基础平台

产品文档

版权所有：腾讯云金融专区

文档目录

- 操作指南
  - 登录
  - 帐号信息
  - 账号注销
  - 主机安全评分
  - 安全设置
  - 资源管理
  - 备案管理
  - 常见登录问题
- 登录
- 帐号信息
- 账号注销
  - 账号注销
- 主机安全评分
  - 主机安全评分
- 安全设置
- 资源管理
- 备案管理
  - 备案管理
- 常见登录问题
  - 常见登录问题
- API文档
  - platform（platform）
    - 版本（2019-03-14）
      - API概览
      - 调用方式
        - 接口签名v1
        - 接口签名v3
        - 请求结构
        - 返回结果
        - 公共参数
      - platform
        - 取消微信通知
        - 获取租户端主账号
        - 获取租户端子账号
        - 查询是否开启微信通道
        - 拉取验证二维码
        - 更换微信
        - 不接收微信通知
        - 查询租户端账号
        - 发送二维码绑定邮件
      - 数据结构
      - 错误码
  - 密钥服务（sts）
    - 版本（2018-08-13）
      - API概览
      - 调用方式
        - 接口签名v1
        - 接口签名v3
        - 请求结构
        - 返回结果
        - 公共参数
      - 秘钥相关接口
        - 申请扮演角色

获取联合身份临时访问凭证
获取用户临时证书（第三方开发商）
数据结构
错误码
组织与项目（tpo）
版本（2020-09-20）
API概览
调用方式
接口签名v1
接口签名v3
请求结构
返回结果
公共参数
Project相关接口
项目添加用户及授权
添加项目配额
项目添加资源
批量导入配额
检查配额合法性
创建项目
删除项目
删除项目配额
项目移出资源
查询各级产品
查询用户的项目策略列表
查询项目成员
查询当前租户下所有非项目成员
查询项目所有策略列表
查询项目配额
查询项目资源列表
查询项目列表
查询具有转入转出资源权限的项目列表
查询所有资源区域
修改单个用户的授权策略
修改项目名称
修改项目配额
项目间移动资源
项目名是否唯一
从项目移除成员
转入转出资源
数据结构
错误码
账号系统（account）
版本（2019-03-25）
API概览
调用方式
接口签名v1
接口签名v3
请求结构
返回结果
公共参数
账号系统
获取多因子登录参数
数据结构
错误码
TCE 平台信息（location）

- 版本（2019-11-28）
  - API概览
  - 调用方式
    - 接口签名v1
    - 接口签名v3
    - 请求结构
    - 返回结果
    - 公共参数
  - 地域相关接口
    - 查询地域信息的统一入口
    - 查询地域列表
    - 查询可用区列表
  - 数据结构
  - 错误码
- 开放平台（open）
  - 版本（2020-12-02）
    - API概览
    - 调用方式
      - 接口签名v1
      - 接口签名v3
      - 请求结构
      - 返回结果
      - 公共参数
    - open相关接口
      - CreateLdapIdp
      - GetLdapIdpConfig
      - ListIdentityProvider
      - TestLdap
      - UpdateLdapIdp
    - 数据结构
    - 错误码
- 平台产品中心（tag）
  - 版本（2018-08-13）
    - API概览
    - 调用方式
      - 接口签名v1
      - 接口签名v3
      - 请求结构
      - 返回结果
      - 公共参数
    - 写接口
      - 标签关联资源
      - 创建标签
      - 标签解绑资源
      - 删除标签
      - 批量修改资源关联的标签
    - 读接口
      - 查询资源关联标签
      - 查看资源关联的标签
      - 查询标签列表
    - 数据结构
    - 错误码
- 访问管理（cam）
  - 版本（2019-01-16）
    - API概览
    - 调用方式

- 接口签名v1
- 接口签名v3
- 请求结构
- 返回结果
- 公共参数

其他接口
- 绑定多个策略到角色
- 绑定权限策略到角色
- 绑定多个角色到策略
- 创建策略
- 创建角色
- 删除策略
- 删除角色
- 获取角色列表
- 解除绑定多个策略到用户组
- 解除绑定策略到多个用户组
- 解除绑定策略到多个用户
- 查看策略详情
- 获取角色详情
- 获取服务及其API列表
- 获取服务角色信息
- 查询用户组关联的策略列表
- 获取角色绑定的策略列表
- 查询策略关联的实体列表
- 查询策略列表
- 修改角色信任策略
- 更新策略

用户相关接口
- 获取CAM密码规则
- 更新CAM密码规则

身份提供商接口
- 新增oauth配置
- 获取用户oauth标识
- 刷新用户userAccessToken
- 更新Oauth配置信息
- 验证用户userAccessToken

数据结构

错误码

操作指南
登录

最近更新时间: 2025-10-11 18:10:00

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 。

版权所有：腾讯云金融专区

帐号信息

最近更新时间: 2025-10-11 18:10:00

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、APPID、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。

版权所有：腾讯云金融专区

账号注销

最近更新时间: 2025-10-11 18:10:00

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

最近更新时间: 2025-10-11 18:10:00

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。

安全设置

最近更新时间: 2025-10-11 18:10:00

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，修改手机号码，修改邮箱。

- 修改密码：修改时必须输入旧密码才能修改，输入两次密码必须相同，密码规则可通过主账号在用户设置中进行设置，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：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：旧手机号码不在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一致，只需要输入新的邮箱号与手机号码；修改邮箱号码使用场景：邮箱号码不在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

用户在安全设置中设置登录保护类型，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 topt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认真工具，用户需要先将账号绑定到三方系统并下载三方系统的客户端，在登录时输入客户端中的口令完成身份认证。

资源管理

最近更新时间: 2025-10-11 18:10:00

单击**云产品**>**管理与审计**>**资源管理**进入资源管理控制台。

项目管理

新建项目

1. 单击**项目管理**>**新建**，新建项目。

2. 在弹出的窗口中输入项目名称，单击**确认**。

3. 在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**操作，在弹出框中点击**确认**，再次弹出删除成功，单击**确认**，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**，出现弹窗，可针对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对其进行**管理**操作。

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**。弹出转出资源成功弹窗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

