.AttachmentFull	principal字段的授权对象关联策略数已达到上限。

绑定多个角色到策略

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

绑定多个角色到策略

默认接口请求频率限制：20次/秒。

接口更新时间：2023-02-09 15:36:51。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AttachRolesPolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	Array of Uint64	角色ID(与角色名称必传一项) 示例值： [4611686018427387904,4611686018427387905]
RoleName	否	否	Array of String	角色名称(与角色ID必传一项) 示例值： ["role-1","role-2"]
PolicyId	否	否	Uint64	策略ID(与策略名必传一项) 示例值： 1000
PolicyName	否	否	String	策略名(与策略ID必传一项) 示例值： policygen-20141112201913

3. 输出参数

参数名称	类型	描述

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

4. 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.PolicyFull	用户策略数超过上限。
InvalidParameter.OperatePoliciesOverLimit	一次操作策略数过多。
InvalidParameter.PasswordLengthTooShort	密码太短。
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdNotExist	策略ID不存在。

创建策略

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（CreatePolicy）可用于创建策略。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-02 16:17:55。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreatePolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyName	是	否	String	策略名 示例值： policygen-20141112201913
Description	否	否	String	策略描述 示例值： 测试描述信息
PolicyDocument	是	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.<no value>","cls.<no value>"]}}]}，principal用于指定角色的授权对象。获取该参数可参阅 获取角色详情（https://<no value>/document/product/598/36221）输出参数RoleInfo 示例值： {"version":"2.0","statement":[{"effect":"allow","action":"cvm:Describe*","resource":"*"}]}
CreateMode	否	否	Uint64	创建模式，1 dbsql带出来的预设策略, 2 按策略模版创建 角色策略 COS侧后台接口创建, 3 按照策略生成器创建, 4 标签相关 示例值： 1

3. 输出参数

参数名称	类型	描述
PolicyId	Uint64	策略id 示例值： 1000
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
ResourceNotFound.UserNotExist	用户不存在。
FailedOperation.PolicyFull	用户策略数超过上限。
InternalError.SystemError	内部错误。
InvalidParameter.ActionNotExist	action不存在

创建角色

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（CreateRole）用于创建角色。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateRole
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleName	是	否	String	角色名称 示例值： QCS
PolicyDocument	是	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.<no value>","cls.<no value>"]}}]}，principal用于指定角色的授权对象。获取该参数可参阅 获取角色详情（https://<no value>/document/product/598/36221）输出参数RoleInfo 示例值： {"version":"2.0","statement":[{"action":"sts:AssumeRole","effect":"allow","principal":{"qcs":["qcs::cam::uin/110000000007:root"]}}]}
Description	否	否	String	角色描述 示例值： adesc
ConsoleLogin	否	否	Uint64	是否允许登录 1 为允许 0 为不允许 示例值： 0

参数名称	必选	允许NULL	类型	描述
SessionDuration	否	否	Uint64	申请角色临时密钥的最长有效期限制(范围：0~43200) 示例值： 3600
RoleType	否	否	String	角色类型(system

3. 输出参数

参数名称	类型	描述
RoleId	String	角色ID 示例值： 4611686018427388476
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.AttachmentFull	principal字段的授权对象关联策略数已达到上限。
InvalidParameter.ConditionError	策略文档的condition字段不合法。
InvalidParameter.DescriptionLengthOverlimit	Description入参长度不能大于300字节。
InvalidParameter.PrincipalError	策略文档的principal字段不合法。
InvalidParameter.RoleFull	角色数量达到上限。
InvalidParameter.RoleNameError	角色名不合法。
InvalidParameter.RoleNameInUse	相同名称的角色已存在。
InvalidParameter.UserNotExist	用户对象不存在。
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。

删除策略

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

删除策略

默认接口请求频率限制：20次/秒。

接口更新时间：2024-11-11 16:48:23。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeletePolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyId	否	否	Array of Uint64	数组，数组成员是策略 id，支持批量删除策略 示例值： [1000,1001]

3. 输出参数

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

4. 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

错误码	描述
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
ResourceNotFound.NotFound	资源不存在。
ResourceNotFound.PolicyIdNotFound	PolicyId指定的资源不存在。

删除角色

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（DeleteRole）用于删除指定角色。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteRole
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	String	角色ID，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： 4611686018427844696
RoleName	否	否	String	角色名称，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： QCS_RoleName

3. 输出参数

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

4. 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
InternalError.SystemError	内部错误。
InvalidParameter.RoleNotExist	角色不存在。
InvalidParameter.ParamError	非法入参。

获取角色列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（DescribeRoleList）用于获取账号下的角色列表。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRoleList
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Page	是	否	Uint64	页码，从1开始 示例值： 1
Rp	是	否	Uint64	每页行数，不能大于200 示例值： 5
Service	否	否	String	按角色的服务账号载体过滤 示例值： cvm
Keyword	否	否	Array of String	按角色名或角色描述过滤 示例值： ["rolename","roleid"]

3. 输出参数

参数名称	类型	描述
List	Array of	角色详情列表。

参数名称	类型	描述
	RoleInfo	示例值：查看
TotalNum	Uint64	角色总数 示例值： 14
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。

解除绑定多个策略到用户组

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

解除绑定多个策略到用户组

默认接口请求频率限制：20次/秒。

接口更新时间：2024-11-11 16:49:40。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DetachGroupPolicies
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
GroupId	是	否	Uint64	用户组ID 示例值： 1000
PolicyId	是	否	Array of Uint64	策略ID list 示例值： [1000,1001]

3. 输出参数

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

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
InternalError.SystemError	内部错误。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。

解除绑定策略到多个用户组

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

解除绑定策略到多个用户组

默认接口请求频率限制：20次/秒。

接口更新时间：2024-11-11 16:50:02。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DetachGroupsPolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
GroupId	是	否	Array of Uint64	用户组ID list 示例值： [1000,1001]
PolicyId	是	否	Uint64	策略ID 示例值： 1000

3. 输出参数

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

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
InternalError.SystemError	内部错误。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。

解除绑定策略到多个用户

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

解除绑定策略到多个用户

默认接口请求频率限制：20次/秒。

接口更新时间：2024-11-11 16:51:47。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DetachUsersPolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
TargetUin	是	否	Array of Uint64	目标用户ID list 示例值： [1000,1001]
PolicyId	是	否	Uint64	策略ID 示例值： 1000

3. 输出参数

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

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
FailedOperation.PolicyFull	用户策略数超过上限。
InternalError.SystemError	内部错误。

查看策略详情

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（GetPolicy）可用于查询查看策略详情。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetPolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyId	是	否	Uint64	策略Id 示例值： 17698703

3. 输出参数

参数名称	类型	描述
PolicyName	String	策略名 示例值： policygen-20141112201913
Description	String	策略描述 示例值： 测试策略
Type	Uint64	1 表示自定义策略，2 表示预设策略 示例值： 1

参数名称	类型	描述
AddTime	Datetime	创建时间 示例值： 2019-04-29 21:18:28
UpdateTime	Datetime	最近更新时间 示例值： 2019-04-29 21:28:32
PolicyDocument	String	策略文档 示例值： {"version":"2.0","statement":[{"effect":"allow","action":["name\/cos:*"],"resource":["*"]}]}
PresetAlias	String	备注 示例值： 备注
IsServiceLinkedRolePolicy	Uint64	是否服务相关策略 示例值： 1
CreateMode	Uint64	1 dbsql带出来的预设策略, 2 按策略模版创建 角色策略 COS侧后台接口创建, 3 按照策略生成器创建, 4 标签相关 示例值： 1
IsCheck	Uint64	秘钥是否验证 1、已验证 0未验证 示例值： 1
PolicyId	Uint64	策略 ID 示例值： 1000
ServiceType	String	服务名称 示例值： cvm
Attachments	Uint64	关联的用户数 示例值： 1
IsAttached	Uint64	当需要查询标记实体是否已经关联策略时不为null。0表示未关联策略，1表示已关联策略 示例值： 1
Deactived	Uint64	是否已下线，传1代表已下线，传0代表未下线 示例值： 0
DeactivedDetail	Array of String	已下线产品列表 示例值： ["deacproduct"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
ResourceNotFound.PolicyIdNotFound	PolicyId指定的资源不存在。
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。

获取角色详情

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（GetRole）用于获取指定角色的详细信息。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetRole
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	String	角色 ID，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： 4611686018427844696
RoleName	否	否	String	角色名，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： QCSRole

3. 输出参数

参数名称	类型	描述
RoleInfo	RoleInfo	角色详情 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。
InvalidParameter.RoleNotExist	角色不存在。

获取服务角色信息

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

获取服务角色信息

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetServiceRoleInfo
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleName	否	否	String	角色名 示例值： QCS
PolicyName	否	否	Array of String	策略名列表 示例值： ["policygen-1","policygen-2"]

3. 输出参数

参数名称	类型	描述
RoleName	String	角色名 示例值： QCS
ServiceType	String	业务类型 示例值： cvm

参数名称	类型	描述
ServiceTypeEn	String	英文业务类型 示例值： cvm
RoleDesc	String	角色描述 示例值： QCS
RoleDescEn	String	角色英文描述 示例值： QCS
PolicyName	String	预设策略名 示例值： policygen-20141112201913
Remark	String	描述 示例值： remark
EnRemark	String	英文描述 示例值： ""
PolicyList	Array of RolePolicyList	策略列表 示例值：查看
RoleDescI18n	String	支持国际化的角色描述 示例值： ""
Id	String	角色Id 示例值： ""
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。

查询用户组关联的策略列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

查询用户组关联的策略列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ListAttachedGroupPolicies
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
TargetGroupId	是	否	Uint64	用户组ID 示例值： 3349
Page	是	否	Uint64	页码，默认值是 1，从 1 开始 示例值： 1
Rp	是	否	Uint64	每页大小，默认值是 20 示例值： 10

3. 输出参数

参数名称	类型	描述
TotalNum	Uint64	策略总数 示例值： 1

参数名称	类型	描述
List	Array of AttachPolicyInfo	策略列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。

获取角色绑定的策略列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（ListAttachedRolePolicies）用于获取角色绑定的策略列表。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ListAttachedRolePolicies
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	String	角色 ID。用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： 4611686018427397905
RoleName	否	否	String	角色名。用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： role_data
PolicyType	否	否	String	按策略类型过滤，User表示仅查询自定义策略，QCS表示仅查询预设策略 示例值： 1
Page	是	否	Uint64	页码，从 1 开始 示例值： 1
Rp	是	否	Uint64	每页行数，不能大于200 示例值： 10

3. 输出参数

参数名称	类型	描述
List	Array of AttachedPolicyOfRole	角色关联的策略列表 示例值：查看
TotalNum	Uint64	角色关联的策略总数 示例值： 1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	非法入参。

查询策略关联的实体列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（ListEntitiesForPolicy）可用于查询策略关联的实体列表。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-02 21:20:38。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ListEntitiesForPolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyId	是	否	Uint64	策略 id 示例值： 524497
Page	否	否	Uint64	页码，默认值是 1，从 1 开始 示例值： 1
Rp	否	否	Uint64	每页大小，默认值是 20 示例值： 10
EntityFilter	否	否	String	可取值 'All'、'User'、'Group' 和 'Role'，'All' 表示获取所有实体类型，'User' 表示只获取子账号，'Group' 表示只获取用户组，'Role' 表示只获取角色，默认取 'All' 示例值： All

3. 输出参数

参数名称	类型	描述
TotalNum	Uint64	实体总数 示例值： 1
List	Array of AttachEntityOfPolicy	实体列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。

查询策略列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（ListPolicies）可用于查询策略列表。

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-09 17:18:22。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ListPolicies
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Rp	否	否	Uint64	每页数量，默认值是 20，必须大于 0 且小于或等于 200 示例值： 1
Page	否	否	Uint64	页码，默认值是 1，从 1开始，不能大于 200 示例值： 10
Scope	否	否	String	可取值 'All'、'QCS' 和 'Local'，'All' 获取所有策略，'QCS' 只获取预设策略，'Local' 只获取自定义策略，默认取 'All' 示例值： All
Keyword	否	否	String	按策略名匹配 示例值： name
TargetUin	否	否	Uint64	按Uin匹配 示例值： 10093
TargetGroupId	否	否	Uint64	按组Id匹配 示例值： 10093

参数名称	必选	允许NULL	类型	描述
TargetRoleId	否	否	Uint64	按角色Id匹配 示例值： 10093
ServiceType	否	否	String	按产品Id匹配，如cvm 示例值： cvm
FlagUin	否	否	Uint64	按Uin标记关联 示例值： 10093
FlagGroupId	否	否	Uint64	按GroupId标记关联 示例值： 10093
FlagRoleId	否	否	Uint64	按角色Id标记关联 示例值： 10093
Order	否	否	String	策略排序参数 示例值： desc
ProjectVisible	否	否	Uint64	项目可见性 示例值： 1
Client	否	否	String	客户端 示例值： console

3. 输出参数

参数名称	类型	描述
TotalNum	Uint64	策略总数 示例值： 239
List	Array of StrategyInfo	策略数组，数组每个成员包括 policyId、policyName、addTime、type、description、 createMode 字段。其中： policyId：策略 id policyName：策略名 addTime：策略创建时间 type：1 表示自定义策略，2 表示预设策略 description：策略描述 createMode：1 表示按业务权限创建的策略，其他值表示可以查看策略语法和通过策略语法更新策略 Attachments: 关联的用户数 ServiceType: 策略关联的产品 IsAttached: 当需要查询标记实体是否已经关联策略时不为null。0表示未关联策略，1表示已关联策略

参数名称	类型	描述
		示例值：查看
ServiceTypeList	Array of String	服务列表 示例值： ["cvm"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InvalidParameter.UinError	Uin字段不合法。
InternalError.SystemError	内部错误。
InvalidParameter.ScopeError	Scope字段不合法。
InvalidParameter.KeywordError	Keyword字段不合法。
InvalidParameter.GroupIdError	GroupId字段不合法。
InvalidParameter.ServiceTypeError	ServiceType字段不合法。

修改角色信任策略

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（UpdateAssumeRolePolicy）用于修改角色信任策略的策略文档。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateAssumeRolePolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleId	否	否	String	角色ID，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： 4611686018427731422
RoleName	否	否	String	角色名称，用于指定角色，入参 RoleId 与 RoleName 二选一 示例值： QCS_RoleName
PolicyDocument	是	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.<no value>","cls.<no value>"]}}]}，principal用于指定角色的授权对象。获取该参数可参阅 获取角色详情（https://<no value>/document/product/598/36221）输出参数RoleInfo 示例值： {"version":"2.0","statement":[{"action":"sts:AssumeRole","effect":"allow","principal":{"qcs":["qcs::cam::uin/110000000007:root"]}}]}

3. 输出参数

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

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ConditionError	策略文档的condition字段不合法。
InvalidParameter.VersionError	策略文档的Version字段不合法。
InternalError.SystemError	内部错误。
InvalidParameter.AttachmentFull	principal字段的授权对象关联策略数已达到上限。
InvalidParameter.PrincipalError	策略文档的principal字段不合法。
InvalidParameter.RoleNotExist	角色不存在。
InvalidParameter.ParamError	非法入参。

更新策略

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口（UpdatePolicy ）可用于更新策略。
如果已存在策略版本，本接口会直接更新策略的默认版本，不会创建新版本，如果不存在任何策略版本，则直接创建一个默认版本。

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:02:25。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdatePolicy
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PolicyId	是	否	Uint64	策略ID 示例值： 17698703
PolicyName	否	否	String	策略名 示例值： QCS_PolicyName
Description	否	否	String	策略描述 示例值： Policy_Description
PolicyDocument	否	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.<no value>","cls.<no value>"]}}]}，principal用于指定角色的授权对象。获取该参数可参阅 获取角色详情（https://<no value>/document/product/598/36221）输出参数RoleInfo 示例值： {"version":"2.0","statement":[{"effect":"allow","action":"cvm:Describe*","resource":"*"}]}

3. 输出参数

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

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.PolicyNameError	PolicyName字段不合法。
InvalidParameter.PrincipalError	策略文档的principal字段不合法。
InvalidParameter.ConditionError	策略文档的condition字段不合法。
InvalidParameter.ParamError	非法入参。
InvalidParameter.PolicyDocumentError	PolicyDocument字段不合法。
InvalidParameter.ActionError	策略文档的Action字段不合法。
ResourceNotFound.PolicyIdNotFound	PolicyId指定的资源不存在。
InvalidParameter.NotSupportProduct	CAM不支持策略文档中所指定的资源类型。
InvalidParameter.PolicyDocumentLengthOverLimit	PolicyDocument字段超过长度限制。
ResourceNotFound.UserNotExist	用户不存在。
InternalError.SystemError	内部错误。
InvalidParameter.StatementError	策略文档的Statement字段不合法。
InvalidParameter.VersionError	策略文档的Version字段不合法。
InvalidParameter.EffectError	策略文档的Effect字段不合法。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。
InvalidParameter.ResourceError	策略文档的Resource字段不合法。
InvalidParameter.UserNotExist	用户对象不存在。
ResourceNotFound.GroupNotExist	用户组不存在。
ResourceNotFound.NotFound	资源不存在。

错误码	描述
FailedOperation.PolicyNameInUse	PolicyName字段指定的策略名已存在。
InvalidParameter.AttachmentFull	principal字段的授权对象关联策略数已达到上限。
UnauthorizedOperation	未授权操作
InvalidParameter.DescriptionLengthOverlimit	Description入参长度不能大于300字节。

用户相关接口
获取CAM密码规则

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

本接口{GetPasswordRules}用于获取用户的密码设置规则

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-23 16:09:45。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetPasswordRules
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
Rules	PasswordRules	密码规则列表 示例值：查看
UpdateTime	String	更新时间 示例值： 2019-04-29 21:18:28
Modifier	String	更新用户 示例值： 130000000001

参数名称	类型	描述
BlackList	String	黑名单字符串列表json string 示例值： aaa
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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。

子账户所属用户组列表

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

查询子账户所属用户组列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-04-23 15:09:13。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetSubsGroup
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Uid	是	否	Uint64	接收者用户id 示例值： 1258042
Rp	是	否	Int64	单页数量 示例值： 20
Page	是	否	Int64	分页数 示例值： 1

3. 输出参数

参数名称	类型	描述
TotalNum	String	总体数量 示例值： 1

参数名称	类型	描述
GroupInfo	Array of GroupInfo	用户信息列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.SystemError	内部错误。

根据SecretId查询Uin

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

根据SecretId查询Uin

默认接口请求频率限制：20次/秒。

接口更新时间：2025-03-17 18:04:03。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetUinBySecretId
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ApiSecretId	是	否	String	密钥ID 示例值： AK****

3. 输出参数

参数名称	类型	描述
Uin	Uint64	用户ID 示例值： 100
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。