最近更新时间: 2025-10-11 18:10:00

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。

常见登录问题

最近更新时间: 2025-10-11 18:10:00

常见登录问题

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

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。

登录

最近更新时间: 2025-10-11 18:10:00

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 。

帐号信息

最近更新时间: 2025-10-11 18:10:00

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、APPID、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

账号注销

最近更新时间: 2025-10-11 18:10:00

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

主机安全评分

最近更新时间: 2025-10-11 18:10:00

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。

安全设置

最近更新时间: 2025-10-11 18:10:00

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，修改手机号码，修改邮箱。

- 修改密码：修改时必须输入旧密码才能修改，输入两次密码必须相同，密码规则可通过主账号在用户设置中进行设置，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：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：旧手机号码不在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一致，只需要输入新的邮箱号与手机号码；修改邮箱号码使用场景：邮箱号码不在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

用户在安全设置中设置登录保护类型，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 topt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认真工具，用户需要先将账号绑定到三方系统并下载三方系统的客户端，在登录时输入客户端中的口令完成身份认证。

资源管理

最近更新时间: 2025-10-11 18:10:00

单击**云产品**>**管理与审计**>**资源管理**进入资源管理控制台。

项目管理

新建项目

1. 单击**项目管理**>**新建**，新建项目。

2. 在弹出的窗口中输入项目名称，单击**确认**。

3. 在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**操作，在弹出框中点击**确认**，再次弹出删除成功，单击**确认**，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**，出现弹窗，可针对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对其进行**管理**操作。

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**。弹出转出资源成功弹窗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

备案管理

最近更新时间: 2025-10-11 18:10:00

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。

常见登录问题

常见登录问题

最近更新时间: 2025-10-11 18:10:00

常见登录问题

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

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。

API文档
platform（platform）
版本（2019-03-14）
API概览

最近更新时间: 2024-10-18 10:38:22

API版本

V3

platform

接口名称	接口功能
CancelWeChatNotice	取消微信通知
GetCustomAccount	获取租户端主账号
GetCustomSubAccount	获取租户端子账号
GetSwitchInfo	查询是否开启微信通道
GetVerifyQRCode	拉取验证二维码
ModifyWeChat	更换微信
NoAcceptNotice	不接收微信通知
QueryCustomAccount	查询租户端账号
SendBindingEmail	发送二维码绑定邮件

调用方式
接口签名v1

最近更新时间: 2024-10-18 10:38:22

tcecloud 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tcecloud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
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	ap-guangzhou
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

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

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : '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=ap-guangzhou&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=ap-guangzhou&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=ap-guangzhou&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tcecloud 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
- JavaScript
- .NET

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： `http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&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 TceCloudAPIDemo {
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("http://imgcache.finance.cloud.tencent.com:80cvm.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)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如： params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
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", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
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' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
# resp = requests.get("http://imgcache.finance.cloud.tencent.com:80" + endpoint, params=data)
# print(resp.url)
```

接口签名v3

最近更新时间: 2024-10-18 10:38:22

tcecloud 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tcecloud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：