更新CAM密码规则

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

更新用户密码设置规则

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:06:02。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdatePasswordRules
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Rules	是	否	PasswordRules	密码设置规则 示例值：查看
BlackList	否	否	String	黑名单字符串列表json string 示例值： aaa

3. 输出参数

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

4. 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。

身份提供商接口
新增oauth配置

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

新增oauth配置

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-12 19:35:47。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：CreateOauthProvider
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Name	是	否	String	身份提供商（企业）名称。示例值: oauthserver 示例值： name
Desc	否	否	String	备注。示例值: mark 示例值： desc
ClientId	是	否	String	注册应用的id。示例值: c1aa4fbce389206a7061 示例值： c1aa4fbce389206a7061
ClientSecret	是	否	String	注册应用的密钥。示例值: 837c669d84f2fe64b11a8853d521a17970df55ca 示例值：944218ad060c8f2483d4d5bbfa64b6256b78aa79
AuthorizeUri	是	否	String	oauth验证授权信息url。示例值: http://test.com/oauth/authorize 示例值： https://github.com/login/oauth/authorize

参数名称	必选	允许NULL	类型	描述
AccessTokenUri	是	否	String	获取access_token url。示例值: http://test.com/oauth/token 示例值： https://github.com/login/oauth/access_token
GetUserInfoUri	是	否	String	获取用户信息url。示例值: http://test.com/oauth/userinfo 示例值： https://api.github.com/user
UserNameField	是	否	String	登录账号对应字段名称。示例值: username 示例值： user
NickNameField	否	否	String	昵称对应字段名称。示例值: nickname 示例值： nick
PhoneNumField	是	否	String	手机号对应字段名称。示例值: phone 示例值： phone
EmailField	是	否	String	邮箱对应字段名称。示例值: email 示例值： email
IsSyncIdpUser	否	否	Int64	是否同步 idp 用户。传0代表不同步, 传1代表同步。示例值: 0 示例值： 0

3. 输出参数

参数名称	类型	描述
Id	Int64	id 示例值： 1
Name	String	名称 示例值： name
SAMLProviderArn	String	SAMLProviderArn 示例值： ""
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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.IdentityExist	身份认证失败。

获取用户oauth标识

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

获取用户第三方开放平台的access token

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:05:48。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetUserAccessToken
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
UserAuthCode	是	否	String	auth code授权码 示例值： ""
OpenAccessToken	否	否	String	第三方access token，复杂授权使用。 示例值： d1e459383977a892fa7b6c549d2f76f8

3. 输出参数

参数名称	类型	描述
AppId	String	app id 示例值： 1255000002
UserOpenId	String	第三方openId 示例值： e5fcfd7cf67d62c85d38b10171641c57

参数名称	类型	描述
UserUnionId	String	第三方unionId 示例值： a64e5abe67a8712c348c547a2760f9ce
UserAccessToken	String	第三方access token 示例值： 905dd67ae2a97e9870364a87d7872da2
ExpiresAt	Int64	过期时间 示例值： 1730433390
UserRefreshToken	String	refresh token 示例值： 905dd67ae2a97e9870364a87d7872da2
Scope	String	授权范围 示例值： login
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

以下仅列出了接口业务逻辑相关的错误码，其他错误码详见公共错误码。

错误码	描述
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。
FailedOperation.AuthCodeError	授权码异常。

刷新用户userAccessToken

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

刷新用户第三方access_token

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:05:48。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：RefreshUserToken
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
UserRefreshToken	是	否	String	用户刷新token 示例值： d1e459383977a892fa7b6c549d2f76f8
UserOpenId	是	否	String	用户openId 示例值： e5fcfd7cf67d62c85d38b10171641c57

3. 输出参数

参数名称	类型	描述
UserAccessToken	String	第三方access_token 示例值： d1e459383977a892fa7b6c549d2f76f8
ExpiresAt	Int64	过期时间 示例值： 1730433390

参数名称	类型	描述
AppId	String	appId 示例值： 1255000002
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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.RefreshTokenError	刷新用户token异常。
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。

更新Oauth配置信息

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

更新Oauth配置信息

默认接口请求频率限制：20次/秒。

接口更新时间：2022-11-16 17:05:50。

接口既验签名又鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：UpdateOauthProvider
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Desc	否	否	String	备注 示例值： desc
Name	是	否	String	身份提供商（企业）名称 示例值： name
Id	是	否	Int64	id 示例值： 1
OwnerUin	是	否	Int64	OwnerUin 示例值： 110000000001
ClientId	是	否	String	注册应用的id 示例值： c1aa4fbce389206a7061
ClientSecret	是	否	String	注册应用的密钥 示例值： 944218ad060c8f2483d4d5bbfa64b6256b78aa79

参数名称	必选	允许NULL	类型	描述
AuthorizeUri	是	否	String	oauth验证授权信息url 示例值： https://github.com/login/oauth/authorize
AccessTokenUri	是	否	String	获取access_token url 示例值： https://github.com/login/oauth/access_token
GetUserInfoUri	是	否	String	获取用户信息url 示例值： https://api.github.com/user
UserNameField	是	否	String	登录账号对应字段名称 示例值： user
NickNameField	否	否	String	昵称对应字段名称 示例值： nick
PhoneNumField	是	否	String	手机号对应字段名称 示例值： phone
EmailField	是	否	String	邮箱对应字段名称 示例值： email
IsSyncIdpUser	是	否	Int64	是否同步 idp 用户数据 示例值： 0

3. 输出参数

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

4. 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。

验证用户userAccessToken

1. 接口描述

接口请求域名： cam.api3.finance.cloud.tencent.com。

验证用户第三方开放平台access_token

默认接口请求频率限制：20次/秒。

接口更新时间：2022-08-11 19:05:48。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：VerifyUserAccessToken
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
UserAccessToken	是	否	String	access token 示例值： d1e459383977a892fa7b6c549d2f76f8
UserOpenId	是	否	String	open id 示例值： e5fcfd7cf67d62c85d38b10171641c57

3. 输出参数

参数名称	类型	描述
UserOpenId	String	第三方平台openId 示例值： e5fcfd7cf67d62c85d38b10171641c57
UserUnionId	String	第三方平台unionId 示例值： d1e459383977a892fa7b6c549d2f76f8
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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.UserrAccessTokenError	用户接入token异常。
InvalidParameter.ParamError	非法入参。
InternalError.SystemError	内部错误。

数据结构

AttachedUserPolicy

用户关联的策略详情

被如下接口引用：ListAttachedUserAllPolicies

名称	必选	允许NULL	类型	描述
PolicyId	否	否	Uint64	策略ID 示例值： 1000
PolicyName	否	否	String	策略名 示例值： policygen-20141112201913
Description	否	否	String	策略描述 示例值： adesc
AddTime	否	否	String	创建时间 示例值： 2014-08-03 12:00:00
StrategyType	否	否	Uint64	策略类型(1表示自定义策略，2表示预设策略) 示例值： 1
CreateMode	否	否	Uint64	创建模式(1表示按产品或项目权限创建的策略，其他表示策略语法创建的策略) 示例值： 1
Groups	否	是	Array of AttachedUserPolicyGroupInfo	随组关联信息 示例值：查看

ServiceItem

服务

被如下接口引用：GetServiceList

名称	必选	允许NULL	类型	描述
AddTime	否	否	Datetime	创建时间

名称	必选	允许NULL	类型	描述
				示例值： 2014-08-03 12:00:00
ArnDocument	否	否	String	ArnDocument 示例值： https://www.example.com/document/product/378/8222965
ColConf	否	是	String	ColConf 示例值： ColConf
DefAddr	否	是	String	DefAddr 示例值： DefAddr
DefaultStrategyList	否	否	String	默认策略 示例值： DefaultStrategyList
IsAllowDefProj	否	否	String	IsAllowDefProj 示例值： 1
IsDisProject	否	否	String	IsDisProject 示例值： 1
IsDisZone	否	否	String	IsDisZone 示例值： 1
IsSeen	否	否	String	是否可见 示例值： 1
Online	否	否	String	Online 示例值： Online
QueryAddr	否	否	String	QueryAddr 示例值： cvm
QueryInterface	否	否	String	QueryInterface 示例值： cvm
ServiceEnName	否	否	String	服务英文名 示例值： cvm
ServiceName	否	否	String	服务名 示例值： cvm
ServiceType	否	否	String	服务类型 示例值： cvm
SynInterface	否	否	String	SynInterface 示例值： cvm

名称	必选	允许NULL	类型	描述
UpdateTime	否	否	Datetime	变更时间 示例值： 2014-08-03 12:00:00
Weight	否	否	String	Weight 示例值： 1
WhiteKey	否	否	String	WhiteKey 示例值： white1
Writter	否	否	String	创建人 示例值： admin
ResourceTypeList	否	否	Array of ResourceTypeItem	资源类型数组 示例值：查看
Type	否	否	String	类型 示例值： cvm

UserList

子账号列表

被如下接口引用：ListUsersForPolicy

名称	必选	允许NULL	类型	描述
Name	是	否	String	子账号名称 示例值： admin
SubAccountUin	是	否	String	子账号uin 示例值： 1000

OwnerAccountAttribute

主账户属性

被如下接口引用：UpdateOwnerAccount

名称	必选	允许NULL	类型	描述
Remark	否	否	String	属性 示例值： remark

PasswordRules

密码规则

被如下接口引用：GetPasswordRules、UpdatePasswordRules

名称	必选	允许NULL	类型	描述
MinimumLength	是	否	Int64	最小密码长度 示例值： 10
MustContain	是	否	String	最少包含 示例值： Aa!
ForcePasswordChange	是	否	Int64	密码有效期 示例值： 0
ReusePasswordLimit	是	否	Int64	密码重复次数 示例值： 2
RetryPasswordLimit	否	否	Int64	登陆最大密码失败次数 示例值： 2
OnlyAdminCanResetPassword	否	否	Int64	是否只有admin可以重置密码 示例值： 0
MustNotContainUsername	否	否	Int64	必须不包含用户名 示例值： 0

SubAccountFilter

带过滤条件的子帐号信息

被如下接口引用：GetGroupList、GetGroupsSubAccount、GetSubsGroup、ListGroups、ListGroupsForConsole、ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
Uid	是	是	Uint64	子用户Uid 示例值： 1258042
Uin	是	是	Uint64	用户Uin 示例值： 130000000001

名称	必选	允许NULL	类型	描述
Name	是	是	String	用户Name 示例值： username
Remark	是	是	String	备注 示例值： remark
CanLogin	是	是	Uint64	是都允许登录 示例值： 1
PhoneNum	是	是	String	电话号码 示例值： 11111111111
CountryCode	是	是	String	区号 示例值： 86
PhoneFlag	是	是	Int64	电话号码是否验证 示例值： 1
Email	是	是	String	邮箱 示例值： mail@mail.com
EmailFlag	是	是	Int64	邮箱是否验证 示例值： 1
UserType	是	是	Int64	用户类型 示例值： 0
CreateTime	是	是	String	创建时间 示例值： 2019-04-29 21:18:28
IsReceiverOwner	是	是	Int64	是否消息接收人 示例值： 1
SystemType	是	是	String	类型 示例值： subaccount
NeedResetPassword	是	是	Int64	是否需要重置密码 示例值： 0
ConsoleLogin	是	是	Int64	是否允许控制台登录 示例值： 1
WxzsStatus	是	是	Int64	微信公众号关注状态 示例值： 0

名称	必选	允许NULL	类型	描述
PermType	是	是	Array of String	权限类型 示例值： ["0"]
NickName	是	是	String	昵称 示例值： nick
QywxUserId	否	否	String	企业微信用户id 示例值： qywxuserid
UserAttributeAndValues	否	是	Array of AccountAttributeAndValue	扩展属性 示例值：查看
Status	否	否	Int64	状态 示例值： 0
LoginStatus	否	否	Int64	登陆状态 示例值： 0
VerifyPhone	否	否	String	VerifyPhone 示例值： 11111111111
VerifyEmail	否	否	String	VerifyEmail 示例值： mail@mail.com
VerifyCountryCode	否	否	String	VerifyCountryCode 示例值： 86

SubAccountInfo

子账户用户信息

被如下接口引用：UpdateSubAccount

名称	必选	允许NULL	类型	描述
CanLogin	否	否	String	能否登陆，0-否，1-可 示例值： 1
ConsoleLogin	否	否	String	是否是控制台登陆，1-是 示例值： 1
CountryCode	否	否	String	国家编码 示例值： 86

名称	必选	允许NULL	类型	描述
Name	否	否	String	用户名 示例值： 名称
NeedResetPassword	否	否	String	是否需要重置密码，1-是 示例值： 0
PhoneNum	否	否	String	手机号 示例值： 11111111111
Remark	否	否	String	备注 示例值： 备注
SystemType	否	否	String	账户类型 示例值： subaccount
Uid	否	否	String	接收者用户ID 示例值： 1280504
Uin	否	否	String	账户唯一id 示例值： 110000000001
Password	否	否	String	密码 示例值： password
WxzsStatus	否	否	Int64	微信消息状态 示例值： 0
UserType	否	否	Int64	用户类型 示例值： 0
Email	否	否	String	联系邮箱 示例值： mail@mail.com
Account	否	否	String	用户名 示例值： username
Lang	否	否	String	语言 示例值： en-US
NickName	否	否	String	昵称 示例值： 昵称

GroupData

用户组相关信息

被如下接口引用：GetAllSubUser

名称	必选	允许NULL	类型	描述
GroupId	否	否	Int64	用户组id 示例值： 1000
GroupName	是	否	String	用户组名称 示例值： group1
GroupNum	是	否	Int64	用户组成员数量 示例值： 100
Channel	是	否	Int64	创建渠道 示例值： 0
GroupMem	否	否	Array of Uint64	组成员uid 示例值： [1280504,1280505]

ApiKey

API密钥数据列表

被如下接口引用：CreateApiKey、CreateCollApiKey、QueryApiKey、QueryCollApiKey、QueryKeyBySecretId

名称	必选	允许NULL	类型	描述
SecretId	否	否	String	密钥ID 示例值： AKID***wdew
CreateTime	否	否	Uint64	创建时间(时间戳) 示例值： 1615990212
Status	否	否	Uint64	状态(2:有效, 3:禁用) 示例值： 2
SecretKey	否	否	String	密钥Key 示例值： ACCF***wdew
Source	否	否	Uint64	来源 示例值： 0
Remark	否	否	String	备注 示例值： remark

GroupMeta

用户组元信息

被如下接口引用：GetUserGroupList、ListAllUserGroup

名称	必选	允许NULL	类型	描述
GroupId	否	是	Int64	用户组id 示例值： 100
GroupName	否	是	String	用户组名称 示例值： group1

OwnerInfo

主账号信息

被如下接口引用：ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
Uin	是	是	Uint64	主帐号Uin 示例值： 130000000001
UserName	是	是	String	用户名 示例值： username
CheckStatus	是	否	Uint64	校验状态 示例值： 0

AccountDetail

账号详情

被如下接口引用：AddSubAccount、ListGroups

名称	必选	允许NULL	类型	描述
ActionFlag	否	否	ActionLoginFlag	敏感操作标识 示例值：查看
ConsoleLogin	否	否	String	是否允许控制台登录, 传0不可登陆控制台, 传1可以登陆控制台。示例值: 1

名称	必选	允许NULL	类型	描述
				示例值： 1
LoginFlag	否	否	ActionLoginFlag	登录保护 示例值：查看
NeedResetPassword	否	否	String	是否需要重置密码, 传0不需要重置密码, 传1需要重置密码。示例值: 0 示例值： 0
Password	否	否	String	用户密码。示例值: password 示例值： password
UseApi	否	否	String	使用Api, 传0不使用Api, 传1使用Api。 示例值: 1 示例值： 1
TokenType	否	否	Int64	分配设备类型,传0不分配设备,传2分配设备。示例值: 0 示例值： 0

GroupUidUinInfo

用户组和用户信息

被如下接口引用：DeleteSubAccount

名称	必选	允许NULL	类型	描述
Uid	是	否	Uint64	子用户Uid 示例值： 1258042
Uin	是	否	Uint64	子用户Uin 示例值： 130000000001
GroupId	是	否	Int64	用户组ID 如果没有任何组传递-1,传入指定组id表示将用户从组删除 示例值： 15094

ApiKeyDetail

持久密钥详情

被如下接口引用：GetApiKey

名称	必选	允许NULL	类型	描述
SecretId	否	否	String	密钥ID 示例值： AK**********
SecretKey	否	否	String	密钥Key 示例值： TR**********
CreateTime	否	否	Uint64	创建时间(时间戳) 示例值： 2014-08-03 12:00:00
Status	否	否	Uint64	状态(2:有效, 3:禁用) 示例值： 1
Source	否	否	Uint64	来源，默认0 示例值： 1
Remark	否	是	String	描述 示例值： aremark

AttachEntityOfPolicy

策略关联的实体信息

被如下接口引用：ListEntitiesForPolicy

名称	必选	允许NULL	类型	描述
Id	是	否	String	实体ID 示例值： 1000001
Name	是	是	String	实体名称 示例值： policygen-20141112201913
Uin	是	是	Uint64	实体Uin 示例值： 3449203261
RelatedType	是	否	Uint64	关联类型。1 用户关联 ； 2 用户组关联 示例值： 1

AttachedStrategyInfo

策略信息

被如下接口引用：DescribeAttachedEntityPolicies

名称	必选	允许NULL	类型	描述
PolicyId	否	否	Uint64	策略ID。 示例值： 1000
PolicyName	否	否	String	策略名称。 示例值： policygen-20141112201913
AddTime	否	是	Datetime	策略创建时间。 示例值： 2014-08-03 12:00:00
CreateMode	否	是	Uint64	创建来源，1 通过控制台创建, 2 通过策略语法创建。 示例值： 1
Description	否	是	String	策略描述。 示例值： adesc

ServicePermItem

接口

被如下接口引用：GetServicePermList

名称	必选	允许NULL	类型	描述
AddTime	否	否	Datetime	创建时间 示例值： 2014-08-03 12:00:00
ApiAddr	否	否	String	ApiAddr 示例值： ApiAddr
ApiZhName	否	否	String	中文描述 示例值： ADMIN
AuthFunction	否	否	String	鉴权接口 示例值： AuthFunction
CWildcardName	否	否	String	CWildcardName 示例值： CWildcardName
InterfaceEnName	否	否	String	接口名 示例值： CVM
InterfaceLevel	否	否	String	鉴权粒度，0:接口级别、1:资源级别 示例值： 1