```
http://imgcache.finance.cloud.tencent.com:80cvm.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: ap-guangzhou
```

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 TceCloudAPITC3Demo {
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-urlencoded";
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 = "ap-guangzhou";
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" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

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

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

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("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 = "http://imgcache.finance.cloud.tencent.com:80" + host
region = "ap-guangzhou"
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

最近更新时间: 2024-10-18 10:38:22

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。tcecloud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

tcecloud 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

最近更新时间: 2024-10-18 10:38:22

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 correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码 (TODO: 重复信息, 是否真的需要?)

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。

错误码	错误描述
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。

公共参数

最近更新时间: 2024-10-18 10:38:22

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。
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。tcecloud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

取消微信通知

最近更新时间: 2024-10-18 10:38:22

1. 接口描述

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

取消微信通知，会发验证二维码邮件

默认接口请求频率限制：20次/秒。

接口更新时间：2019-09-10 10:03:07。

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：CancelWeChatNotice
Version	是	否	String	公共参数，本接口取值：2019-03-14
Region	是	否	String	公共参数，本接口不需要传递此参数。
UserUin	是	否	Uint64	用户Uin
UserId	是	否	String	用户登录账号
UserName	是	否	String	用户昵称

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。

获取租户端主账号

最近更新时间: 2024-10-18 10:38:22

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：2019-09-30 10:59:03。

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，本接口不需要传递此参数。
PageNum	是	否	Int64	分页
PageSize	是	否	Int64	页面数量
GetSecretField	否	否	Bool	是否获取加密key数据
SortKey	否	否	String	排序key
SortTurn	否	否	Int64	排序顺序

3. 输出参数

参数名称	类型	描述
AccountList	AccountList	用户列表
TotalNum	Int64	用户数量
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

最近更新时间: 2024-10-18 10:38:22

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，本接口不需要传递此参数。
OwnerUin	是	否	String	主账号uin
GetSecretField	否	否	Bool	是否获取加密数据

3. 输出参数

参数名称	类型	描述
SubAccountList	AccountList	用户列表
TotalNum	Int64	用户数量
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。

查询是否开启微信通道

最近更新时间: 2024-10-18 10:38:22

1. 接口描述

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

查询是否开启微信通道

默认接口请求频率限制：20次/秒。

接口更新时间：2019-09-10 10:17:34。

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：GetSwitchInfo
Version	是	否	String	公共参数，本接口取值：2019-03-14
Region	是	否	String	公共参数，本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
Status	String	查询是否开启微信通道，值为False或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

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。

拉取验证二维码

最近更新时间: 2024-10-18 10:38:22

1. 接口描述

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

拉取验证二维码

默认接口请求频率限制：20次/秒。

接口更新时间：2019-09-16 20:18:53。

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：GetVerifyQRCode
Version	是	否	String	公共参数，本接口取值：2019-03-14
Region	是	否	String	公共参数，本接口不需要传递此参数。
User_Uin	是	否	Uint64	用户Uin
User_Id	是	否	String	用户登录账号
User_Name	是	否	String	用户昵称

3. 输出参数

参数名称	类型	描述
QRCodeUrl	String	验证二维码url
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。

更换微信

最近更新时间: 2024-10-18 10:38:22

1. 接口描述

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

更换微信，会重发绑定邮件

默认接口请求频率限制：20次/秒。

接口更新时间：2019-09-16 20:18:20。

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：ModifyWeChat
Version	是	否	String	公共参数，本接口取值：2019-03-14
Region	是	否	String	公共参数，本接口不需要传递此参数。
User_Uin	是	否	Uint64	用户Uin
User_Id	是	否	String	用户登录账号
User_Name	是	否	String	用户昵称

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。

不接收微信通知

最近更新时间: 2024-10-18 10:38:22

1. 接口描述

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

不接收微信通知

默认接口请求频率限制：20次/秒。

接口更新时间：2019-09-16 20:19:08。

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：NoAcceptNotice
Version	是	否	String	公共参数，本接口取值：2019-03-14
Region	是	否	String	公共参数，本接口不需要传递此参数。
User_Uin	是	否	Uint64	用户Uin

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。

查询租户端账号

最近更新时间: 2024-10-18 10:38:22

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，本接口不需要传递此参数。
Account	否	否	String	租户端登录名
PhoneNumber	否	否	Int64	租户手机号码
NickName	否	否	String	租户昵称
CustomAppId	否	否	Int64	租户端appid
PageNum	是	否	Int64	分页
PageSize	是	否	Int64	分页大小
CustomUin	否	否	Int64	账号uin
SortKey	否	否	String	排序key
SortTurn	否	否	Int64	顺序或者逆序，0：顺序，1：逆序
StartLastLoginTime	否	否	Int64	开始时间
EndLastLoginTime	否	否	Int64	结束时间

3. 输出参数

参数名称	类型	描述
TotalNum	Int64	用户数量
AccountList	AccountList	用户列表
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。

发送二维码绑定邮件

最近更新时间: 2024-10-18 10:38:22

1. 接口描述

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

发送二维码绑定邮件

默认接口请求频率限制：20次/秒。

接口更新时间：2019-09-16 20:14:08。

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：SendBindingEmail
Version	是	否	String	公共参数，本接口取值：2019-03-14
Region	是	否	String	公共参数，本接口不需要传递此参数。
User_Uin	是	否	Uint64	用户Uin
User_Id	是	否	String	用户登录账号
User_Name	是	否	String	用户昵称

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

最近更新时间: 2024-10-18 10:38:22

ServerDevice

设备信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
DeviceId	是	否	String	设备id
Password	是	否	String	密码
IntranetIp	否	否	String	设备ip

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
AddTimestamp	是	否	String	新增日期
AppId	是	否	Int64	APPID
BindStatus	是	否	Int64	绑定状态
LastLoginTime	是	否	String	上次登录时间
Mail	是	否	String	邮件
ModTimestamp	是	否	String	修改日期
NickName	是	否	String	昵称
OwnerUin	是	否	Int64	输主uin
PhoneNumber	是	否	String	手机号码
Remark	是	否	String	标记
Status	是	否	Int64	状态
SubCount	是	否	Int64	子账号数量
Uid	是	否	Int64	uid
Uin	是	否	Int64	uin
UinType	是	否	Int64	uintype
UserName	是	否	String	username
IdKeys	是	否	Array of IdKeys	秘钥
Password	是	否	String	密码

Filter

过滤条件

被如下接口引用：

名称	必选	允许NULL	类型	描述
Name	是	否	String	名称
Value	是	否	String	值

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
SecretId	是	否	String	秘钥id
SecretKey	是	否	String	秘钥key
Source	是	否	Int64	source
Status	是	否	String	启用状态

错误码

最近更新时间: 2024-10-18 10:38:22

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

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

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
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。
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。

错误码	说明
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

最近更新时间: 2024-10-18 10:38:23

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
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

最近更新时间: 2024-10-18 10:38:23

tcecloud 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tcecloud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
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	ap-guangzhou
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

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

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : '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=ap-guangzhou&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=ap-guangzhou&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=ap-guangzhou&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tcecloud 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
- JavaScript
- .NET

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： `http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&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 TceCloudAPIDemo {
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("http://imgcache.finance.cloud.tencent.com:80cvm.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)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如： params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
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", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
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' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
# resp = requests.get("http://imgcache.finance.cloud.tencent.com:80" + endpoint, params=data)
# print(resp.url)
```

接口签名v3

最近更新时间: 2024-10-18 10:38:23

tcecloud 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tcecloud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：

```
http://imgcache.finance.cloud.tencent.com:80cvm.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: ap-guangzhou
```

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 TceCloudAPITC3Demo {
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-urlencoded";
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 = "ap-guangzhou";
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" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

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

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

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("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 = "http://imgcache.finance.cloud.tencent.com:80" + host
region = "ap-guangzhou"
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

最近更新时间: 2024-10-18 10:38:23

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。tcecloud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

tcecloud 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

最近更新时间: 2024-10-18 10:38:23

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 correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码 (TODO: 重复信息, 是否真的需要?)

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。

错误码	错误描述
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。

公共参数

最近更新时间: 2024-10-18 10:38:23

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。
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。tcecloud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

最近更新时间: 2024-10-18 10:38:24

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，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/tencentcloudServiceRole/4611686018427397920、qcs::cam::uin/12345678:role/tencentcloudServiceRoleName/testServiceRoleName
RoleSessionName	是	否	String	临时会话名称，由用户自定义名称
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
UserUin	否	否	Uint64	腾讯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
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。
AssumeUin	否	否	Uint64	操作子账户Uin，携带至token中，cam-auth鉴权时会返回

3. 输出参数

参数名称	类型	描述
Credentials	Credentials	**此参数对外不可见。** 临时安全证书
ExpiredTime	Uint64	**此参数对外不可见。** 证书无效的时间，返回 Unix 时间戳，精确到秒
Expiration	String	**此参数对外不可见。** 证书无效的时间，以 iso8601 格式的 UTC 时间表示
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	
UnauthorizedOperation	
InternalError.GetAppIdError	
InternalError.SystemError	
InvalidParameter.StrategyInvalid	
InvalidParameter.GrantOtherResource	
InvalidParameter.OverTimeError	
InvalidParameter.ResouceError	
InternalError.DbError	
InternalError.EncryptError	
InvalidParameter.AccountNotAvaliable	
InvalidParameter.StrategyFormatError	
InvalidParameter.ExtendStrategyOverSize	
InvalidParameter.ParamError	
InvalidParameter.PolicyTooLong	
InternalError.GetSeedTokenError	
InvalidParameter.TempCodeNotAvaliable	

获取联合身份临时访问凭证

最近更新时间: 2024-10-18 10:38:24

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，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。
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
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。
SecretType	否	否	Uint64	证书类型（0：长证书；1：短证书）。

3. 输出参数

参数名称	类型	描述
Credentials	Credentials	**此参数对外不可见。** 临时证书
ExpiredTime	Uint64	**此参数对外不可见。** 临时证书有效的时间，返回 Unix 时间戳，精确到秒
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	
InternalError.EncryptError	
InternalError.GetAppIdError	
InternalError.GetSeedTokenError	

错误码	描述
InternalError.SystemError	
InvalidParameter.AccountNotAvaliable	
InvalidParameter.ExtendStrategyOverSize	
InvalidParameter.GrantOtherResource	
InvalidParameter.OverTimeError	
InvalidParameter.ParamError	
InvalidParameter.PolicyTooLong	
InvalidParameter.ResouceError	
InvalidParameter.StrategyFormatError	
InvalidParameter.StrategyInvalid	
InvalidParameter.TempCodeNotAvaliable	
ResourceNotFound.RoleNotFound	
UnauthorizedOperation	

获取用户临时证书（第三方开发商）

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
UserAccessToken	是	否	String	访问Token
Duration	是	否	Int64	有效期
ApiAppId	是	否	Int64	该参数已废除(传0即可)

3. 输出参数

参数名称	类型	描述
Credentials	Credentials	**此参数对外不可见。** 证书秘钥
ExpiredTime	Int64	**此参数对外不可见。** 有效期(时间戳)
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	
FailedOperation.UserUinNotSet	
FailedOperation.UserOwnerUinNotSet	
FailedOperation.OpenUinNotSet	
FailedOperation.OpenOwnerUinNotSet	
FailedOperation.GrantListNotSet	

错误码	描述
FailedOperation.StrategyEmpty	
InternalError.SystemError	
InvalidParameter.ParamError	

数据结构

最近更新时间: 2024-10-18 10:38:24

ApiKey

API密钥数据列表

被如下接口引用：CreateSelfApiKey、GetSelfApiKey、QuerySelfApiKey

名称	必选	允许NULL	类型	描述
SecretId	否	否	String	密钥ID
CreateTime	否	否	Uint64	创建时间(时间戳)
Status	否	否	Uint64	状态(2:有效, 3:禁用)
Remark	否	是	String	备注
Source	否	是	Uint64	数据源

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
TmpSecretId	是	否	String	临时证书密钥ID
TmpSecretKey	是	否	String	临时证书密钥Key

错误码

最近更新时间: 2024-10-18 10:38:24

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

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

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
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。
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。

错误码	说明
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.OpenUinNotSet	
InvalidParameter.GrantOtherResource	
FailedOperation.Accesskey	
UnauthorizedOperation	
InvalidParameter.StrategyFormatError	
ResourceNotFound.RoleNotFound	
FailedOperation.GrantListNotSet	
InvalidParameter.ExtendStrategyOverSize	
InvalidParameter.TempCodeNotAvaliable	
InternalError.GetSeedTokenError	
FailedOperation.OpenOwnerUinNotSet	
InvalidParameter.ResouceError	
InvalidParameter.OverTimeError	
InvalidParameter.AccountNotAvaliable	
FailedOperation.UserOwnerUinNotSet	
FailedOperation.CodeNotSet	
InvalidParameter.PolicyTooLong	
InternalError.SystemError	
InternalError.DbError	
FailedOperation.UserUinNotSet	
InternalError.EncryptError	
InvalidParameter.StrategyInvalid	
InternalError.GetAppIdError	
FailedOperation.StrategyEmpty	
InvalidParameter.ParamError	

组织与项目（tpo）
版本（2020-09-20）
API概览

最近更新时间: 2024-10-18 10:38:24

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
BatchAddProjectQuota	批量导入配额
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
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
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
TransferProjectResource	转入转出资源

调用方式
接口签名v1

最近更新时间: 2024-10-18 10:38:24

tcecloud 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tcecloud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
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	ap-guangzhou
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

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

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : '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=ap-guangzhou&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=ap-guangzhou&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=ap-guangzhou&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tcecloud 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
- JavaScript
- .NET

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： `http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&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 TceCloudAPIDemo {
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("http://imgcache.finance.cloud.tencent.com:80cvm.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)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如： params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
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", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
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' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
# resp = requests.get("http://imgcache.finance.cloud.tencent.com:80" + endpoint, params=data)
# print(resp.url)
```

接口签名v3

最近更新时间: 2024-10-18 10:38:24

tcecloud 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tcecloud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：