名称	必选	允许NULL	类型	描述
IsAuthBusiness	否	否	String	鉴权方式，0:由云API转发鉴权、1:业务自行调用鉴权接口 示例值： 1
IsNeedObject	否	否	String	IsNeedObject 示例值： 1
IsSeen	否	否	Uint64	IsSeen 示例值： 1
IsSeenAtGenerator	否	否	String	策略生成器是否可见 示例值： 1
IsSpResource	否	否	String	IsSpResource 示例值： 1
IsUserSet	否	否	String	IsUserSet 示例值： 1
PermId	否	否	String	Id 示例值： 1000
ReadWriteDetail	否	否	String	接口类别 示例值： 接口类别：0.读取，1.写入，2.标记，3.列表
ResourceType	否	是	String	资源类别 示例值： cvm
UpdateTime	否	否	String	更新时间 示例值： 2014-08-03 12:00:00
Weight	否	否	String	Weight 示例值： 1
Writter	否	否	String	操作者 示例值： admin
ServiceName	否	否	String	服务名 示例值： cvm
ProductShortCode	否	否	String	ProductShortCode 示例值： p_cvm
ProductShortName	否	否	String	ProductShortName 示例值： p_cvm

名称	必选	允许NULL	类型	描述
ServiceType	否	否	String	服务类型 示例值： cvm
Interface	否	否	String	接口名 示例值： CVM
InterfaceName	否	否	String	接口名 示例值： CVM

ListOpenPlatform

第三方平台详细信息

被如下接口引用：ListOpenPlatforms

名称	必选	允许NULL	类型	描述
OpenId	是	否	Int64	openid 示例值： 1
AppId	是	否	String	app id 示例值： 1255000002
OpenName	是	否	String	app name 示例值： ""
OpenLogo	是	否	String	open logo 示例值： ""
OpenHome	是	否	String	第三方平台主页 示例值： ""
OpenType	是	否	Int64	授权类型 示例值： 0
Uin	是	否	Uint64	申请账号 示例值： 110000000001
Status	是	否	Int64	状态 示例值： 0
Domain	是	否	String	第三方平台域名 示例值： ""

名称	必选	允许NULL	类型	描述
State	是	否	Int64	冻结状态，0-非冻结，1-冻结 示例值： 0
Modifier	是	否	String	修改人 示例值： 110000000001
ModifyTime	是	否	String	更新时间 示例值： 2019-04-29 21:18:28
CreateTime	是	否	String	创建时间 示例值： 2019-04-29 21:18:28
Memo	是	否	String	备注 示例值： ""

UserInfo

用户信息

被如下接口引用：AddSubAccount、ListGroups

名称	必选	允许NULL	类型	描述
CanLogin	否	否	String	子账号类型，传0不可登陆控制台，传1可以登陆控制台。 示例值： 1
CountryCode	否	否	String	区号。 示例值： 86
Detail	否	否	AccountDetail	详情 示例值：查看
Name	否	否	String	名称。 示例值： name
PhoneNum	否	否	String	电话号码。 示例值： 11111111111
SystemType	否	否	String	账号类型。 示例值： Subaccount
Email	否	否	String	安全邮箱。 示例值： mail@mail.com

名称	必选	允许NULL	类型	描述
NickName	否	否	String	昵称。 示例值： nickname
Remark	否	否	String	备注。 示例值： remark
WxzsStatus	否	否	Int64	微信登陆状态。 示例值： 0
ContactMail	否	否	String	联系邮箱。 示例值： mail@mail.com
IsReceiverOwner	否	否	Int64	是否是主账号。传1代表主账号, 传0代表子账号。 示例值： 1
IdentifyType	否	否	Int64	身份类型。 示例值： 0

AttachPolicyInfo

关联策略信息

被如下接口引用：ListAttachedGroupPolicies

名称	必选	允许NULL	类型	描述
PolicyId	是	否	Uint64	策略id 示例值： 1000
PolicyName	是	是	String	策略名称 示例值： policygen-20141112201913
AddTime	是	是	Datetime	创建时间 示例值： 2014-08-03 12:00:00
CreateMode	是	是	Uint64	创建来源，1 通过控制台创建, 2 通过策略语法创建。 示例值： 1

GroupMember

用户组成员

被如下接口引用：AddUserToGroup、RemoveUserFromGroup、UpdateGroupMember

名称	必选	允许NULL	类型	描述
Uid	是	否	String	用户id 示例值： 100
GroupId	是	否	String	组id 示例值： 100

StrategyInfo

策略信息

被如下接口引用：ListPolicies

名称	必选	允许NULL	类型	描述
PolicyId	是	否	Uint64	策略ID。 示例值： 1
PolicyName	是	否	String	策略名称。 示例值： AdministratorAccess
AddTime	是	是	Datetime	策略创建时间。 示例值： 2016-06-02 19:40:09
Type	是	否	Uint64	策略类型。1 表示自定义策略，2 表示预设策略。 示例值： 1
Description	是	是	String	策略描述。 示例值： 该策略允许您管理账户内所有用户及其权限、财务相关的信息、云服务资产。
CreateMode	是	否	Uint64	创建来源，1 通过控制台创建, 2 通过策略语法创建。 示例值： 1
Attachments	否	否	Uint64	关联的用户数 示例值： 1
ServiceType	是	是	String	策略关联的产品 示例值： cam
IsAttached	否	是	Uint64	当需要查询标记实体是否已经关联策略时不为null。0表示未关联策略，1表示已关联策略 示例值： 1

名称	必选	允许NULL	类型	描述
Deactived	否	是	Uint64	是否已下线，传1代表已下线，传0代表未下线 示例值： 0
DeactivedDetail	否	是	Array of String	已下线产品列表 示例值： ["deacproduct"]
IsCheck	是	是	Uint64	是否进行安全性校验，传1代表进行安全校验，传0代表不校验 示例值： 1
PolicyDocument	否	否	String	策略语法 示例值： {\"version\":\"2.0\",\"statement\":[{\"effect\":\"allow\",\"action\":[\"cvm:ModifyDiskAttributes\"],\"resource\":[\"*\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}},{\"effect\":\"allow\",\"action\":[\"cos:PutObject\"],\"resource\":[\"qcs::cos::uid/1255000018:rules-package-ftp-1255000018/waf_rules.zip\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}}]}
UpdateTime	否	否	String	修改时间 示例值： 2016-06-02 19:40:09

AttributeInfo

属性

被如下接口引用：AddAttributeValues

名称	必选	允许NULL	类型	描述
AttributeName	否	是	String	属性名称 示例值： ""
Attribute	否	是	String	属性 示例值： ""
AttributeValue	否	是	String	属性值 示例值： ""

ResourceTypeItem

资源类型

被如下接口引用：GetServiceList

名称	必选	允许NULL	类型	描述
ResourceEnName	否	否	String	英文名 示例值： ""
ResourceName	否	否	String	中文名 示例值： ""
ResourceType	否	否	String	资源类型 示例值： ""
ServiceType	否	否	String	服务类型 示例值： ""

UserLists

GetUserListByUinList

被如下接口引用：GetUserListByUinList

名称	必选	允许NULL	类型	描述
CountryCode	否	否	String	CountryCode 示例值： 86
IsAuthed	否	否	Int64	IsAuthed 示例值： 0
UserCamUid	否	否	Int64	UserCamUid 示例值： 1280504
UserCellphone	否	否	String	UserCellphone 示例值： 11111111111
UserEmail	否	否	String	UserEmail 示例值： mail@mail.com
UserId	否	否	String	UserId 示例值： 1280504
UserUin	否	否	Uint64	UserUin 示例值： 110000000001

名称	必选	允许NULL	类型	描述
UserName	否	否	String	UserName 示例值： username
UserIsLocked	否	否	Int64	UserIsLocked 示例值： 1

FilterItem

过滤条件数据结构

被如下接口引用：ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
Operator	是	否	String	操作符 示例值： ""
Attr	是	否	String	属性 示例值： ""
Value	是	否	String	匹配值 示例值： ""

RoleInfo

角色详细信息

被如下接口引用：DescribeRoleList、GetRole

名称	必选	允许NULL	类型	描述
RoleId	是	否	String	角色ID 示例值： 12
RoleName	是	否	String	角色名称 示例值： QCS
PolicyDocument	是	否	String	角色的策略文档 示例值： {\"version\":\"2.0\",\"statement\":[{\"effect\":\"allow\",\"action\":[\"cvm:ModifyDiskAttributes\"],\"resource\":[\"*\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}},{\"effect\":\"allow\",\"action\":

名称	必选	允许NULL	类型	描述
				[\"cos:PutObject\"],\"resource\":[\"qcs::cos::uid/1255000018:rules-package-ftp-1255000018/waf_rules.zip\"],\"condition\":{\"ip_equal\":{\"qcs:ip\":[\"1.1.1.1\"]}}}]}
Description	是	否	String	角色描述 示例值： Description
AddTime	是	否	String	角色的创建时间 示例值： 2020-01-01 01:01:01
UpdateTime	是	否	String	角色的最近一次时间 示例值： 2020-01-01 01:01:01
DeletionTaskId	否	是	String	兼容公有云字段，无含义 示例值： 1
ConsoleLogin	是	否	Uint64	角色是否允许登录，传1代表允许登录，传0代表不允许 示例值： 1
RoleType	否	是	String	角色类型，取user、system或service_linked 示例值： user
SessionDuration	否	是	Uint64	有效时间 示例值： 7200

RolePolicyList

角色策略列表

被如下接口引用：GetServiceRoleInfo

名称	必选	允许NULL	类型	描述
IsHidden	否	否	Uint64	状态 示例值： 1
PolicyId	否	否	String	策略Id 示例值： 100
PolicyName	否	否	String	策略名 示例值： policygen-20141112201913

AttachedPolicyOfRole

角色关联的策略信息

被如下接口引用：ListAttachedRolePolicies

名称	必选	允许NULL	类型	描述
PolicyId	是	否	Uint64	策略ID 示例值： 1000
PolicyName	是	否	String	策略名称 示例值： policygen-20141112201913
AddTime	是	否	String	绑定时间 示例值： 2014-08-03 12:00:00
PolicyType	是	是	String	策略类型，User表示自定义策略，QCS表示预设策略 示例值： 按策略类型过滤，User表示仅查询自定义策略，QCS表示仅查询预设策略
CreateMode	是	否	Uint64	策略创建方式，1表示按产品功能或项目权限创建，其他表示按策略语法创建 示例值： 1

Filter

过滤条件组合

被如下接口引用：ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
Keywords	否	否	Array of FilterItem	过滤条件组合 示例值：查看
Operator	是	否	String	操作符 示例值： ""

GroupInfoWithRemark

GroupInfoWithRemark

被如下接口引用：ListUsersForGroup

名称	必选	允许NULL	类型	描述
GroupId	否	否	Int64	GroupId 示例值： 100
GroupName	否	否	String	GroupName 示例值： name
Remark	否	否	String	Remark 示例值： remark

AddSubAccountDetail

新增用户详情

被如下接口引用：AddSubAccount

名称	必选	允许NULL	类型	描述
Name	否	否	String	用户名 示例值： name
Token	否	否	String	token 示例值： ""
Password	否	否	String	密码 示例值： password
SecretId	否	否	String	secretid 示例值： ""
SecretKey	否	否	String	secretKey 示例值： ""
NickName	否	否	String	昵称 示例值： nickname
Uin	否	否	Uint64	uin 示例值： 110000000001
Nickname	否	否	String	Nickname 示例值： nickname

AttachedStrategyInfoPack

策略信息包

被如下接口引用：DescribeAttachedEntityPolicies

名称	必选	允许NULL	类型	描述
List	否	否	Array of AttachedStrategyInfo	策略数组，数组每个成员包括 policyId、policyName、addTime、type、description、 createMode 字段。其中： 示例值：查看
TotalNum	否	否	Uint64	策略数 示例值： 100
Id	否	否	String	入参Type=1时表示uin，2时表示groupId 示例值： 100

ExtAttr

用户属性集合

被如下接口引用：DescribeSubReceiver

名称	必选	允许NULL	类型	描述
NeedResetToken	是	否	Int64	需要重置mfa的token 示例值： 0
NeedResetStoken	是	否	Int64	需要重置mfa的stoken 示例值： 0

UserData

用户信息数据

被如下接口引用：GetAllSubUser

名称	必选	允许NULL	类型	描述
Uid	是	否	Uint64	子用户id 示例值： 1258042
Uin	是	否	Uint64	账号唯一序列号 示例值： 110000000001

名称	必选	允许NULL	类型	描述
Name	是	否	String	用户名称 示例值： name
PhoneNum	是	否	String	电话号码 示例值： 11111111111
CountryCode	是	否	String	区号 示例值： 86
PhoneFlag	是	否	Int64	电话认证标志 示例值： 1
Email	是	否	String	邮箱地址 示例值： mail@mail.com
EmailFlag	是	否	Int64	邮箱是否认证 示例值： 1
UserType	是	否	Int64	用户类型 示例值： 0
CreateTime	是	否	String	创建时间 示例值： 2019-04-29 21:18:28
WechatFlag	是	否	Int64	微信标识 示例值： 0
SystemType	是	否	String	账号系统类型 示例值： Subaccount
IsReceiverOwner	是	否	Int64	是否为主账号 示例值： 1
PermType	否	否	Array of String	PermType 示例值： ["0"]

DescribeGroupsInfo

DescribeGroupsInfo

被如下接口引用：DescribeGroups

名称	必选	允许NULL	类型	描述
Channel	否	否	Int64	Channel

名称	必选	允许NULL	类型	描述
				示例值： 3
CreateTime	否	否	String	CreateTime 示例值： 2025-02-21 11:30:44
GroupId	否	否	Int64	GroupId 示例值： 100
GroupName	否	否	String	GroupName 示例值： name
GroupNum	否	否	Int64	GroupNum 示例值： 3
GroupType	否	否	Int64	GroupType 示例值： 0
Remark	否	否	String	Remark 示例值： Remark
UserInfo	否	否	Array of ListGroupUserInfo	UserInfo 示例值：查看

ServiceApiListInfo

服务的API信息

被如下接口引用：GetServiceApiList

名称	必选	允许NULL	类型	描述
Name	是	否	String	API名称 示例值： CreateInstance
IsNeedObject	是	是	String	是否需要关联对象 示例值： ["1"]
Desc	是	是	String	描述 示例值： 创建实例
ReadWriteDetail	是	是	Uint64	接口类别：0.读取，1.写入，2.标记，3.列表 示例值： 1
InterfaceLevel	是	是	Uint64	授权粒度：0.接口级，1.资源级 示例值： 1

名称	必选	允许NULL	类型	描述
ResourceExample	是	是	String	资源六段式范例 示例值： [qcs::cvm:region:uin/110000001207:volume/*]

GroupInfo

用户组信息

被如下接口引用：GetGroupList、GetGroupsSubAccount、GetSubsGroup、ListGroups、ListGroupsForConsole、ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
GroupId	否	否	Uint64	组id 示例值： 15094
GroupName	否	否	String	组名称 示例值： groupname
Channel	否	是	Int64	息接收渠道 0:无 1: 短信 2：邮件 3：短信+邮件 示例值： 0
Remark	否	是	String	备注 示例值： remark
CreateTime	否	是	String	创建时间 示例值： 2019-04-29 21:18:28
UserInfo	否	是	Array of SubAccountFilter	用户组成员信息 示例值：查看
GroupType	否	是	Int64	用户组类型，0-自定义，1-预设 示例值： 0
GroupNum	否	否	Int64	组成员数 示例值： 2

GroupUserInfo

用户组下用户信息

被如下接口引用：GetGroupsSubAccount

名称	必选	允许NULL	类型	描述
Uid	是	是	Uint64	接收者用户id 示例值： 1258042
Uin	是	是	Uint64	账户唯一id 示例值： 130000000001
Name	是	是	String	用户名 示例值： username
PhoneNum	是	是	String	手机号 示例值： 11111111111
CountryCode	是	是	String	国家编码 示例值： 86
PhoneFlag	是	是	String	手机号标识 示例值： 1
Email	是	是	String	邮箱 示例值： mail@mail.com
EmailFlag	是	是	String	邮箱标识 示例值： 1
UserType	是	是	Int64	用户类型 示例值： 0
CreateTime	是	是	String	创建时间 示例值： 2019-04-29 21:18:28
IsReceiverOwner	是	是	String	是否是主账户 示例值： 1
SystemType	是	是	String	账户类型 示例值： subaccount
NickName	是	是	String	昵称 示例值： nick

Receiver

消息接收人信息

被如下接口引用：DescribeSubAccountContacts

名称	必选	允许NULL	类型	描述
Uid	是	否	Uint64	id 示例值： 1258042
Name	是	否	String	名字 示例值： username
Remark	是	否	String	备注 示例值： remark
PhoneNumber	是	否	String	手机号码 示例值： 11111111111
PhoneFlag	是	否	Int64	手机号码是否验证 示例值： 1
Email	是	否	String	邮箱 示例值： mail@mail.com
EmailFlag	是	否	Int64	邮箱是否验证 示例值： 1
IsReceiverOwner	是	否	Int64	是否主联系人 示例值： 1
WechatFlag	否	否	Int64	是否允许微信接收通知 示例值： 0
Uin	是	否	Uint64	账号uin 示例值： 130000000001

ListGroupUserInfo

ListGroupUserInfo

被如下接口引用：DescribeGroups、ListUsersForGroup

名称	必选	允许NULL	类型	描述
CountryCode	否	否	String	CountryCode 示例值： 86
CreateTime	否	否	String	CreateTime 示例值： 2025-01-17 14:21:07

名称	必选	允许NULL	类型	描述
Email	否	否	String	Email 示例值： mail@mail.com
EmailFlag	否	是	Int64	EmailFlag 示例值： 0
IsReceiverOwner	否	否	Int64	IsReceiverOwner 示例值： 0
JoinTime	否	否	String	JoinTime 示例值： 2025-01-17 14:21:07
Name	否	否	String	Name 示例值： Name
NickName	否	否	String	NickName 示例值： NickName
PhoneFlag	否	否	Int64	PhoneFlag 示例值： 0
PhoneNum	否	否	String	PhoneNum 示例值： 11111111111
SystemType	否	否	Int64	SystemType 示例值： 0
Uid	否	否	Uint64	Uid 示例值： 1258683
Uin	否	否	Uint64	Uin 示例值： 130000000640
UserType	否	否	Int64	UserType 示例值： 3
QywxUserId	否	否	String	QywxUserId 示例值： ""

ActionLoginFlag

登录操作敏感标识

被如下接口引用：AddSubAccount、ListGroups

名称	必选	允许NULL	类型	描述
Phone	否	否	String	电话, 传0不开启敏感操作保护, 传1开启敏感操作保护。示例值: 0 示例值： 0
Stoken	否	否	String	软Token, 传0不开启敏感操作保护, 传1开启敏感操作保护。示例值: 0 示例值： 0
Token	否	否	String	硬Token, 传0不开启敏感操作保护, 传1开启敏感操作保护。示例值: 0 示例值： 0
Ukey	否	否	String	ukey, 传0不开启敏感操作保护, 传1开启敏感操作保护。示例值: 0 示例值： 0