```
http://imgcache.finance.cloud.tencent.com:80cvm.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: ap-guangzhou
```

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 TceCloudAPITC3Demo {
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-urlencoded";
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 = "ap-guangzhou";
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" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

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

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

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("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 = "http://imgcache.finance.cloud.tencent.com:80" + host
region = "ap-guangzhou"
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

最近更新时间: 2024-10-18 10:38:24

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。tcecloud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

tcecloud 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

最近更新时间: 2024-10-18 10:38:24

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 correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码 (TODO: 重复信息, 是否真的需要?)

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。

错误码	错误描述
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。

公共参数

最近更新时间: 2024-10-18 10:38:24

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。
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。tcecloud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

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
Uins	是	否	Array of Uint64	用户ID列表
PolicyNames	是	否	Array of String	待添加的策略名

3. 输出参数

参数名称	类型	描述
SuccessfulUins	UinPolicy	**此参数对外不可见。** 成功列表
FailedUins	UinPolicy	**此参数对外不可见。** 失败列表
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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
InternalError.DatabaseError	
ResourceNotFound.ProjectNotFoundError	
UnauthorizedOperation.AuthFailedError	

添加项目配额

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
ProductCode	是	否	String	一级产品定义
SubProductCode	否	否	String	二级产品定义
BillingItemCode	否	否	String	三级产品定义
SubBillingItemCode	否	否	String	四级产品定义
ProductQuota	否	否	Uint64	一级产品配额
SubProductQuota	否	否	Uint64	二级产品配额
BillingItemQuota	否	否	Uint64	三级产品配额
SubBillingItemQuota	否	否	Uint64	四级产品配额

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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	

错误码	描述
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	
InvalidParameter.InvalidProjectQuota	

项目添加资源

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
ResourceList	是	否	Array of TransferResource	添加的资源列表

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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	
InvalidParameter.UnsupportedProductCodeError	
ResourceNotFound.ProjectResourceNotFound	

批量导入配额

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
QuotaSet	是	否	Array of AddQuota	待添加的配额

3. 输出参数

参数名称	类型	描述
AddList	AddQuota	**此参数对外不可见。** 添加列表
AddSuccess	Bool	**此参数对外不可见。** 是否添加成功
UpdateList	AddQuota	**此参数对外不可见。** 更新列表
UpdateSuccess	Bool	**此参数对外不可见。** 是否更新成功
ErrorList	AddQuota	**此参数对外不可见。** 错误列表
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	
InternalError.ComponentFail	

错误码	描述
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	

检查配额合法性

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
QuotaSet	否	否	Array of QuotaNames	配额名

3. 输出参数

参数名称	类型	描述
QuotaSet	QuotaChecking	**此参数对外不可见。** 配额列表
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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	

创建项目

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectName	是	否	String	项目名
Organization	否	否	String	归属组织
ProjectDescription	否	否	String	项目描述

3. 输出参数

参数名称	类型	描述
ProjectId	String	**此参数对外不可见。** 项目ID
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	
InvalidParameter.EmptyParameter	
InternalError.ComponentFail	

删除项目

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID

3. 输出参数

参数名称	类型	描述
ProjectId	String	**此参数对外不可见。** 项目ID
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	
InternalError.ComponentFail	
InternalError.DatabaseError	
ResourceNotFound.ProjectNotFoundError	
FailedOperation.ProjectResourceNotEmpty	
FailedOperation.ProjectQuotaNotEmpty	

删除项目配额

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
ResourceList	是	否	Array of DeleteQuota	资源列表

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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	

项目移出资源

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
ResourceList	是	否	Array of TransferResource	资源列表

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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	
InvalidParameter.UnsupportedProductCodeError	
FailedOperation.ProjectCountError	
ResourceNotFound.ProjectResourceNotFound	

查询各级产品

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProductCode	否	否	String	一级产品
SubProductCode	否	否	String	二级产品
BillingItemCode	否	否	String	三级产品
SubBillingItemCode	否	否	String	四级产品
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

3. 输出参数

参数名称	类型	描述
ProductSet	Product	**此参数对外不可见。** 一级产品列表
SubProductSet	Product	**此参数对外不可见。** 二级产品列表
BillingItemSet	Product	**此参数对外不可见。** 三级产品列表
SubBillingItemSet	Product	**此参数对外不可见。** 四级产品列表
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	

错误码	描述
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	

查询用户的项目策略列表

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
AccountUin	是	否	Uint64	用户Uin
Filter	否	否	PolicyFilter	过滤参数

3. 输出参数

参数名称	类型	描述
OwnedPolicies	Policy	**此参数对外不可见。** 已添加策略
Policies	Policy	**此参数对外不可见。** 未添加策略
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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	

查询项目成员

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
PageNumber	否	否	Uint64	页码
PageSize	否	否	Uint64	每页数量
Filter	否	否	ProjectMemberFilter	搜索条件

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	**此参数对外不可见。** 总数
MemberSet	ProjectMember	**此参数对外不可见。** 成员列表
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	
InternalError.ComponentFail	
InternalError.DatabaseError	
ResourceNotFound.ProjectNotFoundError	
UnauthorizedOperation.AuthFailedError	

查询当前租户下所有非项目成员

最近更新时间: 2024-10-18 10:38:24

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：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：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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
PageNumber	否	否	Uint64	页码
PageSize	否	否	Uint64	每页数量
Filter	否	否	ProjectMemberFilter	搜索参数

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	**此参数对外不可见。** 总数
MemberSet	ProjectMember	**此参数对外不可见。** 成员列表
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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	

查询项目所有策略列表

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
PageNumber	否	否	Uint64	页码
PageSize	否	否	Uint64	每页数量
Filter	否	否	PolicyFilter	模糊查询

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	**此参数对外不可见。** 总数
PolicySet	Policy	**此参数对外不可见。** 策略列表
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	

查询项目配额

最近更新时间: 2024-10-18 10:38:24

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：2023-03-07 11:20:20。

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
PageNumber	否	否	Uint64	页码
PageSize	否	否	Uint64	每页数量
Filter	否	否	ProjectQuotaFilter	搜索条件

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	**此参数对外不可见。** 总数
QuotaSet	ProjectQuota	**此参数对外不可见。** 配额列表
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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	

错误码	描述
UnauthorizedOperation.AuthFailedError	

查询项目资源列表

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
PageNumber	否	否	Uint64	页码
PageSize	否	否	Uint64	每页数量
Filter	否	否	ProjectResourceFilter	搜索条件

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	总数
ResourceSet	ProjectResource	资源列表
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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	

查询项目列表

最近更新时间: 2024-10-18 10:38:24

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，本接口不需要传递此参数。
PageNumber	否	否	Uint64	页码
PageSize	否	否	Uint64	每页数量
Filter	否	否	DescribeProjectsFilter	过滤条件

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	**此参数对外不可见。** 总数
ProjectSet	Project	**此参数对外不可见。** 项目列表
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	
InternalError.ComponentFail	
InternalError.DatabaseError	

查询具有转入转出资源权限的项目列表

最近更新时间: 2024-10-18 10:38:24

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，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	**此参数对外不可见。** 总数
ProjectSet	ResourceAdminProject	**此参数对外不可见。** 项目列表
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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	

查询所有资源区域

最近更新时间: 2024-10-18 10:38:24

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，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	Region	**此参数对外不可见。** 地域列表
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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	

修改单个用户的授权策略

最近更新时间: 2024-10-18 10:38:25

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
AccountUin	是	否	Uint64	用户UID
PolicyNames	是	否	Array of String	待添加的策略名列表

3. 输出参数

参数名称	类型	描述
PolicyNames	String	**此参数对外不可见。** 策略名称列表
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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	

修改项目名称

最近更新时间: 2024-10-18 10:38:25

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目id
ProjectName	是	否	String	项目名称
ProjectDescription	否	否	String	项目描述

3. 输出参数

参数名称	类型	描述
ProjectId	String	**此参数对外不可见。** 项目id
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	
ResourceNotFound.ProjectNotFoundError	
InvalidParameter.ProjectNameTooLong	
InternalError.ComponentFail	
InvalidParameter.EmptyParameter	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	

修改项目配额

最近更新时间: 2024-10-18 10:38:25

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
ProductCode	是	否	String	一级产品定义
QuotaKey	是	否	String	配额键值
QuotaValue	是	否	String	配额数

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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
InvalidParameter.EmptyParameter	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
InvalidParameter.InvalidParameter	
InvalidParameter.UsedQuotaNotEnough	

错误码	描述
UnauthorizedOperation.AuthFailedError	

项目间移动资源

最近更新时间: 2024-10-18 10:38:25

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，本接口不需要传递此参数。
OldProjectId	是	否	String	原项目
NewProjectId	是	否	String	新项目
ResourceList	是	否	Array of TransferResource	资源

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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	
InvalidParameter.UnsupportedProductCodeError	
ResourceNotFound.ProjectResourceNotFound	
FailedOperation.ProjectCountError	
FailedOperation.ProjectResourceNotEmpty	

项目名是否唯一

最近更新时间: 2024-10-18 10:38:25

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，本接口不需要传递此参数。
ProjectName	是	否	String	项目名

3. 输出参数

参数名称	类型	描述
Exist	Bool	**此参数对外不可见。** 是否存在
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	
InternalError.ComponentFail	
InternalError.DatabaseError	

从项目移除成员

最近更新时间: 2024-10-18 10:38:25

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，本接口不需要传递此参数。
ProjectId	是	否	String	项目ID
Uins	是	否	Array of Uint64	待移除的用户UID列表

3. 输出参数

参数名称	类型	描述
Uins	Uint64	**此参数对外不可见。** 移除的用户uin列表
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	
InternalError.ComponentFail	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectNotFoundError	
InternalError.DatabaseError	
UnauthorizedOperation.AuthFailedError	

转入转出资源

最近更新时间: 2024-10-18 10:38:25

1. 接口描述

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

转入转出资源

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：TransferProjectResource
Version	是	否	String	公共参数，本接口取值：2020-09-20
Region	是	否	String	公共参数，本接口不需要传递此参数。
OldProjectId	是	否	String	原项目ID
NewProjectId	是	否	String	新项目ID
ResourceList	是	否	Array of String	变动的资源列表

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

最近更新时间: 2024-10-18 10:38:25

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
RegionId	否	否	Uint64	地域
ProductCode	否	否	String	产品Code
ResourceType	否	否	String	资源类型（暂不支持）
Keyword	否	否	String	资源ID或资源名称模糊查询
Product	否	否	String	产品Code
ServiceType	否	否	String	服务类型

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

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	uin
PolicyName	是	否	String	策略名

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
SubProductCode	否	否	String	二级产品编码
BillingItemCode	否	否	String	三级产品编码
SubBillingItemCode	否	否	String	四级产品编码

名称	必选	允许NULL	类型	描述
QuotaValue	是	否	Int64	配额值
QuotaKey	否	否	String	配额键

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
ProjectName	是	否	String	项目名称
Organization	是	否	String	项目名

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
ResourceTypes	是	否	Array of Resource	资源类型

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
RegionId	是	否	String	地域ID
ResourceId	是	否	String	资源ID
Uin	否	否	String	用户Uin
Region	否	否	String	地域

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
BillingItemName	是	否	String	产品项
Error	是	否	String	错误内容

名称	必选	允许NULL	类型	描述
Exists	是	否	Bool	是否已有配额
ProductCode	是	否	String	产品code
ProductName	是	否	String	产品名称
QuotaKey	是	否	String	配额键
QuotaLeft	是	否	String	剩余配额
QuotaUsed	是	否	String	已使用配额
QuotaValue	是	否	String	配额值
SubBillingItemCode	是	否	String	四级产品
SubBillingItemName	是	否	String	产品细项
SubProductCode	是	否	String	二级产品
SubProductName	是	否	String	子产品名
Success	是	否	Bool	是否合法

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
ResourceId	是	否	String	资源ID
ResourceName	是	否	String	资源名称
ProductCode	是	否	String	产品Code
ProductName	是	否	String	产品名称
ProductGroupName	是	否	String	产品大类名称
RegionId	是	否	Int64	地域ID
ProjectName	是	否	String	项目名称
RegionEnName	是	否	String	地域英文名
RegionName	是	否	String	地域名称
ResourceType	是	否	String	资源类型
ServiceType	是	否	String	产品对应的serviceType

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

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
SubProductCode	否	否	String	二级产品
BillingItemCode	否	否	String	三级产品
SubBillingItemCode	否	否	String	四级产品
Keyword	否	否	String	配额项名称模糊查询

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
RegionName	是	否	String	地域名称

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
QuotaKey	是	否	String	配额键

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
ProductName	是	否	String	产品名
Unit	是	否	String	单位

Resource

资源类型

被如下接口引用：DescribeProducts

版权所有：腾讯云金融专区

名称	必选	允许NULL	类型	描述
ProductCode	是	否	String	产品Code
ResourceType	是	否	String	资源类型
ServiceType	是	否	String	服务类型，对应cam serivceType

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
ProductName	是	否	String	产品名
ProductCode	是	否	String	产品Code
SubProductCode	是	是	String	二级产品
BillingItemCode	是	是	String	三级产品
SubBillingItemCode	是	是	String	四级产品
QuotaKey	是	否	String	配额键
QuotaName	是	是	String	配额项名称
QuotaValue	是	否	String	总配额
QuotaLeft	是	否	String	剩余配额
QuotaUsed	是	否	String	已使用配额
CreateTime	是	否	String	创建时间
UpdateTime	是	否	String	更新时间
BillingItemName	是	否	String	产品项
SubBillingItemName	是	否	String	产品细项
SubProductName	是	否	String	子产品名
Unit	是	否	String	单位

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

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
Description	是	否	String	策略描述
PolicyId	是	否	Uint64	策略ID

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
ProjectName	是	否	String	项目名称
Organization	是	否	String	组织
Creator	是	否	String	创建者
CreateTime	是	否	Datetime	创建时间
CreatorUin	是	否	Uint64	创建者Uin
OrgId	是	否	String	组织id
OrgName	是	否	String	组织名称
OrgOperationTime	是	否	String	项目转入项目目录时间
OrgOperator	是	否	String	项目转入项目目录操作人
ProjectDescription	否	否	String	项目描述

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
SubProductName	否	否	String	二级产品名
BillingItemName	否	否	String	三级产品名
SubBillingItemName	否	否	String	四级产品名

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
Uid	是	否	Uint64	用户Uid

版权所有：腾讯云金融专区

名称	必选	允许NULL	类型	描述
Name	是	否	String	用户名
Policies	是	否	Array of Policy	具有的项目策略

错误码

最近更新时间: 2024-10-18 10:38:25

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

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

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
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。
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。

版权所有：腾讯云金融专区

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

业务错误码

错误码	说明
InvalidParameter.InvalidProjectQuota	
UnauthorizedOperation.AuthFailedError	
InvalidParameter.UnsupportedProductCodeError	
InternalError.ComponentFail	
FailedOperation.ProjectCountError	
FailedOperation.ProjectResourceNotEmpty	
InternalError.InvalidHTTPCode	
ResourceNotFound.ProjectResourceNotFound	
InvalidParameter.InvalidParameter	
InternalError.DatabaseError	
ResourceNotFound.ProjectNotFoundError	
InternalError.ThirdPartServiceBadRequest	
InvalidParameter.EmptyParameter	
InvalidParameter.UsedQuotaNotEnough	
FailedOperation.ProjectQuotaNotEmpty	
InvalidParameter.ProjectNameTooLong	

版权所有：腾讯云金融专区

账号系统（account）
版本（2019-03-25）
API概览

最近更新时间: 2024-10-18 10:38:26

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

最近更新时间: 2024-10-18 10:38:26

tcecloud 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tcecloud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
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	ap-guangzhou
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

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

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : '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=ap-guangzhou&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=ap-guangzhou&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=ap-guangzhou&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tcecloud 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
- JavaScript
- .NET

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： `http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&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 TceCloudAPIDemo {
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("http://imgcache.finance.cloud.tencent.com:80cvm.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)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如： params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
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", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
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' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
# resp = requests.get("http://imgcache.finance.cloud.tencent.com:80" + endpoint, params=data)
# print(resp.url)
```