ServiceApiInfo

服务及其API信息

被如下接口引用：GetServiceApiList

名称	必选	允许NULL	类型	描述
Name	是	否	String	服务名称 示例值： 用户与权限
ServiceType	是	否	String	服务ID 示例值： cam
ArnDocument	是	是	String	服务介绍文档链接 示例值： https://www.example.com/document/product/378/8965
ApiList	否	是	Array of ServiceApiListInfo	API信息列表 示例值：查看
ConditionKeyList	否	是	Array of String	条件规则列表 示例值： ["qcs:ip"]

StrategyInfoForAction

策略详情

被如下接口引用：ListPoliciesByAction

名称	必选	允许NULL	类型	描述
PolicyId	否	否	Uint64	策略ID。 示例值： 1
PolicyName	否	否	String	策略名称。 示例值： AdministratorAccess
AddTime	否	否	Datetime	策略创建时间。 示例值： 2016-06-02 19:40:09
Type	否	否	Uint64	策略类型。1 表示自定义策略，2 表示预设策略。 示例值： 1
Description	否	否	String	策略描述。 示例值： 该策略允许您管理账户内所有用户及其权限、财务相关的信息、云服务资产。
CreateMode	否	否	Uint64	创建来源，1 通过控制台创建, 2 通过策略语法创建。 示例值： 1
Attachments	否	是	Uint64	关联的用户数 示例值： 1
ServiceType	否	否	String	策略关联的产品 示例值： cam
IsAttached	否	是	Uint64	当需要查询标记实体是否已经关联策略时不为null。0表示未关联策略，1表示已关联策略 示例值： 1
Deactived	否	是	Uint64	是否已下线，传1代表已下线，传0代表未下线 示例值： 0
DeactivedDetail	否	是	Array of String	已下线产品列表 示例值： ["cvm"]
IsCheck	否	是	String	是否进行安全性校验，传1代表进行安全校验，传0代表不校验 示例值： 1
PolicyDocument	否	否	String	策略语法 示例值： {"version":"2.0","statement":[{"action":["cam:*","account:*"],"resource":["*"],"effect":"allow"}]}
UpdateTime	否	否	String	修改时间 示例值： 2016-06-02 19:40:09

UserGroup

用户及用户组信息

被如下接口引用：GetUserGroupList、ListAllUserGroup

名称	必选	允许NULL	类型	描述
Uid	是	是	Uint64	接收者用户id 示例值： 1258042
Uin	是	是	Uint64	账户唯一id 示例值： 110000000001
Name	是	是	String	用户名 示例值： name
IsReceiverOwner	是	是	Int64	是否是主账户 示例值： 1
Group	是	是	Array of GroupMeta	用户组信息 示例值：查看

AttachedUserPolicyGroupInfo

用户关联策略(随组管理)信息

被如下接口引用：ListAttachedUserAllPolicies

名称	必选	允许NULL	类型	描述
GroupId	否	否	Uint64	分组ID 示例值： 10012
GroupName	否	否	String	分组名称 示例值： group_one

UpdateGroupInfo

更新用户组信息

被如下接口引用：UpdateSubAccount

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
GroupId	是	否	Uint64	用户组id 示例值： 15094
Uid	是	否	Uint64	用户id 示例值： 1258042
GroupName	否	否	String	用户组名称 示例值： name
Channel	否	否	Int64	息接收渠道 0:无 1: 短信 2：邮件 3：短信+邮件 示例值： 0

CasProviderItem

cas server 配置信息

被如下接口引用：DescribeCasProvider

名称	必选	允许NULL	类型	描述
Id	是	否	Int64	provider id 示例值： 1
CreateUin	是	否	Uint64	创建账户uin 示例值： 110000000001
OwnerUin	是	否	Uint64	主账户uin 示例值： 110000000001
Name	是	否	String	名称 示例值： name
Desc	是	否	String	描述 示例值： desc
ProviderType	是	否	Int64	provider类型 示例值： 8
Status	是	否	Int64	状态 示例值： 0
ModifyTime	是	否	String	修改时间 示例值： 2019-04-29 21:18:28

名称	必选	允许NULL	类型	描述
CreateTime	是	否	String	创建时间 示例值： 2019-04-29 21:18:28
SAMLMetadata	是	否	String	SAML元数据 示例值： ""
SAMLEntityId	是	否	String	SAML实例id 示例值： ""
SAMLSingleSignOn	是	否	String	SAML登陆跳转 示例值： ""
SAMLSingleLogout	是	否	String	SAML登出跳转 示例值： ""
SAMLKeys	是	否	String	SAML关键字 示例值： ""
Cas	是	否	String	Cas 示例值： ""
CasRoot	是	否	String	cas根地址 示例值： https://cas.yfm13.fsphere.cn/
CasLoginUrl	是	否	String	cas登陆url 示例值： https://cas.yfm13.fsphere.cn/cas/login
CasValidateUrl	是	否	String	cas校验url 示例值： https://cas.yfm13.fsphere.cn/cas/serviceValidate
CasLogoutUrl	是	否	String	cas登出url 示例值： https://cas.yfm13.fsphere.cn/cas/logout
Oauth	是	否	String	oauth配置 示例值： ""

AccountAttributeAndValue

扩展属性值

被如下接口引用：GetGroupList、GetGroupsSubAccount、GetSubsGroup、ListGroups、ListGroupsForConsole、ListMaskedSubAccounts、ListSubAccounts

名称	必选	允许NULL	类型	描述
AttributeName	否	否	String	属性名称 示例值： ""
Attribute	否	否	String	属性 示例值： ""
AttributeId	否	否	Int64	属性id 示例值： ""
ValueId	否	否	Int64	值id 示例值： ""
Uin	否	否	Int64	uin 示例值： 110000000001
Value	否	否	String	值 示例值： ""

错误码

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 cor
rect."
    },
    "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
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接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
FailedOperation.PolicyFull	用户策略数超过上限。
InvalidParameter.ParamError	非法入参。
FailedOperation.AccountSettingValueCalculateError	配置值计算异常。
InvalidParameter.RoleNameError	角色名不合法。

错误码	说明
InvalidParameter.ResourceError	策略文档的Resource字段不合法。
InvalidParameter.PolicyIdNotExist	策略ID不存在。
FailedOperation.InAsyncModifyError	数据修改中
ResourceNotFound.URLError	url解析异常。
InvalidParameter.ScopeError	Scope字段不合法。
FailedOperation.PolicyNameInUse	PolicyName字段指定的策略名已存在。
InvalidParameter.OperatePoliciesOverLimit	一次操作策略数过多。
InvalidParameter.EffectError	策略文档的Effect字段不合法。
ResourceNotFound.GroupNotExist	用户组不存在。
InvalidParameter.ActionNotExist	action不存在
InvalidParameter.ActionError	策略文档的Action字段不合法。
FailedOperation.NameAlreadyExist	用户名已存在
FailedOperation.SetLoginRuleFail	设置登录策略失败。
InvalidParameter.AttachmentFull	principal字段的授权对象关联策略数已达到上限。
InvalidParameter.PasswordLengthTooShort	密码太短。
InvalidParameter.NotSupportProduct	CAM不支持策略文档中所指定的资源类型。
InvalidParameter.VersionError	策略文档的Version字段不合法。
InvalidParameter.PolicyIdError	输入参数PolicyId不合法。
FailedOperation.IdentityExist	身份认证失败。
InternalError.SystemError	内部错误。
ResourceNotFound.PolicyIdNotFound	PolicyId指定的资源不存在。
FailedOperation.CheckPasswordError	检查密码失败
ResourceUnavailable.IDPMaxLimit	超过idp最大数量限制。
FailedOperation.SkeyExpired	Skey已过期。
InvalidParameter.PolicyNameError	PolicyName字段不合法。

错误码	说明
InvalidParameter.UserNotExist	用户对象不存在。
InvalidParameter.GroupIdError	GroupId字段不合法。
FailedOperation.Accesskey	操作访问密钥错误
FailedOperation.RefreshTokenError	刷新用户token异常。
FailedOperation.AccountSettingConfigError	账号配置元数据异常。
InvalidParameter.CreateGroupErr	创建用户组失败
InvalidParameter.RoleFull	角色数量达到上限。
InvalidParameter.RoleNotExist	角色不存在。
InvalidParameter.RoleNameInUse	相同名称的角色已存在。
InvalidParameter.UinError	Uin字段不合法。
FailedOperation.SubAccountHasKey	子账号存在密钥。
FailedOperation.SecretIdExist	SecretId已存在。
InvalidParameter.ServiceTypeError	ServiceType字段不合法。
InvalidParameter.PolicyDocumentError	PolicyDocument字段不合法。
FailedOperation.AuthCodeError	授权码异常。
FailedOperation.NoneValue	缺少配置。
UnauthorizedOperation	未授权操作
ResourceNotFound.RecordNotExist	记录不存在
FailedOperation.UnknownAccountSettingKey	未知的账号配置项。
ResourceNotFound.IdentityNotExist	身份不存在。
FailedOperation.UserrAccessTokenError	用户接入token异常。
ResourceNotFound.UserNotExist	用户不存在。
FailedOperation.AccountGroupNameNotMatch	配置组名不匹配。
InvalidParameter.PrincipalError	策略文档的principal字段不合法。
FailedOperation.UnknownAccountSettingGroup	未知的账号配置组。

错误码	说明
InvalidParameter.ConditionError	策略文档的condition字段不合法。
FailedOperation.AddIdentityError	添加认证身份失败。
InvalidParameter.DescriptionLengthOverlimit	Description入参长度不能大于300字节。
InvalidParameter.PolicyDocumentLengthOverLimit	PolicyDocument字段超过长度限制。
InvalidParameter.KeywordError	Keyword字段不合法。
ResourceNotFound.NotFound	资源不存在。
InvalidParameter.StatementError	策略文档的Statement字段不合法。

地域信息（region）
版本（2022-06-27）
API 概览

API版本

V3

公有云地域

接口名称	接口功能
DescribeRegions	查询可用地域列表
DescribeRegionsAndZones	查询地域信息的统一入口
DescribeZones	查询可用区列表

调用方式
接口签名v1

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐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。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
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

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' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    '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=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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.finance.cloud.tencent.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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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 http 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TCloudFinanceZone 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
- Node

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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

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

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

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

Java

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

public class CloudAPIDemo {
    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.finance.cloud.tencent.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.finance.cloud.tencent.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)).app
end("&");
        }
        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.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 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", "shjr"); // 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
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        '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)
```

接口签名v3

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：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 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，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。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

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

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- 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，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- 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）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 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
```

- 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 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- 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，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

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

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

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

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

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

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 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）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //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 = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\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 = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************ 步骤 3：计算签名 ************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        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("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

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

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

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

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

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

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

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

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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。

公共参数

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

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	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。
X-TC-Timestamp	Integer	是	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。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	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。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

公有云地域
查询可用地域列表

1. 接口描述

接口请求域名： region.api3.finance.cloud.tencent.com。

DescribeRegions

默认接口请求频率限制：20次/秒。

接口更新时间：2025-02-27 21:08:23。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRegions
Version	是	否	String	公共参数，本接口取值：2022-06-27
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Product	否	否	String	Product 示例值： "VPC"
Scene	否	否	Int64	Scene 示例值： ""

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	TotalCount 示例值： 10

参数名称	类型	描述
RegionSet	Array of RegionInfo	RegionSet 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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。

查询地域信息的统一入口

1. 接口描述

接口请求域名： region.api3.finance.cloud.tencent.com。

DescribeRegionsAndZones

默认接口请求频率限制：20次/秒。

接口更新时间：2025-02-27 21:05:58。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRegionsAndZones
Version	是	否	String	公共参数，本接口取值：2022-06-27
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Product	否	否	String	TCE 产品Code 示例值： "VPC"
Scene	否	否	Int64	Scene 示例值： ""
IsIgnoreWhiteList	否	否	Bool	IsIgnoreWhiteList (扩展字段) 示例值： false
ExcludeDisabledRegions	否	否	Bool	ExcludeDisabledRegions (扩展字段) 示例值： false

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	TotalCount 示例值： 3
ResourceRegionSet	Array of ResourceRegionInfo	RegionSet 示例值：查看
ProductType	String	ProductType 示例值： ""
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询可用区列表

1. 接口描述

接口请求域名： region.api3.finance.cloud.tencent.com。

DescribeZones

默认接口请求频率限制：20次/秒。

接口更新时间：2025-02-27 21:10:17。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeZones
Version	是	否	String	公共参数，本接口取值：2022-06-27
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Product	否	否	String	Product 示例值： "VPC"
Scene	否	否	Int64	Scene 示例值： ""

3. 输出参数

参数名称	类型	描述
TotalCount	Int64	TotalCount 示例值： 10
ZoneSet	Array of ZoneInfo	ZoneSet 示例值：查看

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

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

数据结构

ResourceRegionInfo

ResourceRegionInfo

被如下接口引用：DescribeRegionsAndZones

名称	必选	允许NULL	类型	描述
Region	否	是	String	Region 示例值： "Region1"
RegionId	否	是	String	RegionId 示例值： "60000052"
RegionName	否	是	String	RegionName 示例值： "Region1"
RegionArea	否	是	String	RegionArea 示例值： "Region1"
RegionState	否	是	String	RegionState 示例值： "UNAVAILABLE"
Location	否	是	String	Location 示例值： "南部"
LocationEn	否	是	String	LocationEn 示例值： ""
LocationJp	否	是	String	LocationJp 示例值： ""
LocationKo	否	是	String	LocationKo 示例值： ""
RegionNameEn	否	是	String	RegionNameEn 示例值： "hqtest-ops"
RegionNameJp	否	是	String	RegionNameJp 示例值： ""
RegionNameKo	否	是	String	RegionNameKo 示例值： "小型化双AZ集成测试环境"

名称	必选	允许NULL	类型	描述
RegionShortName	否	是	String	RegionShortName 示例值： "小型化双AZ集成测试环境"
InnerDomainName	否	是	String	InnerDomainName 示例值： ""
OuterDomainName	否	是	String	OuterDomainName 示例值： ""
RegionType	否	是	String	RegionType 示例值： ""
Weight	否	是	Int64	Weight 示例值： 0
OnlineState	否	是	String	OnlineState 示例值： ""
ProductWhiteList	否	是	String	ProductWhiteList 示例值： ""
RegionTypeMC	否	是	Int64	RegionTypeMC 示例值： 0
ResourceZoneSet	否	是	Array of ResourceZoneInfo	ResourceZoneSet 示例值：查看

ResourceZoneInfo

ResourceZoneInfo

被如下接口引用：DescribeRegionsAndZones

名称	必选	允许NULL	类型	描述
Zone	否	是	String	Zone 示例值： "hqtest-ops-1"
ZoneName	否	是	String	ZoneName 示例值： "小型化双AZ集成测试环境一区"
ZoneState	否	是	String	ZoneState 示例值： "UNAVAILABLE"

名称	必选	允许NULL	类型	描述
ZoneId	否	是	String	ZoneId 示例值： "60520001"
ZoneNameEn	否	是	String	ZoneNameEn 示例值： hqtest-ops-1
ZoneNameJp	否	是	String	ZoneNameJp 示例值： ""
ZoneNameKo	否	是	String	ZoneNameKo 示例值： ""
WhiteList	否	是	String	WhiteList 示例值： ""
ProductWhiteList	否	是	String	ProductWhiteList 示例值： ""
Weight	否	是	Int64	Weight 示例值： 0
SaleType	否	是	String	SaleType 示例值： ""
MachineRoomType	否	是	String	MachineRoomType 示例值： ""
LifeState	否	是	String	LifeState 示例值： ""
OnlineState	否	是	String	OnlineState 示例值： ""
ParentZone	否	是	String	ParentZone 示例值： ""
ParentZoneId	否	是	String	ParentZoneId 示例值： ""
ParentZoneName	否	是	String	ParentZoneName 示例值： ""
ZoneType	否	是	String	ZoneType 示例值： ""

RegionInfo

RegionInfo

被如下接口引用：DescribeRegions

名称	必选	允许NULL	类型	描述
Region	否	否	String	Region 示例值： "hqtest-ops"
RegionName	否	否	String	RegionName 示例值： "小型化双AZ集成测试环境"
RegionState	否	否	String	RegionState 示例值： "AVAILABLE"
RegionTypeMC	否	否	String	RegionTypeMC 示例值： ""
LocationMC	否	否	String	LocationMC 示例值： "南部"
RegionNameMC	否	否	String	RegionNameMC 示例值： "小型化双AZ集成测试环境"
RegionIdMC	否	否	String	RegionIdMC 示例值： "60000052"

ZoneInfo

ZoneInfo

被如下接口引用：DescribeZones

名称	必选	允许NULL	类型	描述
Zone	否	是	String	Zone 示例值： "hqtest-ops-1"
ZoneName	否	是	String	ZoneName 示例值： "小型化双AZ集成测试环境一区"
ZoneState	否	是	String	ZoneState 示例值： "AVAILABLE"
ZoneId	否	是	String	ZoneId 示例值： "60520001"

名称	必选	允许NULL	类型	描述
ZoneIdMC	否	是	String	ZoneIdMC 示例值： "60520001"
ParentZone	否	是	String	ParentZone 示例值： ""
ParentZoneId	否	是	String	ParentZoneId 示例值： ""
ParentZoneName	否	是	String	ParentZoneName 示例值： ""
ZoneType	否	是	String	ZoneType 示例值： ""
MachineRoomType	否	是	String	MachineRoomType 示例值： ""

错误码

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 cor
rect."
    },
    "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
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接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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

账号系统（account）
版本（2019-03-25）
API 概览

API版本

V3

账号系统

接口名称	接口功能
GetMultiFactorParas	获取多因子登录参数

调用方式
接口签名v1

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐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。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
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

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' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    '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=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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.finance.cloud.tencent.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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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 http 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TCloudFinanceZone 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
- Node

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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

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

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

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

Java

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

public class CloudAPIDemo {
    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.finance.cloud.tencent.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.finance.cloud.tencent.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)).app
end("&");
        }
        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.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 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", "shjr"); // 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
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        '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)
```

接口签名v3

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：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 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，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。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

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

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- 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，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- 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）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 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
```

- 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 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- 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，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

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

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

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

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

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

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 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）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //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 = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\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 = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        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("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

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

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

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

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

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

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

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

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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。

公共参数

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

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	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。
X-TC-Timestamp	Integer	是	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。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	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。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

账号系统
获取多因子登录参数

1. 接口描述

接口请求域名： account.api3.finance.cloud.tencent.com。

获取多因子登录参数

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-13 00:44:24。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetMultiFactorParas
Version	是	否	String	公共参数，本接口取值：2019-03-25
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
Data	MultIFactor	多因子配置 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.SystemError	系统异常。
InvalidParameter.ParamError	参数错误。