接口签名v3

最近更新时间: 2024-10-18 10:38:26

tcecloud 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tcecloud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：

```
http://imgcache.finance.cloud.tencent.com:80cvm.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: ap-guangzhou
```

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 TceCloudAPITC3Demo {
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-urlencoded";
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 = "ap-guangzhou";
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" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

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

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

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("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 = "http://imgcache.finance.cloud.tencent.com:80" + host
region = "ap-guangzhou"
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

最近更新时间: 2024-10-18 10:38:26

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。tcecloud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

tcecloud 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

最近更新时间: 2024-10-18 10:38:26

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 correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码 (TODO: 重复信息, 是否真的需要?)

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。

错误码	错误描述
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。

公共参数

最近更新时间: 2024-10-18 10:38:26

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。
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。tcecloud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

最近更新时间: 2024-10-18 10:38:26

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：2022-12-12 19:44:24。

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，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	**此参数对外不可见。** 多因子配置
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	
InvalidParameter.ParamError	

数据结构

最近更新时间: 2024-10-18 10:38:27

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
SecretKey	否	否	String	密钥Key
CreateTime	否	否	Uint64	创建时间(时间戳)
Status	否	否	Uint64	状态(2:有效, 3:禁用)
Remark	否	否	String	备注
Source	否	否	Uint64	数据源

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
UserName	否	否	String	用户名称
NickName	否	否	String	用户名称

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
Attribute	否	否	String	属性
AttributeId	否	否	Int64	属性id
ValueId	否	否	Int64	值id
Uin	否	否	Int64	uin
Value	否	否	String	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
AttributeName	否	否	String	属性名称
AttributeValue	否	否	String	属性值
AttributeId	否	否	Int64	属性id

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
Token	是	否	Int64	是否进行token认证，0-否，1-是
Stoken	是	否	Int64	是否进行stoken认证，0-否，1-是
Ukey	否	否	Int64	是否进行ukey检查

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
AuthType	是	否	Int64	认证类型
AuthPhone	是	否	String	认证手机号

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
ActionFlag	是	是	SafeAuthFlag	action标识
OffsiteFlag	是	是	SafeAuthOffsiteFlag	外围标识

LoginInfo

登录信息

被如下接口引用：GetLastLoginInfo

名称	必选	允许NULL	类型	描述
ClientIP	否	否	String	客户端ip

版权所有：腾讯云金融专区

名称	必选	允许NULL	类型	描述
ClientUA	否	否	String	客户端代理
DeviceId	否	否	String	设备id
Id	否	否	Int64	id
Location	否	否	String	登录地址
LoginFlag	否	否	Int64	登录标识
LoginTime	否	否	String	登录时间
OwnerUin	否	否	Uint64	主账户uin
Uin	否	否	Uint64	uin
UpdateTime	否	否	String	更新时间

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
NeedResetStoken	是	是	Int64	需要重置mfa的token

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
CreateTime	否	否	Uint64	创建时间(时间戳)
Status	否	否	Uint64	状态(2:有效, 3:禁用)
Remark	否	否	String	备注
Source	否	否	Uint64	数据源

AccountExistData

账户存在时的账户信息

被如下接口引用：

名称	必选	允许NULL	类型	描述
Uin	是	否	Uint64	账户唯一id
Account	是	否	String	用户名
Nick	是	否	String	昵称
Status	是	否	Int64	账户状态

名称	必选	允许NULL	类型	描述
TypeBind	是	否	Int64	账户绑定类型
IsRegAccount	是	否	Int64	是否为注册账户
UpdateTime	是	否	String	最近一次更新时间

AccountDetail

账号详情

被如下接口引用：AddSubAccount

名称	必选	允许NULL	类型	描述
ActionFlag	否	否	ActionLoginFlag	敏感操作标识
ConsoleLogin	否	否	String	是否允许控制台登录
LoginFlag	否	否	ActionLoginFlag	登录保护
NeedResetPassword	否	否	String	是否需要重置密码
Password	否	否	String	用户密码
UseApi	否	否	String	使用Api