数据结构

AccountBankInfo

银行账号信息

被如下接口引用：GetMaskedUserInfoByLoginUin

名称	必选	允许NULL	类型	描述
Accountid	否	否	String	账号id 示例值： ""
Accountname	否	否	String	账号名称 示例值： ""
Accountbank	否	否	String	账号所属银行 示例值： ""
Provincename	否	否	String	省 示例值： ""
Provinceid	否	否	String	省ID 示例值： ""
Cityname	否	否	String	城市 示例值： ""
Cityid	否	否	String	城市ID 示例值： ""
Bankname	否	否	String	银行名称 示例值： ""
Bankid	否	否	String	银行ID 示例值： ""

AccountAttributeValue

扩展属性值

被如下接口引用：SetAttributeValues

名称	必选	允许NULL	类型	描述
Attribute	否	否	String	属性 示例值： ""
AttributeName	否	否	String	属性名称 示例值： ""
AttributeValue	否	否	String	属性值 示例值： ""
AttributeId	否	否	Int64	属性id 示例值： 0

AccountAttributeAndValue

扩展属性值

被如下接口引用：GetAttributeValues

名称	必选	允许NULL	类型	描述
AttributeName	否	否	String	属性名称 示例值： ""
Attribute	否	否	String	属性 示例值： ""
AttributeId	否	否	Int64	属性id 示例值： 0

名称	必选	允许NULL	类型	描述
ValueId	否	否	Int64	值id 示例值： 0
Uin	否	否	Int64	uin 示例值： 110000000001
Value	否	否	String	值 示例值： ""
Name	否	否	String	名字 示例值： ""

ApiKey

API密钥数据列表

被如下接口引用：CreateSelfApiKey、QuerySelfApiKey

名称	必选	允许NULL	类型	描述
SecretId	否	否	String	密钥ID 示例值： AK**********
CreateTime	否	否	Uint64	创建时间(时间戳) 示例值： 1730433390
Status	否	否	Uint64	状态(2:有效, 3:禁用) 示例值： 0
Remark	否	否	String	备注 示例值： aremark
Source	否	否	Uint64	数据源 示例值： 1
SecretKey	否	否	String	密钥Key 示例值： AB**********

ExtAttr

额外属性

被如下接口引用：DescribeSubAccounts、GetMasterList、GetMasterListV2、GetMasterListWithStatus

名称	必选	允许NULL	类型	描述
NeedResetToken	是	是	Int64	需要重置mfa的token 示例值： 0
NeedResetStoken	是	是	Int64	需要重置mfa的token 示例值： 0

SafeAuthOffsiteFlag

安全认证外围标识

被如下接口引用：GetSafeAuthConfig

名称	必选	允许NULL	类型	描述
VerifyFlag	是	否	Int64	验证标识 示例值： 0
NotifyPhone	是	否	Int64	是否进行电话通知 示例值： 0
NotifyEmail	是	否	Int64	是否进行 示例值： 0

MultIFactor

多因子配置

被如下接口引用：GetMultiFactorParas

名称	必选	允许NULL	类型	描述
MultiFactorChoices	否	否	Int64	MultiFactorChoices 示例值： 0
MultiFactorcChangable	否	否	Int64	MultiFactorcChangable 示例值： 0
MultiActionChoices	否	否	Int64	MultiActionChoices 示例值： 0
MultiActionChangable	否	否	Int64	MultiActionChangable 示例值： 0
MfaHardTokenSource	否	否	String	MfaHardTokenSource 示例值： ""

SubAccountUser

子账号列表

被如下接口引用：DescribeSubAccounts

名称	必选	允许NULL	类型	描述
Uid	否	是	Uint64	子用户 UID 示例值： 1258042
Uin	否	是	Int64	子用户用户 ID 示例值： 110000000001
OwnerUin	否	是	Uint64	主账号ID 示例值： 110000000001
Name	否	是	String	账号名称 示例值： username
PhoneNum	否	是	String	联系手机 示例值： 11111111111
PhoneFlag	否	是	Int64	电话号码是否验证 示例值： 1
Email	否	是	String	联系邮箱 示例值： mail@mail.com
EmailFlag	否	是	Int64	邮箱是否验证 示例值： 1
CountryCode	否	是	String	区号 示例值： 86
CanLogin	否	是	Int64	是否允许登录 示例值： 1
ConsoleLogin	否	是	Int64	是否允许控制台登录 示例值： 1
SystemType	否	是	String	类型 示例值： Subaccount
WxzsStatus	否	是	Int64	微信公众号关注状态 示例值： 0
NeedResetPassword	否	是	Int64	是否需要重置密码 示例值： 0
ExtAttr	否	是	ExtAttr	额外属性 示例值：查看
Remark	否	是	String	Remark 示例值： Remark
UserType	否	是	Int64	UserType 示例值： 2

名称	必选	允许NULL	类型	描述
LastLoginTime	否	是	String	LastLoginTime 示例值： time
CreateTime	否	是	String	CreateTime 示例值： time
LastLoginIP	否	是	String	LastLoginIP 示例值： ip
LastLoginIp	否	是	String	LastLoginIp 示例值： ip

AccountCurInfo

AccountCurInfo

被如下接口引用：GetMaskedUserInfoByLoginUin、GetUserInfoByLoginUin

名称	必选	允许NULL	类型	描述
Name	否	否	String	Name 示例值： name
Type	否	否	Int64	Type 示例值： 0
Area	否	否	Int64	Area 示例值： 0
Idcardtype	否	否	Int64	Idcardtype 示例值： 0
Idcard	否	否	String	Idcard 示例值： ""
Organizationcode	否	否	String	Organizationcode 示例值： ""
AuthenticateType	否	否	String	AuthenticateType 示例值： ""

OwnerInfo

主账户信息

被如下接口引用：GetMasterList、GetMasterListV2、GetMasterListWithStatus

名称	必选	允许NULL	类型	描述
Uin	否	是	Uint64	账户唯一id 示例值： 110000000001
UserName	否	是	String	用户名 示例值： username
DisplayName	否	是	String	显示名称 示例值： username
CheckStatus	否	是	Int64	账户状态 示例值： 0
Account	否	是	String	用户名 示例值： username
IsDefaultOwner	否	是	Bool	是否为该子账户的默认主账户 示例值： 1
Phone	否	是	String	手机号，打掩码处理 示例值： 11111111111
Email	否	是	String	邮箱，掩码处理 示例值： mail@mail.com
ExtAttr	否	是	ExtAttr	扩展属性 示例值：查看

名称	必选	允许NULL	类型	描述
LoginMfa	否	是	LoginMfa	多因子认证 示例值：查看

MfaActions

MfaActions

被如下接口引用：ModifySensitiveAction

名称	必选	允许NULL	类型	描述
ActionName	否	否	String	云api名字 示例值： cam:Test
Description	否	否	String	备注 示例值： desc
CreateTime	否	否	String	创建时间 示例值： 2024-10-21T07:44:53Z
UpdateTime	否	否	String	更新时间 示例值： 2024-10-21T07:44:53Z
IsActivate	否	否	Int64	是否激活，0代表未激活，1代表激活 示例值： 0代表未激活，1代表激活

LoginInfo

登录信息

被如下接口引用：GetLastLoginInfo、GetLoginInfo

名称	必选	允许NULL	类型	描述
ClientIP	否	否	String	客户端ip 示例值： 111.206.94.148
ClientUA	否	否	String	客户端代理 示例值： Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/130.0.0.0 Safari/537.36
DeviceId	否	否	String	设备id 示例值： 546d984daae133f2434c1aa1b0ca1791
Id	否	否	Int64	id 示例值： 0
Location	否	否	String	登录地址 示例值： location
LoginFlag	否	是	Int64	登录标识 示例值： 0
LoginTime	否	否	String	登录时间 示例值： 2019-04-29 21:18:28
OwnerUin	否	否	Uint64	主账户uin 示例值： 110000000001
Uin	否	否	Uint64	uin 示例值： 110000000001
UpdateTime	否	否	String	更新时间 示例值： 2019-04-29 21:18:28
LoginId	否	是	Int64	LoginId 示例值： 20

SafeAuthFlag

安全认证标识

被如下接口引用：GetSafeAuthConfig、SetSafeAuthFlag

名称	必选	允许NULL	类型	描述
Phone	是	否	Int64	是否进行手机号认证，0-否，1-是 示例值： 0
Token	是	否	Int64	是否进行token认证，0-否，1-是 示例值： 0
Stoken	是	否	Int64	是否进行stoken认证，0-否，1-是 示例值： 0
Ukey	否	否	Int64	是否进行ukey检查 示例值： 0
Global	否	否	Int64	Global 示例值： 0

ApiKeyDetail

持久密钥详情

被如下接口引用：GetSelfApiKey

名称	必选	允许NULL	类型	描述
SecretId	否	否	String	密钥ID 示例值： AK**********
SecretKey	否	否	String	密钥Key 示例值： TRGBnnaq3YS7oHos6tZjgt271kttOFmb
CreateTime	否	否	Uint64	创建时间(时间戳) 示例值： 1730433390
Status	否	否	Uint64	状态(2:有效, 3:禁用) 示例值： 1
Remark	否	否	String	备注 示例值： aremark
Source	否	否	Uint64	数据源 示例值： 1

AccountAttribute

扩展属性

被如下接口引用：GetAttributeName

名称	必选	允许NULL	类型	描述
AttributeId	否	否	Int64	id 示例值： 0
AttributeName	否	否	String	属性名称 示例值： ""
Attribute	否	否	String	属性 示例值： ""

LoginMfa

登陆多因子认证

被如下接口引用：GetMasterList、GetMasterListV2、GetMasterListWithStatus

名称	必选	允许NULL	类型	描述
NeedAuth	是	否	Int64	是否需要认证 示例值： 0
AuthType	是	否	Int64	认证类型 示例值： 0

名称	必选	允许NULL	类型	描述
AuthPhone	是	否	String	认证手机号 示例值： 11111111111

AttributeUserInfo

用户信息

被如下接口引用：GetUserByAttributeValue

名称	必选	允许NULL	类型	描述
Uin	否	否	Uint64	uin 示例值： 110000000001
UserName	否	否	String	用户名称 示例值： username
NickName	否	否	String	用户名称 示例值： nickname

CategoryList

账号类别

被如下接口引用：DescribeUserCategory

名称	必选	允许NULL	类型	描述
CategoryId	否	否	Int64	类别Id 示例值： ""
CategoryKey	否	否	String	类别键 示例值： ""
CategoryValue	否	否	String	类别值 示例值： ""

SafeAuthFlagInfo

安全认证信息

被如下接口引用：GetSafeAuthConfig

名称	必选	允许NULL	类型	描述
LoginFlag	是	是	SafeAuthFlag	登陆标识 示例值：查看
ActionFlag	是	是	SafeAuthFlag	action标识 示例值：查看
OffsiteFlag	是	是	SafeAuthOffsiteFlag	外围标识 示例值：查看

SecretIDLastUsedRow

密钥ID最近访问列表

被如下接口引用：GetSecurityLastUsed

名称	必选	允许NULL	类型	描述
SecretId	否	否	String	密钥ID 示例值： AK**********
LastUsedDate	否	是	String	最后访问日期 示例值： null
LastSecretUsedDate	否	是	Int64	最后密钥访问日期 示例值： null

SafeAuthTokenInfo

安全认证token信息

被如下接口引用：GetSafeAuthConfig

名称	必选	允许NULL	类型	描述
Status	否	否	Int64	状态，0-未分配，1-已分配，2-已绑定，3-已解绑 示例值： 1
TokenSn	否	否	String	token序列号 示例值： URDA6GO7X73ONMIA
TokenType	否	否	Int64	token类型，1-hard token, 2-soft token，3-ukey 示例值： 1
TokenSource	否	否	String	设备厂商来源 示例值： URDA6GO7X73ONMIA
Id	否	否	Int64	Id 示例值： 1

错误码

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 cor
rect."
        },
        "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
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接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
FailedOperation.UnknownTokenType	不识别的设备类型。
ResourceNotFound.UserNotExists	用户不存在。
FailedOperation.CheckPasswordError	密码校验失败
FailedOperation.MfaTokenNotDistribute	mfa设备未分配。

错误码	说明
FailedOperation.PasswordMismatch	密码不一致
FailedOperation.BindNoneHardToken	三方设备未绑定。
FailedOperation.MfaTokenAlreadyDistribute	mfa设备已分配。
InvalidParameter.ParamError	参数错误。
InvalidParameter.TokenTypeError	token类型参数异常。
FailedOperation.MfaTokenUnBindFail	mfa设备解绑失败。
FailedOperation.UrgentCodeFrequentLimit	紧急口令创建限频。
FailedOperation.MfaTokenNoDistributeError	mfa绑定失败
FailedOperation.DistributeMfaTokenError	mfa设备分配异常。
FailedOperation.BindNoneSoftToken	虚拟设备未绑定。
InvalidParameter.HardTokenSnMiss	hard token 密钥参数缺失。
LimitExceeded.SubAccountIsFull	子账号操作了限制。
InternalError.SystemError	系统异常。
InvalidParameter.PtLoginVerifyNoPass	校验失败
FailedOperation.MfaTokenAlreadyBind	已绑定mfa设备。
FailedOperation.HardTokenAlreadyBind	用户已绑定hard token。
InvalidParameter.VerifyFailed	验证失败
FailedOperation.SoftTokenNotDistribute	虚拟设备未分配。
InvalidParameter.PasswordDecryptError	密码加密方式错误。
FailedOperation.HardTokenNotDistribute	三方设备未分配。
InvalidParameter.PasswordNotMatch	密码不匹配。
FailedOperation.InSafeAuthProtection	安全认证保护中。
FailedOperation.MfaTokenBindFail	mfa设备绑定失败。

项目目录管理（org）
版本（2021-10-01）
API 概览

API版本

V3

组织管理相关接口

接口名称	接口功能
AddOrganization	添加组织
AddOrganizationMemberPolicy	添加用户到组织并授权
DeleteOrganization	删除组织
DeleteOrganizationMembers	删除组织成员
DescribeOrganizationMembers	查看组织成员
DescribeOrganizationNonMembers	查询所有非组织成员
DescribeOrganizationPoliciesTemplate	查询组织通用策略模板
DescribeOrganizationProjects	查询组织下的项目列表
DescribeOrganizationQuotas	查询组织下的项目配额
DescribeOrganizationResources	查询组织下项目资源列表
DescribeOrganizations	查询组织
ModifyOrganization	修改组织
ModifyOrganizationMemberPolicy	修改组织成员策略
ModifyOrganizationProjects	修改组织项目

调用方式
接口签名v1

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐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。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
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

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' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    '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=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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.finance.cloud.tencent.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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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 http 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TCloudFinanceZone 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
- Node

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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

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

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

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

Java

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

public class CloudAPIDemo {
    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.finance.cloud.tencent.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.finance.cloud.tencent.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)).app
end("&");
        }
        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.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 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", "shjr"); // 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
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        '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)
```

接口签名v3

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：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 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，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。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

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

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- 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，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- 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）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 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
```

- 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 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- 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，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

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

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

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

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

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

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 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）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //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 = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\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 = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        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("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

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

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

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

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

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

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

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

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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。

公共参数

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

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	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。
X-TC-Timestamp	Integer	是	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。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	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。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

组织管理相关接口
添加组织

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

添加组织

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-09 16:59:51。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AddOrganization
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ParentId	是	否	String	父组织id，如果创建一级组织则填写“root” 示例值： root
OrgName	是	否	String	组织名称，长度最大64 示例值： org1

3. 输出参数

参数名称	类型	描述
OrgId	String	组织id，例如：org-7737f12a 示例值： org-9842cb32
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InvalidParameter.EmptyParameter	有参数为空
InvalidParameter.OrganizationNameTooLong	名称太长
UnauthorizedOperation.AuthFailedError	鉴权失败
InternalError.DatabaseError	db 错误

添加用户到组织并授权

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

添加用户到组织并授权

默认接口请求频率限制：20次/秒。

接口更新时间：2021-11-11 15:10:44。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AddOrganizationMemberPolicy
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OrgId	是	否	String	组织id 示例值： org-9842cb32
Uins	是	否	Array of Uint64	用户uin列表 示例值： [111,112]
PolicyNames	是	否	Array of String	策略名称列表 示例值： ["policygen-20141112201913","policygen-20141112201914"]

3. 输出参数

参数名称	类型	描述
FailedUins	Array of AddOrgMemberPolicyRes	添加失败的用户列表 示例值：查看
SuccessfulUins	Array of AddOrgMemberPolicyRes	添加成功的用户列表 示例值：查看

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败

删除组织

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

删除组织，要求当前组织和其子组织没有关联项目，否则不允许删除

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-09 17:00:47。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteOrganization
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OrgId	是	否	String	组织id 示例值： org-9842cb32

3. 输出参数

参数名称	类型	描述
OrgId	String	组织id 示例值： org-9842cb32
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InvalidParameter.EmptyParameter	有参数为空
UnauthorizedOperation.AuthFailedError	鉴权失败
InternalError.DatabaseError	db 错误
FailedOperation.OrganizationProjectNotEmpty	还关联项目Error

删除组织成员

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

删除组织成员

默认接口请求频率限制：20次/秒。

接口更新时间：2021-11-11 15:10:44。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DeleteOrganizationMembers
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OrgId	是	否	String	组织id 示例值： org-9842cb32
Uins	是	否	Array of Uint64	uin列表 示例值： [111,112]

3. 输出参数

参数名称	类型	描述
Uins	Array of Uint64	uin列表 示例值： [111,112]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败

查看组织成员

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

查看组织成员

默认接口请求频率限制：20次/秒。

接口更新时间：2024-10-14 19:37:38。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeOrganizationMembers
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PageNumber	否	否	Uint64	页数 示例值： 100
PageSize	否	否	Uint64	每页显示条数 示例值： 100
Filter	否	否	Filter	过滤条件 示例值：查看
OrgId	是	否	String	组织id 示例值： org-9842cb32

3. 输出参数

参数名称	类型	描述
MemberSet	Array of	组织成员

参数名称	类型	描述
	OrganizationMember	示例值：查看
TotalCount	Uint64	总数 示例值： 100
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败

查询所有非组织成员

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

查询所有非组织成员

默认接口请求频率限制：20次/秒。

接口更新时间：2024-10-14 19:32:17。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeOrganizationNonMembers
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OrgId	是	否	String	组织id 示例值： org-9842cb32
PageNumber	否	否	Uint64	页数 示例值： 100
PageSize	否	否	Uint64	每页显示条数 示例值： 100
Filter	否	否	Filter	过滤条件 示例值：查看

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
MemberSet	Array of OrganizationNonMember	非组织成员 示例值：查看
TotalCount	Uint64	总数 示例值： 100
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败

查询组织通用策略模板

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

查询组织通用策略模板

默认接口请求频率限制：20次/秒。

接口更新时间：2024-10-14 19:32:05。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeOrganizationPoliciesTemplate
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PageNumber	否	否	Uint64	页数 示例值： 100
PageSize	否	否	Uint64	每页显示条数 示例值： 100

3. 输出参数

参数名称	类型	描述
PolicySet	Array of Policy	策略列表 示例值：查看
TotalCount	Uint64	总数 示例值： 100

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败

查询组织下的项目列表

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

查询组织下的项目列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-09 16:57:10。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeOrganizationProjects
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PageNumber	否	否	Uint64	页数，例如：1 示例值： 100
PageSize	否	否	Uint64	每页显示条数 示例值： 100
Filter	否	否	Filter	过滤条件 示例值：查看
OrgId	是	否	String	组织id 示例值： org-9842cb32

3. 输出参数

参数名称	类型	描述
ProjectSet	Array of	项目列表

参数名称	类型	描述
	Project	示例值：查看
TotalCount	Uint64	总数 示例值： 100
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
UnauthorizedOperation.AuthFailedError	鉴权失败
InternalError.DatabaseError	db 错误
InvalidParameter.EmptyParameter	有参数为空

查询组织下的项目配额

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

查询组织下的项目配额

默认接口请求频率限制：20次/秒。

接口更新时间：2023-03-07 14:51:47。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeOrganizationQuotas
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OrgId	是	否	String	组织id 示例值： org-9842cb32
PageNumber	否	否	Uint64	页数 示例值： 100
PageSize	否	否	Uint64	每页显示条数 示例值： 100
Filter	否	否	ProjectQuotaFilter	过滤条件 示例值：查看

3. 输出参数

参数名称	类型	描述
QuotaSet	Array of ProjectQuota	配额列表 示例值：查看
TotalCount	Uint64	总数 示例值： 100
ProductSet	Array of ProductInfo	产品信息列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InvalidParameter.EmptyParameter	有参数为空
UnauthorizedOperation.AuthFailedError	鉴权失败
InternalError.DatabaseError	db 错误

查询组织下项目资源列表

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

查询组织下项目资源列表

默认接口请求频率限制：20次/秒。

接口更新时间：2024-10-14 19:31:52。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeOrganizationResources
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OrgId	是	否	String	组织id 示例值： org-9842cb32
PageNumber	是	否	Uint64	页数 示例值： 100
PageSize	是	否	Uint64	每页条数 示例值： 100
Filter	是	否	ProjectResourceFilter	过滤条件 示例值：查看

3. 输出参数

参数名称	类型	描述
ResourceSet	Array of Resource	资源列表 示例值：查看
TotalCount	Uint64	总数 示例值： 100
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
UnauthorizedOperation.AuthFailedError	鉴权失败
InternalError.DatabaseError	db 错误
InvalidParameter.EmptyParameter	有参数为空

查询组织

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

查询组织

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-09 17:00:25。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeOrganizations
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Filter	否	否	DescribeOrganizationsFilter	过滤条件 示例值：查看
MemberUin	否	否	String	成员UIN 示例值： 100

3. 输出参数

参数名称	类型	描述
OrgSet	Array of Organization	组织列表 示例值：查看

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InvalidParameter.EmptyParameter	有参数为空
UnauthorizedOperation.AuthFailedError	鉴权失败
InternalError.DatabaseError	db 错误

修改组织

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

修改组织

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-09 17:00:10。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyOrganization
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OrgId	是	否	String	组织id 示例值： org-9842cb32
OrgName	是	否	String	组织名称 示例值： org1

3. 输出参数

参数名称	类型	描述
OrgId	String	组织id 示例值： org-9842cb32
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败
InvalidParameter.EmptyParameter	有参数为空
InvalidParameter.OrganizationNameTooLong	名称太长
UnauthorizedOperation.AuthFailedError	鉴权失败
InternalError.DatabaseError	db 错误

修改组织成员策略

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

修改组织成员策略

默认接口请求频率限制：20次/秒。

接口更新时间：2021-11-11 15:10:44。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyOrganizationMemberPolicy
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OrgId	是	否	String	组织id 示例值： org-9842cb32
AccountUin	是	否	Uint64	账号uin 示例值： 100
PolicyNames	是	否	Array of String	需要具有的策略列表 示例值： ["policygen-20141112201913","policygen-20141112201914"]

3. 输出参数

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

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败

修改组织项目

1. 接口描述

接口请求域名： org.api3.finance.cloud.tencent.com。

添加/移除组织项目

默认接口请求频率限制：20次/秒。

接口更新时间：2021-11-11 15:10:44。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：ModifyOrganizationProjects
Version	是	否	String	公共参数，本接口取值：2021-10-01
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OrgId	是	否	String	组织id 示例值： org-9842cb32
Operate	是	否	String	操作类型，只能是“Add”或“Move” 示例值： Add
Projects	是	否	Array of String	需要添加/移除的项目 示例值： ["pr-dcd34c11","pr-dcd34c12"]

3. 输出参数

参数名称	类型	描述
FailedProjects	Array of String	添加/移除失败的项目列表 示例值： ["pr-dcd34c11","pr-dcd34c12"]

参数名称	类型	描述
SuccessfulProjects	Array of String	添加/移除成功的项目列表 示例值： ["pr-dcd34c11","pr-dcd34c12"]
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.ThirdPartServiceBadRequest	服务调用失败

数据结构

Policy

组织策略

被如下接口引用：DescribeOrganizationPoliciesTemplate

名称	必选	允许NULL	类型	描述
PolicyName	是	否	String	策略名称 示例值： policygen-1
Description	是	否	String	策略描述 示例值： policydesc

ProjectQuota

项目配额

被如下接口引用：DescribeOrganizationQuotas

名称	必选	允许NULL	类型	描述
ProjectId	是	否	String	项目id 示例值： pr-9842cb32
ProjectName	是	否	String	项目名称 示例值： pr1
ProductName	是	否	String	产品名称 示例值： cvm
ProductCode	是	否	String	产品code 示例值： p_cvm
SubProductCode	是	是	String	二级产品 示例值： sp_cvm_vself2
BillingItemCode	是	是	String	三级产品 示例值： v_cvm_cpu
SubBillingItemCode	是	是	String	四级产品 示例值： sv_cvm_mem_s3

名称	必选	允许NULL	类型	描述
QuotaKey	是	否	String	配额键 示例值： p_cvm###
QuotaName	是	是	String	配额项名称 示例值： u4e91u786cu76d8CBS-u5e26u5bbdu6027u80fd
QuotaValue	是	否	String	总配额 示例值： 100
QuotaLeft	是	否	Uint64	剩余配额 示例值： 100
QuotaUsed	是	否	Uint64	已使用配额 示例值： 100
CreateTime	是	否	String	创建时间 示例值： 2014-08-03 12:00:00
UpdateTime	是	否	String	更新时间 示例值： 2014-08-03 12:00:00
SubProductName	是	否	String	子产品名称 示例值： eip_hour
Unit	是	否	String	单位 示例值： 100
BillingItemName	是	否	String	产品项 示例值： CPU
SubBillingItemName	是	否	String	产品细项 示例值： 分布式数据库TDSQL-IO版-CPU-单份

OwnedPolicies

具有的组织策略

被如下接口引用：DescribeOrganizationMembers

名称	必选	允许NULL	类型	描述
PolicyId	是	否	Uint64	策略id 示例值： 100

名称	必选	允许NULL	类型	描述
PolicyName	是	否	String	策略名称 示例值： policygen-1
Description	是	否	String	策略描述 示例值： policydesc

ProjectResourceFilter

项目资源过滤条件

被如下接口引用：DescribeOrganizationResources

名称	必选	允许NULL	类型	描述
ProductCode	否	否	String	产品code 示例值： p_cvm
OrgIds	否	否	Array of String	组织id列表 示例值： ["org-9842cb32","org-9842cb33"]
Product	否	否	String	产品名称 示例值： cvm

AddOrgMemberPolicyRes

添加用户到组织并授权结果

被如下接口引用：AddOrganizationMemberPolicy

名称	必选	允许NULL	类型	描述
Uin	是	否	Uint64	用户uin 示例值： 100
PolicyName	是	否	String	策略名称 示例值： policy1

OrganizationMember

组织成员

被如下接口引用：DescribeOrganizationMembers

名称	必选	允许NULL	类型	描述
Uin	是	否	Uint64	用户uin 示例值： 100
Name	是	否	String	用户名 示例值： admin
OwnedPolicies	是	否	Array of OwnedPolicies	具有的组织策略 示例值：查看
JoinTime	是	否	Datetime	加入组织时间 示例值： 2014-08-03 12:00:00

ProjectQuotaFilter

配额查询条件

被如下接口引用：DescribeOrganizationQuotas

名称	必选	允许NULL	类型	描述
ProductCode	否	否	String	一级产品 示例值： p_cvm
OrgIds	否	否	Array of String	组织id列表 示例值： ["org-9842cb32","org-9842cb33"]

OrganizationNonMember

非组织成员

被如下接口引用：DescribeOrganizationNonMembers

名称	必选	允许NULL	类型	描述
Uin	是	否	Uint64	用户uin 示例值： 100
Name	是	否	String	用户名 示例值： admin

ProductInfo

产品信息

被如下接口引用：DescribeOrganizationQuotas

名称	必选	允许NULL	类型	描述
ProductCode	否	否	String	产品code 示例值： p_cvm
ProductName	否	否	String	产品名称 示例值： cvm

Project

项目

被如下接口引用：DescribeOrganizationProjects

名称	必选	允许NULL	类型	描述
ProjectId	是	否	String	项目ID 示例值： pr-9842cb32
ProjectName	是	否	String	项目名称 示例值： pr1
Creator	是	否	String	创建者 示例值： admin
CreateTime	是	否	Datetime	创建时间 示例值： 2014-08-03 12:00:00
CreatorUin	是	否	Uint64	创建者uin 示例值： 100
OrgId	是	否	String	组织id，不为空说明已经加入其它组织 示例值： org-9842cb32
OrgName	是	否	String	组织名称 示例值： org1
OrgOperator	是	否	String	转入组织操作人 示例值： admin

名称	必选	允许NULL	类型	描述
OrgOperationTime	是	否	Datetime	转入组织时间 示例值： 2014-08-03 12:00:00

Filter

过滤条件

被如下接口引用：DescribeOrganizationMembers、DescribeOrganizationNonMembers、DescribeOrganizationProjects

名称	必选	允许NULL	类型	描述
Keyword	否	否	String	过滤条件 示例值： 100
OrgIds	否	否	Array of String	组织id列表 示例值： ["org-9842cb32","org-9842cb33"]

Organization

组织

被如下接口引用：DescribeOrganizations

名称	必选	允许NULL	类型	描述
Id	否	否	Uint64	Id 示例值： 100
OrgId	否	否	String	组织id 示例值： org-9842cb32
OrgName	否	否	String	组织名称 示例值： org-9842cb32
CreatorUin	否	否	String	创建人uin 示例值： 100
Creator	否	否	String	创建人 示例值： admin

名称	必选	允许NULL	类型	描述
CreateTime	否	否	String	创建时间 示例值： 2014-08-03 12:00:00
Children	否	否	Array of Organization	子组织 示例值：查看

Resource

资源

被如下接口引用：DescribeOrganizationResources

名称	必选	允许NULL	类型	描述
ProductCode	是	否	String	产品code 示例值： p_cvm
ProductGroupName	是	否	String	产品大类名称 示例值： cvm
ProductName	是	否	String	产品名称 示例值： cvm
ProjectId	是	否	String	项目id 示例值： pr-dcd34c11
ProjectName	是	否	String	项目名称 示例值： pr1
RegionId	是	否	Uint64	地域ID 示例值： 5000001
RegionName	是	否	String	地域名 示例值： region1
RegionEnName	是	否	String	地域英文名 示例值： region1
ResourceId	是	否	String	资源ID 示例值： ins-asd223
ResourceName	是	否	String	资源名称 示例值： ins1

名称	必选	允许NULL	类型	描述
ResourceType	是	否	String	资源类型 示例值： cvm
ServiceType	是	否	String	ServiceType 示例值： cvm

DescribeOrganizationsFilter

查询组织搜索条件

被如下接口引用：DescribeOrganizations

名称	必选	允许NULL	类型	描述
Level	否	否	Uint64	级别，默认为3 表示搜索第一级、第二级和第三级符合搜索条件的组织 示例值： 3
Keyword	否	否	String	搜索条件 示例值： 100
OrgId	否	否	String	组织id 示例值： org-9842cb32

错误码

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 cor
rect."
    },
    "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
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接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s)请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
FailedOperation.OrganizationProjectNotEmpty	还关联项目Error
InternalError.ThirdPartServiceBadRequest	服务调用失败
InternalError.DatabaseError	db 错误
UnauthorizedOperation.AuthFailedError	鉴权失败

错误码	说明
InvalidParameter.EmptyParameter	有参数为空
InvalidParameter.OrganizationNameTooLong	名称太长

平台（platform）
版本（2019-03-14）
API 概览

API版本

V3

platform

接口名称	接口功能
GetCustomAccount	获取租户端主账号
GetCustomSubAccount	获取租户端子账号
QueryCustomAccount	查询租户端账号

调用方式
接口签名v1

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐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。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
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

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' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    '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=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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.finance.cloud.tencent.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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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 http 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TCloudFinanceZone 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
- Node

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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

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

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

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

Java

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

public class CloudAPIDemo {
    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.finance.cloud.tencent.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.finance.cloud.tencent.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)).app
end("&");
        }
        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.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 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", "shjr"); // 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
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        '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)
```

接口签名v3

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：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 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，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。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

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

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- 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，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- 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）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 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
```

- 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 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- 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，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

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

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

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

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

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

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 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）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //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 = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\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 = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        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("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

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

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

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

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

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

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

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

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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。

公共参数

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

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	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。
X-TC-Timestamp	Integer	是	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。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	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。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

platform
获取租户端主账号

1. 接口描述

接口请求域名： platform.api3.finance.cloud.tencent.com。

获取全部租户端主账号

默认接口请求频率限制：30次/秒。

接口更新时间：2025-02-28 14:55:14。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetCustomAccount
Version	是	否	String	公共参数，本接口取值：2019-03-14
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
PageNum	是	否	Int64	分页 示例值： 1
PageSize	是	否	Int64	页面数量 示例值： 100
GetSecretField	否	否	Bool	是否获取加密key数据 示例值： true
SortKey	否	否	String	排序key 示例值： updateTime
SortTurn	否	否	Int64	排序顺序 示例值： 0

3. 输出参数

参数名称	类型	描述
AccountList	Array of AccountList	用户列表 示例值：查看
TotalNum	Int64	用户数量 示例值： 100
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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。

获取租户端子账号

1. 接口描述

接口请求域名： platform.api3.finance.cloud.tencent.com。

通过主账号拉取子账号

默认接口请求频率限制：30次/秒。

接口更新时间：2019-09-30 10:58:45。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetCustomSubAccount
Version	是	否	String	公共参数，本接口取值：2019-03-14
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
OwnerUin	是	否	String	主账号uin 示例值： 110000000145
GetSecretField	否	否	Bool	是否获取加密数据 示例值： true

3. 输出参数

参数名称	类型	描述
SubAccountList	Array of AccountList	用户列表 示例值：查看
TotalNum	Int64	用户数量 示例值： 100

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

4. 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。

查询租户端账号

1. 接口描述

接口请求域名： platform.api3.finance.cloud.tencent.com。

运营端客户管理查询租户端账号

默认接口请求频率限制：20次/秒。

接口更新时间：2019-09-30 11:21:49。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：QueryCustomAccount
Version	是	否	String	公共参数，本接口取值：2019-03-14
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Account	否	否	String	租户端登录名 示例值： account_name
PhoneNumber	否	否	Int64	租户手机号码 示例值： 13773345315
NickName	否	否	String	租户昵称 示例值： nickname
CustomAppId	否	否	Int64	租户端appid 示例值： 1255000140
PageNum	是	否	Int64	分页 示例值： 100
PageSize	是	否	Int64	分页大小 示例值： 100

参数名称	必选	允许NULL	类型	描述
CustomUin	否	否	Int64	账号uin 示例值： 110000000145
SortKey	否	否	String	排序key 示例值： updateTime
SortTurn	否	否	Int64	顺序或者逆序，0：顺序，1：逆序 示例值： 0
StartLastLoginTime	否	否	Int64	开始时间 示例值： 1736331132
EndLastLoginTime	否	否	Int64	结束时间 示例值： 1736331132

3. 输出参数

参数名称	类型	描述
TotalNum	Int64	用户数量 示例值： 100
AccountList	Array of AccountList	用户列表 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

数据结构

IdKeys

秘钥数据

被如下接口引用：GetCustomAccount、GetCustomSubAccount、QueryCustomAccount

名称	必选	允许NULL	类型	描述
CreateTime	是	否	Int64	createTime 示例值： 1736331132
SecretId	是	否	String	秘钥id 示例值： akid
SecretKey	是	否	String	秘钥key 示例值： aksk
Source	是	否	Int64	source 示例值： 0
Status	是	否	String	启用状态 示例值： 0

AccountList

主账号列表

被如下接口引用：GetCustomAccount、GetCustomSubAccount、QueryCustomAccount

名称	必选	允许NULL	类型	描述
Active	是	否	Int64	是否激活 示例值： 1
AddTimestamp	是	否	String	新增日期 示例值： 2025-01-08 18:12:46
AppId	是	否	Int64	APPID 示例值： 1255000140
BindStatus	是	否	Int64	绑定状态 示例值： 0

名称	必选	允许NULL	类型	描述
LastLoginTime	是	否	String	上次登录时间 示例值： 2025-01-08 18:12:46
Mail	是	否	String	邮件 示例值： example@mail.com
ModTimestamp	是	否	String	修改日期 示例值： 2025-01-08 18:12:46
NickName	是	否	String	昵称 示例值： nickname
OwnerUin	是	否	Int64	输主uin 示例值： 110000000145
PhoneNumber	是	否	String	手机号码 示例值： 13773345315
Remark	是	否	String	标记 示例值： remark
Status	是	否	Int64	状态 示例值： 0
SubCount	是	否	Int64	子账号数量 示例值： 110000000145
Uid	是	否	Int64	uid 示例值： 0
Uin	是	否	Int64	uin 示例值： 110000000145
UinType	是	否	String	uintype 示例值： 1
UserName	是	否	String	username 示例值： name
IdKeys	否	否	Array of IdKeys	秘钥 示例值：查看
Password	否	否	String	密码 示例值： password
IsRegSucc	否	否	Int64	IsRegSucc 示例值： 1

名称	必选	允许NULL	类型	描述
ContactEmail	否	否	String	ContactEmail 示例值： mail@mail.com
UserAttributeAndValues	否	否	Array of String	UserAttributeAndValues 示例值： value
Account	否	否	String	Account 示例值： account
AllowReportLoss	否	否	Int64	AllowReportLoss 示例值： 1
LoginStatus	否	否	Int64	LoginStatus 示例值： 1
IdentifyType	否	否	Int64	IdentifyType 示例值： 1
QywxUserId	否	否	String	QywxUserId 示例值： userid
UserCategory	否	否	Array of String	UserCategory 示例值： user
CountryCode	否	否	String	CountryCode 示例值： 86

错误码

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 cor
rect."
    },
    "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
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接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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

密钥服务（sts）
版本（2018-08-13）
API 概览

API版本

V3

秘钥相关接口

接口名称	接口功能
AssumeRole	申请扮演角色
GetConsoleSessionToken	获取控制台临时证书
GetFederationToken	获取联合身份临时访问凭证
GetThirdPartyFederationToken	获取用户临时证书（第三方开发商）

调用方式
接口签名v1

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐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。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
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

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' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    '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=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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.finance.cloud.tencent.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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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 http 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TCloudFinanceZone 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
- Node

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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

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

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

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

Java

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

public class CloudAPIDemo {
    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.finance.cloud.tencent.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.finance.cloud.tencent.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)).app
end("&");
        }
        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.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 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", "shjr"); // 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
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        '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)
```

接口签名v3

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：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 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，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。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

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

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- 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，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- 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）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 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
```

- 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 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- 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，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

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

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

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

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

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

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 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）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //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 = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\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 = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        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("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

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

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

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

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

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

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

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

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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。

公共参数

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

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	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。
X-TC-Timestamp	Integer	是	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。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	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。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

秘钥相关接口
申请扮演角色

1. 接口描述

接口请求域名： sts.api3.finance.cloud.tencent.com。

申请扮演角色

默认接口请求频率限制：20次/秒。

接口更新时间：2022-07-06 16:48:44。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：AssumeRole
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
RoleArn	是	否	String	角色的资源描述。 普通角色： qcs::cam::uin/12345678:role/4611686018427397919、 qcs::cam::uin/12345678:roleName/testRoleName 服务角色： qcs::cam::uin/12345678:role/ cloudServiceRole/4611686018427397920、 qcs::cam::uin/12345678:role/cloudServiceRoleName/ testServiceRoleName 示例值： qcs::cam::uin/909619400:roleName/ report_QCSRole
RoleSessionName	是	否	String	临时会话名称，由用户自定义名称 示例值： role_forwarded_by_capi
DurationSeconds	否	否	Uint64	指定临时证书的有效期，单位：秒，默认 7200 秒，最长可设定有效期为 43200 秒 示例值： 3600

参数名称	必选	允许NULL	类型	描述
UserUin	否	否	Uint64	云业务帐号代理用户操作其帐户资源时，传入用户uin，缺省为用户的ownerUin。其他情况无需该参数 示例值： 1000
Policy	否	否	String	策略描述注意：1、policy 需要做 urlencode（如果通过 GET 方法请求云 API，发送请求前，所有参数都需要按照云 API 规范再 urlencode 一次）。2、策略语法参照 CAM 策略语法。3、策略中不能包含 principal 元素。 示例值： policygen-20141112201913
AssumeUin	否	否	Uint64	操作子账户Uin，携带至token中，cam-auth鉴权时会返回 示例值： 1000

3. 输出参数

参数名称	类型	描述
Credentials	Credentials	临时安全证书 示例值：查看
ExpiredTime	Uint64	证书无效的时间，返回 Unix 时间戳，精确到秒 示例值： 3600
Expiration	String	证书无效的时间，以 iso8601 格式的 UTC 时间表示 示例值： 2014-10-21 08:46:55
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
ResourceNotFound.RoleNotFound	账号对应的角色不存在。
UnauthorizedOperation	未授权操作
InternalError.GetAppIdError	获取appid错误。
InternalError.SystemError	内部错误。
InvalidParameter.StrategyInvalid	非法策略。
InvalidParameter.GrantOtherResource	越权访问资源。

错误码	描述
InvalidParameter.OverTimeError	过期时间超过阈值。
InvalidParameter.ResouceError	资源错误。
InternalError.DbError	DB错误。
InternalError.EncryptError	加密失败。
InvalidParameter.AccountNotAvaliable	账号不存在或不可用。
InvalidParameter.StrategyFormatError	策略语法错误。
InvalidParameter.ExtendStrategyOverSize	扩展策略过大。
InvalidParameter.ParamError	参数错误。
InvalidParameter.PolicyTooLong	策略过长。
InternalError.GetSeedTokenError	获取种子token失败。
InvalidParameter.TempCodeNotAvaliable	TCB temp code 无效。
InvalidParameter.RoleNotExist	角色不存在

获取控制台临时证书

1. 接口描述

接口请求域名： sts.api3.finance.cloud.tencent.com。

获取控制台临时证书

默认接口请求频率限制：20次/秒。

接口更新时间：2020-03-05 17:05:07。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetConsoleSessionToken
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Skey	否	否	String	Skey 示例值： TR**********
Sid	是	否	String	sessionId的md5值 示例值： 78b83774d3a907fbea42783d58a95204
ClientUA	是	否	String	客户端UserAgent 示例值： Chrome
ClientDevice	是	否	String	客户端设备(pc或者mobile) 示例值： PC
ConSecretId	是	否	String	分配给控制台的secretId 示例值： AK**********
Noncet	是	否	Uint64	生成签名时使用 示例值： 16385182231312482027

参数名称	必选	允许NULL	类型	描述
Signaturet	是	否	String	根据签名的计算方法生成的签名 示例值： AK**********
Resource	否	否	String	操作资源 示例值： qcs::cvm:region:uin/110000001207:volume/*
Duration	是	否	Uint64	过期时间 示例值： 3600
ConsoleInfo	否	否	String	控制台信息，cos v4 控制台填写"COSV4"，其他情况不需要传入该参数 示例值： cos v4 控制台填写"COSV4"，其他情况不需要传入该参数
ClientIPt	是	否	String	客户端IP 示例值： 1.1.1.1
Actiont	是	否	String	操作接口，获取临时证书后要操作的api名称 示例值： Describe
Timestampt	是	否	Uint64	时间戳(计算签名时使用) 示例值： 1699852800

3. 输出参数

参数名称	类型	描述
Token	String	token 示例值： 142be10ace46db098d2d89a7f054ccffa619ffa310001
SecretId	String	临时证书秘钥ID 示例值： AK**********
SecretKey	String	临时证书秘钥Key 示例值： TR**********
SecretExpire	Uint64	过期时间 示例值： 3600
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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。

获取联合身份临时访问凭证

1. 接口描述

接口请求域名： sts.api3.finance.cloud.tencent.com。

获取联合身份临时访问凭证

默认接口请求频率限制：20次/秒。

接口更新时间：2022-07-05 19:59:17。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetFederationToken
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Name	是	否	String	您可以自定义调用方英文名称，由字母组成。 示例值： cam
Policy	是	否	String	策略描述注意：1、policy 需要做 urlencode（如果通过 GET 方法请求云 API，发送请求前，所有参数都需要按照云 API 规范再 urlencode 一次）。2、策略语法参照CAM 策略语法。3、策略中不能包 含 principal 元素 示例值： policygen-20141112201913
DurationSeconds	否	否	Uint64	指定临时证书的有效期，单位：秒，默认1800秒，最长可设定有效期为7200秒。 示例值： 3600
SecretType	否	否	Uint64	证书类型（0：长证书；1：短证书）。 示例值： 0

3. 输出参数

参数名称	类型	描述
Credentials	Credentials	临时证书 示例值：查看
ExpiredTime	Uint64	临时证书有效的时间，返回 Unix 时间戳，精确到秒 示例值： 3600
Expiration	String	过期时间 示例值： 2025-03-12 20:28:18
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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
InternalError.DbError	DB错误。
InternalError.EncryptError	加密失败。
InternalError.GetAppIdError	获取appid错误。
InternalError.GetSeedTokenError	获取种子token失败。
InternalError.SystemError	内部错误。
InvalidParameter.AccountNotAvaliable	账号不存在或不可用。
InvalidParameter.ExtendStrategyOverSize	扩展策略过大。
InvalidParameter.GrantOtherResource	越权访问资源。
InvalidParameter.OverTimeError	过期时间超过阈值。
InvalidParameter.ParamError	参数错误。
InvalidParameter.PolicyTooLong	策略过长。
InvalidParameter.ResouceError	资源错误。
InvalidParameter.StrategyFormatError	策略语法错误。

错误码	描述
InvalidParameter.StrategyInvalid	非法策略。
InvalidParameter.TempCodeNotAvaliable	TCB temp code 无效。
ResourceNotFound.RoleNotFound	账号对应的角色不存在。
UnauthorizedOperation	未授权操作

获取用户临时证书（第三方开发商）

1. 接口描述

接口请求域名： sts.api3.finance.cloud.tencent.com。

获取用户临时证书（第三方开发商）

默认接口请求频率限制：20次/秒。

接口更新时间：2021-06-18 14:30:13。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：GetThirdPartyFederationToken
Version	是	否	String	公共参数，本接口取值：2018-08-13
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
UserAccessToken	是	否	String	访问Token 示例值：142be10ace46db098d2d89a7f054ccffa619ffa310001
Duration	是	否	Int64	有效期 示例值： 3600
ApiAppId	是	否	Int64	该参数已废除(传0即可) 示例值： 0

3. 输出参数

参数名称	类型	描述
Credentials	Credentials	证书秘钥 示例值：查看

参数名称	类型	描述
ExpiredTime	Int64	有效期(时间戳) 示例值： 3600
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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.CodeNotSet	请求的code不存在。
FailedOperation.UserUinNotSet	用户uin不存在。
FailedOperation.UserOwnerUinNotSet	用户主账号uin不存在。
FailedOperation.OpenUinNotSet	开放平台uin不存在。
FailedOperation.OpenOwnerUinNotSet	开放拼图主账号uin不存在。
FailedOperation.GrantListNotSet	授权列表不存在。
FailedOperation.StrategyEmpty	策略空。
InternalError.SystemError	内部错误。
InvalidParameter.ParamError	参数错误。

数据结构

Credentials

临时证书

被如下接口引用：AssumeRole、GetFederationToken、GetThirdPartyFederationToken

名称	必选	允许NULL	类型	描述
Token	是	否	String	token 示例值： 142be10ace46db098d2d89a7f054ccffa619ffa310001
TmpSecretId	是	否	String	临时证书密钥ID 示例值： AK**********
TmpSecretKey	是	否	String	临时证书密钥Key 示例值： TR**********

错误码

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 cor
rect."
    },
    "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
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接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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
InternalError.SystemError	内部错误。
InvalidParameter.OverTimeError	过期时间超过阈值。
FailedOperation.OpenUinNotSet	开放平台uin不存在。
InvalidParameter.GrantOtherResource	越权访问资源。

错误码	说明
FailedOperation.OpenOwnerUinNotSet	开放拼图主账号uin不存在。
InternalError.EncryptError	加密失败。
FailedOperation.UserOwnerUinNotSet	用户主账号uin不存在。
InvalidParameter.TempCodeNotAvaliable	TCB temp code 无效。
UnauthorizedOperation	未授权操作
InvalidParameter.StrategyInvalid	非法策略。
ResourceNotFound.RoleNotFound	账号对应的角色不存在。
InvalidParameter.ExtendStrategyOverSize	扩展策略过大。
FailedOperation.GrantListNotSet	授权列表不存在。
FailedOperation.StrategyEmpty	策略空。
InvalidParameter.AccountNotAvaliable	账号不存在或不可用。
InvalidParameter.PolicyTooLong	策略过长。
InternalError.GetAppIdError	获取appid错误。
InternalError.GetSeedTokenError	获取种子token失败。
InvalidParameter.StrategyFormatError	策略语法错误。
InvalidParameter.RoleNotExist	角色不存在
FailedOperation.CodeNotSet	请求的code不存在。
InvalidParameter.ResouceError	资源错误。
InvalidParameter.ParamError	参数错误。
InternalError.DbError	DB错误。
FailedOperation.UserUinNotSet	用户uin不存在。

TCE 平台信息（location）
版本（2019-11-28）
API 概览

API版本

V3

地域相关接口

接口名称	接口功能
DescribeRegionZone	查询地域信息的统一入口
DescribeRegions	查询地域列表
DescribeZones	查询可用区列表
SetErrorRegions	设置故障地域列表

调用方式
接口签名v1

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐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。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
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

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' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    '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=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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.finance.cloud.tencent.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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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 http 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TCloudFinanceZone 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
- Node

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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

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

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

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

Java

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

public class CloudAPIDemo {
    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.finance.cloud.tencent.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.finance.cloud.tencent.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)).app
end("&");
        }
        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.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 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", "shjr"); // 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
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        '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)
```

接口签名v3

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：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 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，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。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

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

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- 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，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- 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）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 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
```

- 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 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- 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，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

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

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

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

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

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

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 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）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //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 = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\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 = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        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("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

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

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

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

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

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

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

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

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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。

公共参数

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

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	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。
X-TC-Timestamp	Integer	是	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。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	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。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

地域相关接口
查询地域信息的统一入口

1. 接口描述

接口请求域名： location.api3.finance.cloud.tencent.com。

查询地域/可用区信息. 集成: 1. 国际化; 2. 租户白名单; 3. 产品部署信息

默认接口请求频率限制：2000次/秒。

接口更新时间：2025-02-28 15:18:19。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRegionZone
Version	是	否	String	公共参数，本接口取值：2019-11-28
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
ProductId	是	否	String	TCE 产品Code 示例值： "VPC"
SubProductId	否	否	String	TCE 子产品Code 示例值： "DCGW"
Regions	否	否	Array of String	待过滤的Region列表 示例值： ["Region1"]

3. 输出参数

参数名称	类型	描述

参数名称	类型	描述
RegionCount	Int64	地域数量 示例值： 3
RegionSet	Array of RegionEx	地域信息 示例值：查看
Products	String	产品 示例值： ""
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 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。

查询地域列表

1. 接口描述

接口请求域名： location.api3.finance.cloud.tencent.com。

本接口(DescribeRegions)用于查询地域信息。

默认接口请求频率限制：2000次/秒。

接口更新时间：2024-12-26 10:46:38。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeRegions
Version	是	否	String	公共参数，本接口取值：2019-11-28
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	地域数量 示例值： 100
RegionSet	Array of RegionInfo	地域列表信息 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

查询可用区列表

1. 接口描述

接口请求域名： location.api3.finance.cloud.tencent.com。

本接口(DescribeZones)用于查询可用区信息。

默认接口请求频率限制：2000次/秒。

接口更新时间：2024-12-26 10:47:19。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：DescribeZones
Version	是	否	String	公共参数，本接口取值：2019-11-28
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	可用区数量 示例值： 10
ZoneSet	Array of ZoneInfo	可用区列表信息 示例值：查看
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

设置故障地域列表

1. 接口描述

接口请求域名： location.api3.finance.cloud.tencent.com。

故障切换设置故障的 Region 列表

默认接口请求频率限制：20次/秒。

接口更新时间：2025-01-16 20:39:44。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：SetErrorRegions
Version	是	否	String	公共参数，本接口取值：2019-11-28
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
Regions	是	否	Array of String	Region 列表 示例值： ["Region1"]
Enable	是	否	Int64	Region 状态1 正常，0 故障 示例值： 0

3. 输出参数

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

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

数据结构

ZoneEx

DescribeRegionZone 接口返回的 Zone 描述信息

被如下接口引用：DescribeRegionZone

名称	必选	允许NULL	类型	描述
RegionID	是	否	String	所属Region编号 示例值： "50000001"
Zone	是	否	String	可用区名称(固定为英文字符串, API使用) 示例值： "yf1"
ZoneID	是	否	String	可用区编号 示例值： "50010001"
ZoneName	是	否	String	可用区名称(国际化支持, 前端展示使用) 示例值： "云福1"
ZoneState	是	否	String	可用区状态(AVAILABLE/UNAVAILABLE) 示例值： "AVAILABLE"
ZoneStateRemark	是	否	String	可用区状态(原因描述) 示例值： ""
ZoneRole	否	否	String	可用区角色（MAZ:主可用区, SAZ:从可用区, OAZ:其它可用区） 示例值： "MAZ"
Idcs	否	是	Array of ZoneIDC	Idc 列表 示例值：查看

ZoneIDC

区域 IDC

被如下接口引用：DescribeRegionZone

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
IdcID	否	否	String	idc id 示例值： "60520201"
IdcName	否	否	String	idc 名称 示例值： "sztestops2-idc"
IdcUUID	否	否	String	idc uuid 示例值： "05e3d5c3f67b41fab4b86ac8b3b7deac"
IdcNameCN	否	否	String	idc 中文名称 示例值： "双AZ集成测试环境二区IDC"
ModuleID	否	否	String	产品 id 示例值： "102"

RegionEx

DescribeRegionZone 返回的 Region 描述信息

被如下接口引用：DescribeRegionZone

名称	必选	允许NULL	类型	描述
Region	是	否	String	地域名称(固定为英文字符串, API使用) 示例值： "Region1"
RegionID	是	否	String	地域编号(国际化支持, 前端展示使用) 示例值： "50000001"
RegionName	是	否	String	地域名称 示例值： "Region1"
RegionState	是	否	String	地域状态(AVAILABLE/UNAVAILABLE) 示例值： "AVAILABLE"
RegionStateRemark	是	否	String	地域状态描述 示例值： "no available zone"
ZoneCount	是	否	Int64	可用区数量 示例值： 5
ZoneSet	是	否	Array of ZoneEx	可用区列表 示例值：查看

名称	必选	允许NULL	类型	描述
RegionRole	否	否	String	地域角色(MR:主地域, SR:从地域, OR:其它地域) 示例值： "MR"
RegionArea	否	否	String	地域 示例值： "南部"

RegionInfo

地域信息

被如下接口引用：DescribeRegions

名称	必选	允许NULL	类型	描述
Region	是	否	String	地域名称，例如，ap-Region1 示例值： "Region1"
RegionName	是	否	String	地域描述，例如，xxx地区(Region1) 示例值： "Region1"
RegionState	是	否	String	地域是否可用状态 示例值： "AVAILABLE"

ZoneInfo

可用区信息

被如下接口引用：DescribeZones

名称	必选	允许NULL	类型	描述
Zone	是	否	String	可用区名称，例如，ap-Region1-3 示例值： "yf1"
ZoneName	是	否	String	可用区描述，例如，Region1三区 示例值： "云福1"
ZoneId	是	否	String	可用区ID 示例值： "50010001"
ZoneState	是	否	String	可用区ID 示例值： "AVAILABLE"

名称	必选	允许NULL	类型	描述
ZoneRole	否	否	String	Zone 角色 示例值： "MAZ"

错误码

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 cor
rect."
    },
    "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
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接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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

租户端消息服务（tcloudmsg）
版本（2020-12-07）
API 概览

API版本

V3

产品自定义通道号

接口名称	接口功能
PublicThemeAuto	PublicThemeAuto

调用方式
接口签名v1

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐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。以下是生成签名串的详细过程：

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

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

参数名称	中文	参数值
Action	方法名	DescribeInstances
SecretId	密钥Id	AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886
Region	实例所在区域	shjr
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

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' : 'shjr',
    'SecretId' : 'AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE',
    '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=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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.finance.cloud.tencent.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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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 = 'Gu5t9xGARNpq86cd98joQYCN3EXAMPLE';
$srcStr = 'GETcvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&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:

```
EliP9YW3pW28FpsEdkXt/+WcGeI=
```

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 EliP9YW3pW28FpsEdkXt/+WcGeI= ，最终得到的签名串请求参数（Signature）为：EliP9YW3pW28FpsEdkXt%2f%2bWcGeI%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 http 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TCloudFinanceZone 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
- Node

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.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=shjr&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE&Signature=EliP9YW3pW28FpsEdkXt%2F%2BWcGeI%3D&Timestamp=1465185768&Version=2017-03-12

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

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

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

Java

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

public class CloudAPIDemo {
    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.finance.cloud.tencent.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.finance.cloud.tencent.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)).app
end("&");
        }
        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.Intege
r.MAX_VALUE));
        params.put("Nonce", 11886); // 公共参数
        // 实际调用时应当使用系统当前时间，例如：  params.put("Timestamp", System.currentTimeMillis() /
1000);
        params.put("Timestamp", 1465185768); // 公共参数
        params.put("SecretId", "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"); // 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", "shjr"); // 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
        params.put("Signature", sign(getStringToSign(params), "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE
", "HmacSHA1")); // 公共参数
        System.out.println(getUrl(params));
    }
}
```

Python

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

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

```
import hashlib
import hmac
import time

import requests

secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

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.finance.cloud.tencent.com"
    data = {
        'Action' : 'DescribeInstances',
        'InstanceIds.0' : 'ins-09dx96dg',
        'Limit' : 20,
        'Nonce' : 11886,
        'Offset' : 0,
        'Region' : 'shjr',
        '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)
```

接口签名v3

TCloudFinanceZone 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TCloudFinanceZone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，点击【新建】即可以创建一对SecretId/SecretKey

注意：开发商帐号最多可以拥有两对 SecretId / SecretKey。

2. TC3-HMAC-SHA256 签名方法

注意：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 格式默认所有业务接口均支持，multipart 格式只有特定业务接口支持，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。

下面以云服务器查询广州区实例列表作为例子，分步骤介绍签名的计算过程。我们仅用到了查询实例列表的两个参数：Limit 和 Offset，使用 GET 方法调用。

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE 和 Gu5t9xGARNpq86cd98joQYCN3EXAMPLE

2.1. 拼接规范请求串

按如下格式拼接规范请求串（CanonicalRequest）：

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

- HTTPRequestMethod：HTTP 请求方法（GET、POST ），本示例中为 GET；
- 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，本示例取值为：Limit=10&Offset=0。注意：CanonicalQueryString 需要经过 URL 编码。
- CanonicalHeaders：参与签名的头部信息，至少包含 host 和 content-type 两个头部，也可加入自定义的头部参与签名以提高自身请求的唯一性和安全性。拼接规则：1）头部 key 和 value 统一转成小写，并去掉首尾空格，按照 key:value\n 格式拼接；2）多个头部，按照头部 key（小写）的字典排序进行拼接。此例中为： content-type:application/x-www-form-urlencoded\nhost:cvm.finance.cloud.tencent.com\n
- 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）按照字典排序进行拼接，并且以分号（;）分隔。此例中为： content-type;host
- HashedRequestPayload：请求正文的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(RequestPayload)))，对 HTTP 请求整个正文 payload 做 SHA256 哈希，然后十六进制编码，最后编码串转换成小写字母。注意：对于 GET 请求，RequestPayload 固定为空字符串，对于 POST 请求，RequestPayload 即为 HTTP 请求正文 payload。

根据以上规则，示例中得到的规范请求串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
GET
/
Limit=10&Offset=0
content-type:application/x-www-form-urlencoded
host:cvm.finance.cloud.tencent.com

content-type;host
e3b0c44298fc1c149afbf4c8996fb92427ae41e4649b934ca495991b7852b855
```

2.2. 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
```

- 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 X-TC-Timestamp 取值，如上示例请求为 1539084154；
- 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，例如 cvm。如上示例请求，取值为 2018-10-09/cvm/tc3_request；
- HashedCanonicalRequest：前述步骤拼接所得规范请求串的哈希值，计算方法为 Lowercase(HexEncode(Hash.SHA256(CanonicalRequest)))。

注意：

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

根据以上规则，示例中得到的待签名字符串如下（为了展示清晰，\n 换行符通过另起打印新的一行替代）：

```
TC3-HMAC-SHA256
1539084154
2018-10-09/cvm/tc3_request
91c9c192c14460df6c1ffc69e34e6c5e90708de2a6d282cccf957dbf1aa7f3a7
```

2.3. 计算签名

1）计算派生签名密钥，伪代码如下

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

- SecretKey：原始的 SecretKey；
- Date：即 Credential 中的 Date 字段信息，如上示例，为2018-10-09；
- Service：即 Credential 中的 Service 字段信息，如上示例，为 cvm；

2）计算签名，伪代码如下

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

- SecretSigning：即以上计算得到的派生签名密钥；
- StringToSign：即步骤2计算得到的待签名字符串；

2.4. 拼接 Authorization

按如下格式拼接 Authorization：

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

- Algorithm：签名方法，固定为 TC3-HMAC-SHA256；
- SecretId：密钥对中的 SecretId；
- CredentialScope：见上文，凭证范围；
- SignedHeaders：见上文，参与签名的头部信息；
- Signature：签名值

根据以上规则，示例中得到的值为：

```
TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
```

最终完整的调用信息如下：

```
https://cvm.finance.cloud.tencent.com/?Limit=10&Offset=0

Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE/2018-10-09/cvm/tc3_request, SignedHeaders=content-type;host, Signature=5da7a33f6993f0614b047e5df4582db9e9bf4672ba50567dba16c6ccf174c474
Content-Type: application/x-www-form-urlencoded
Host: cvm.finance.cloud.tencent.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1539084154
X-TC-Region: shjr
```

3. 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）

4. 签名演示

Java

```
import java.io.BufferedReader;
import java.io.InputStream;
import java.io.InputStreamReader;
import java.net.URL;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Map;
import java.util.TimeZone;
import java.util.TreeMap;
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import javax.net.ssl.HttpsURLConnection;
import javax.xml.bind.DatatypeConverter;

import org.apache.commons.codec.digest.DigestUtils;

public class CloudAPITC3Demo {
    private final static String CHARSET = "UTF-8";
    private final static String ENDPOINT = "cvm.finance.cloud.tencent.com";
    private final static String PATH = "/";
    private final static String SECRET_ID = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE";
    private final static String SECRET_KEY = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE";
    private final static String CT_X_WWW_FORM_URLENCODED = "application/x-www-form-urlencode
d";
    private final static String CT_JSON = "application/json";
```

```
    private final static String CT_FORM_DATA = "multipart/form-data";

    public static byte[] sign256(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(CHARSET));
    }

    public static void main(String[] args) throws Exception {
        String service = "cvm";
        String host = "cvm.finance.cloud.tencent.com";
        String region = "shjr";
        String action = "DescribeInstances";
        String version = "2017-03-12";
        String algorithm = "TC3-HMAC-SHA256";
        String timestamp = "1539084154";
        //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 = "GET";
        String canonicalUri = "/";
        String canonicalQueryString = "Limit=10&Offset=0";
        String canonicalHeaders = "content-type:application/x-www-form-urlencoded\n" + "host:" + host
+ "\n";
        String signedHeaders = "content-type;host";
        String hashedRequestPayload = DigestUtils.sha256Hex("");
        String canonicalRequest = httpRequestMethod + "\n" + canonicalUri + "\n" + canonicalQueryStri
ng + "\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 = DigestUtils.sha256Hex(canonicalRequest.getBytes(CHARSET));
        String stringToSign = algorithm + "\n" + timestamp + "\n" + credentialScope + "\n" + hashedCan
onicalRequest;
        System.out.println(stringToSign);

        // ************* 步骤 3：计算签名 *************
        byte[] secretDate = sign256(("TC3" + SECRET_KEY).getBytes(CHARSET), date);
        byte[] secretService = sign256(secretDate, service);
        byte[] secretSigning = sign256(secretService, "tc3_request");
```

```
        String signature = DatatypeConverter.printHexBinary(sign256(secretSigning, stringToSign)).toLow
erCase();
        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("Host", host);
        headers.put("Content-Type", CT_X_WWW_FORM_URLENCODED);
        headers.put("X-TC-Action", action);
        headers.put("X-TC-Timestamp", timestamp);
        headers.put("X-TC-Version", version);
        headers.put("X-TC-Region", region);
    }
}
```

Python

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

# 密钥参数
secret_id = "AKIDz8krbsJ5yKBZQpn74WFkmLPx3EXAMPLE"
secret_key = "Gu5t9xGARNpq86cd98joQYCN3EXAMPLE"

service = "cvm"
host = "cvm.finance.cloud.tencent.com"
endpoint = "https://" + host
region = "shjr"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
timestamp = 1539084154
date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 10, "Offset": 0}

# ************* 步骤 1：拼接规范请求串 *************
http_request_method = "GET"
canonical_uri = "/"
canonical_querystring = "Limit=10&Offset=0"
ct = "x-www-form-urlencoded"
```

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

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

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

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

# 公共参数添加到请求头部
headers = {
    "Authorization": authorization,
    "Host": host,
    "Content-Type": "application/%s" % ct,
```

```
    "X-TC-Action": action,
    "X-TC-Timestamp": str(timestamp),
    "X-TC-Version": version,
    "X-TC-Region": region,
}
```

请求结构

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

2. 通信协议

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

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

4. 字符编码

均使用UTF-8编码。

返回结果

正确返回结果

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

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

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

错误返回结果

若调用失败，其返回值示例如下为：

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

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。

- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。 Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码，下表列出了公共错误码。

错误码	错误描述
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。
AuthFailure.SignatureExpire	签名过期。
AuthFailure.SignatureFailure	签名错误。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
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。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。

错误码	错误描述
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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。

公共参数

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

签名方法 v3

使用 TC3-HMAC-SHA256 签名方法时，公共参数需要统一放到 HTTP Header 请求头部中，如下：

参数名称	类型	必选	描述
X-TC-Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
X-TC-Region	String	是	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。
X-TC-Timestamp	Integer	是	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。
X-TC-Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=AKIDEXAMPLE/Date/service/tc3_request, SignedHeaders=content-type;host, Signature=fe5f80f77d5fa3beca038a248ff027d0445342fe2855ddc963176630326f1024 其中， - TC3-HMAC-SHA256：签名方法，目前固定取该值； - Credential：签名凭证，AKIDEXAMPLE 是 SecretId；Date 是 UTC 标准时间的日期，取值需要和公共参数 X-TC-Timestamp 换算的 UTC 标准时间日期一致；service为产品名，必须与调用的产品域名一致，例如cvm； - 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。
X-TC-Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token。

签名方法 v1

使用 HmacSHA1 和 HmacSHA256 签名方法时，公共参数需要统一放到请求串中，如下

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	是	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。

参数名称	类型	必选	描述
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在云API密钥上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey ，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见接口鉴权文档。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式，目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时，才使用 HmacSHA256 算法验证签名，其他情况均使用 HmacSHA1 验证签名。
Token	String	否	临时证书所用的 Token ，需要结合临时密钥一起使用。临时密钥和 Token 需要到访问管理服务调用接口获取。长期密钥不需要 Token 。

地域列表

地域（Region）是指物理的数据中心的地理区域。TCloudFinanceZone交付验证不同地域之间完全隔离，保证不同地域间最大程度的稳定性和容错性。为了降低访问时延、提高下载速度，建议您选择最靠近您客户的地域。

您可以通过 API接口 查询地域列表 查看完成的地域列表。

产品自定义通道号 PublicThemeAuto

1. 接口描述

接口请求域名： tcloudmsg.api3.finance.cloud.tencent.com。

PublicThemeAuto

默认接口请求频率限制：20次/秒。

接口更新时间：2023-12-29 17:14:49。

接口只验签名不鉴权。

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	允许NULL	类型	描述
Action	是	否	String	公共参数，本接口取值：PublicThemeAuto
Version	是	否	String	公共参数，本接口取值：2020-12-07
Region	是	否	String	公共参数，地域信息本接口不需要传递此参数。
MessageTheme	否	否	MessageTheme	MessageTheme 示例值：查看

3. 输出参数

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

4. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

数据结构

MessageType

消息模板消息类型

被如下接口引用：PublicThemeAuto

名称	必选	允许NULL	类型	描述
MsgType	否	否	Uint64	MsgType 示例值： 204
FType	否	否	Uint64	FType 示例值： 2
Name	否	否	String	Name 示例值： Product
Desc	否	否	String	Desc 示例值： Description
Creater	否	否	String	Creater 示例值： Tom
Channels	否	否	Int64	Channels 示例值： 63
DefaultChannels	否	否	Int64	DefaultChannels 示例值： 63
DefaultStrategy	否	否	String	DefaultStrategy 示例值： 1
IsDisplay	否	否	Uint64	IsDisplay 示例值： 1
DisplayWeight	否	否	Int64	DisplayWeight 示例值： 100
InvokePattern	否	否	Uint64	InvokePattern 示例值： 1
ModifyTime	否	否	Datetime	ModifyTime 示例值： 2024-08-03T12:00:00+08:00

名称	必选	允许NULL	类型	描述
NameW	否	否	String	NameW 示例值： {"en","Name"}
DescW	否	否	String	DescW 示例值： {"en","Description"}
Station	否	否	Int64	Station 示例值： 1
CCategory	否	否	Array of CCategory	CCategory 示例值：查看

TDefineSmsChannel

TDefineSmsChannel

被如下接口引用：QueryDefineSmsChannel

名称	必选	允许NULL	类型	描述
Id	否	否	Int64	id 示例值： 1
UpdateName	否	否	String	修改人 示例值： Lucy
Channel	否	否	String	渠道 示例值： 2
CreateTime	否	否	Datetime_iso	创建时间 示例值： 2024-08-03T12:00:00+08:00
ModifyTime	否	否	Datetime_iso	修改时间 示例值： 2024-08-03T12:00:00+08:00
Status	否	否	Int64	状态 示例值： 1
ProductCode	否	否	String	产品代码 示例值： BMPPro
ProductName	否	否	String	产品名 示例值： Tcenter

MessageTheme

消息模板

被如下接口引用：PublicThemeAuto

名称	必选	允许NULL	类型	描述
ThemeId	否	否	Uint64	ThemeId 示例值： 1000
ThemeTitle	否	否	String	ThemeTitle 示例值： Title
Priority	否	否	Int64	Priority 示例值： 1
MsgType	否	否	Uint64	MsgType 示例值： 214
IsTemplate	否	否	Uint64	IsTemplate 示例值： 1
Title	否	否	String	Title 示例值： Title
SiteMsgContent	否	否	String	SiteMsgContent 示例值： Content
MailContent	否	否	String	MailContent 示例值： Content
SMSContent	否	否	String	SMSContent 示例值： Content
IsMpboss	否	否	Uint64	IsMpboss 示例值： 1
MpbossId	否	否	Int64	MpbossId 示例值： 1
Creater	否	否	String	Creater 示例值： Tom
Admin	否	否	String	Admin 示例值： Lucy
ProductId	否	否	Uint64	ProductId 示例值： 1

名称	必选	允许NULL	类型	描述
SendChannel	否	否	Int64	SendChannel 示例值： 2
CreateTime	否	否	Datetime	CreateTime 示例值： 2024-08-03T12:00:00+08:00
ModifyTime	否	否	Datetime	ModifyTime 示例值： 2024-08-03T12:00:00+08:00
IsImportant	否	否	Int64	IsImportant 示例值： 1
SmsType	否	否	Int64	SmsType 示例值： 0
TitleUinPosition	否	否	Int64	TitleUinPosition 示例值： 1
SiteMsgUinPosition	否	否	Int64	SiteMsgUinPosition 示例值： 1
MailUinPosition	否	否	Int64	MailUinPosition 示例值： 1
SmsUinPosition	否	否	Int64	SmsUinPosition 示例值： 1
TitleW	否	否	String	TitleW 示例值： {"en","Title"}
SiteMsgContentW	否	否	String	SiteMsgContentW 示例值： {"en","Content"}
MailContentW	否	否	String	MailContentW 示例值： {"en","Content"}
SMSContentW	否	否	String	SMSContentW 示例值： {"en","Content"}
MpbossIdW	否	否	String	MpbossIdW 示例值： {"en":0,"zh":0}
SiteSub	否	否	String	SiteSub 示例值： 0

名称	必选	允许NULL	类型	描述
MailSubW	否	否	String	MailSubW 示例值： {"en":0,"zh":0}
SmsSendType	否	否	Int64	SmsSendType 示例值： 1
PreSmsContent	否	否	String	PreSmsContent 示例值： Content
SmsExtend	否	否	String	SmsExtend 示例值： SmsExtend
MailType	否	否	Int64	MailType 示例值： 1
WechatSendType	否	否	String	WechatSendType 示例值： 0
WechatContentW	否	否	String	WechatContentW 示例值： {"en":0,"zh":0}
Status	否	否	Int64	Status 示例值： 1
ProductCode	否	否	String	ProductCode 示例值： BMPPro
ProductName	否	否	String	ProductName 示例值： Tcenter
ThemeTitleW	否	否	String	ThemeTitleW 示例值： {"en","Title"}
QywxContent	否	否	String	QywxContent 示例值： Content
QywxContentW	否	否	String	QywxContentW 示例值： {"en","Content"}
VoiceContent	否	否	String	VoiceContent 示例值： Content
VoiceContentW	否	否	String	VoiceContentW 示例值： {"en","Content"}
MessageType	否	否	Array of MessageType	MessageType 示例值：查看

FilterString

参数

被如下接口引用：QueryDefineSmsChannel

名称	必选	允许NULL	类型	描述
Name	是	否	String	key过滤字段 示例值： Product
Value	是	否	String	value过滤字段的值 示例值： BMPro

CCategory

CCategory

被如下接口引用：PublicThemeAuto

名称	必选	允许NULL	类型	描述
CategoryID	否	否	Uint64	CategoryID 示例值： 1
CategoryName	否	否	String	CategoryName 示例值： Name
CategoryNameW	否	否	String	CategoryNameW 示例值： {"en","Name"}
DisplayWeight	否	否	Uint64	DisplayWeight 示例值： 100

错误码

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 cor
rect."
        },
        "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
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接口鉴权文档检查签名计算过程。
AuthFailure.TokenFailure	token 错误。
AuthFailure.UnauthorizedOperation	请求未 CAM 授权。
DryRunOperation	DryRun 操作，代表请求将会是成功的，只是多传了 DryRun 参数。

错误码	说明
FailedOperation	操作失败。
InternalError	内部错误。
InvalidAction	接口不存在。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误。
UnsupportedOperation	操作不支持。
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