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
Uin	否	是	Int64	子用户用户 ID
OwnerUin	否	是	Uint64	主账号ID
Name	否	是	String	账号名称
PhoneNum	否	是	String	联系手机
PhoneFlag	否	是	Int64	电话号码是否验证
Email	否	是	String	联系邮箱
EmailFlag	否	是	Int64	邮箱是否验证
CountryCode	否	是	String	区号
CanLogin	否	是	Int64	是否允许登录
ConsoleLogin	否	是	Int64	是否允许控制台登录
SystemType	否	是	String	类型
WxzsStatus	否	是	Int64	微信公众号关注状态
NeedResetPassword	否	是	Int64	是否需要重置密码
ExtAttr	否	是	ExtAttr	额外属性

版权所有：腾讯云金融专区

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
NotifyPhone	是	否	Int64	是否进行电话通知
NotifyEmail	是	否	Int64	是否进行

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
AttributeName	否	否	String	属性名称
Attribute	否	否	String	属性

ActionLoginFlag

登录操作敏感标识

被如下接口引用：AddSubAccount

名称	必选	允许NULL	类型	描述
Phone	否	否	String	电话
Stoken	否	否	String	软Token
Token	否	否	String	硬Token

UserProductUsableInfo

用户可用产品信息

被如下接口引用：GetUserProductUsable

名称	必选	允许NULL	类型	描述
Unit	否	否	String	单位
ProductId	否	否	String	产品id
IsOpen	否	否	Int64	是否打开
Nums	否	否	Int64	总量
DeadNums	否	否	Int64	DeadNums
Prefix	否	否	String	前缀
RegionId	否	否	String	区域id

版权所有：腾讯云金融专区

UserInfo

用户信息

被如下接口引用：AddSubAccount

名称	必选	允许NULL	类型	描述
CanLogin	否	否	String	子账号类型
CountryCode	否	否	String	区号
Detail	否	否	AccountDetail	详情
Name	否	否	String	名称
PhoneNum	否	否	String	电话号码
SyStemType	否	否	String	系统类型

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
TokenSn	否	否	String	token序列号
TokenType	否	否	Int64	token类型，1-hard token, 2-soft token，3-ukey
TokenSource	否	否	String	设备厂商来源

SubAccounts

子用户信息

被如下接口引用：AddSubAccount

名称	必选	允许NULL	类型	描述
Name	否	是	String	名字
Token	否	是	String	Token
Password	否	是	String	密码
SecretId	否	是	String	秘钥Id
SecretKey	否	是	String	秘钥Key

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
UserName	否	是	String	用户名

版权所有：腾讯云金融专区

名称	必选	允许NULL	类型	描述
DisplayName	否	是	String	显示名称
CheckStatus	否	是	Int64	账户状态
Account	否	是	String	用户名
IsDefaultOwner	否	是	Bool	是否为该子账户的默认主账户
Phone	否	是	String	手机号，打掩码处理
Email	否	是	String	邮箱，掩码处理
ExtAttr	否	是	ExtAttr	扩展属性
LoginMfa	否	是	LoginMfa	多因子认证

AccountRequest

getInfoByFields中账户信息参数

被如下接口引用：GetInfoByFields

名称	必选	允许NULL	类型	描述
Type	是	否	String	账户类型
Id	是	否	String	账户id
OwnerUin	是	否	String	主账户id
SubAccountList	否	否	String	子账户id列表
UidList	否	否	String	uid列表
RoleList	否	否	String	角色列表
GroupList	否	否	String	用户组列表

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
MultiFactorcChangable	否	否	Int64	MultiFactorcChangable
MultiActionChoices	否	否	Int64	MultiActionChoices
MultiActionChangable	否	否	Int64	MultiActionChangable
MfaHardTokenSource	否	否	String	MfaHardTokenSource

CategoryList

账号类别

被如下接口引用：DescribeUserCategory

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
CategoryId	否	否	String	类别Id
CategoryKey	否	否	String	类别键
CategoryValue	否	否	String	类别值

错误码

最近更新时间: 2024-10-18 10:38:27

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

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

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
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。
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。

错误码	说明
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.MfaTokenAlreadyDistribute	
FailedOperation.SoftTokenNotDistribute	
InvalidParameter.TokenTypeError	
FailedOperation.MfaTokenUnBindFail	
FailedOperation.UnknownTokenType	
FailedOperation.BindNoneHardToken	
InvalidParameter.ParamError	
FailedOperation.PasswordMismatch	
FailedOperation.HardTokenNotDistribute	
FailedOperation.HardTokenAlreadyBind	
FailedOperation.UrgentCodeFrequentLimit	
LimitExceeded.SubAccountIsFull	
FailedOperation.BindNoneSoftToken	
FailedOperation.MfaTokenNotDistribute	
InternalError.SystemError	
FailedOperation.InSafeAuthProtection	
InvalidParameter.HardTokenSnMiss	
InvalidParameter.PasswordNotMatch	
FailedOperation.MfaTokenAlreadyBind	
ResourceNotFound.UserNotExists	
InvalidParameter.PasswordDecryptError	
FailedOperation.DistributeMfaTokenError	
FailedOperation.MfaTokenBindFail	

版权所有：腾讯云金融专区

TCE 平台信息（location）
版本（2019-11-28）
API概览

最近更新时间: 2024-10-18 10:38:27

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

调用方式
接口签名v1

最近更新时间: 2024-10-18 10:38:27

tcecloud 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tcecloud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
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	ap-guangzhou
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

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

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : '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=ap-guangzhou&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=ap-guangzhou&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=ap-guangzhou&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tcecloud 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
- JavaScript
- .NET

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： `http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&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 TceCloudAPIDemo {
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("http://imgcache.finance.cloud.tencent.com:80cvm.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)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如： params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
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", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
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' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
# resp = requests.get("http://imgcache.finance.cloud.tencent.com:80" + endpoint, params=data)
# print(resp.url)
```

接口签名v3

最近更新时间: 2024-10-18 10:38:27

tcecloud 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tcecloud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：

```
http://imgcache.finance.cloud.tencent.com:80cvm.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: ap-guangzhou
```

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 TceCloudAPITC3Demo {
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-urlencoded";
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 = "ap-guangzhou";
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" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

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

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

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("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 = "http://imgcache.finance.cloud.tencent.com:80" + host
region = "ap-guangzhou"
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

最近更新时间: 2024-10-18 10:38:27

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。tcecloud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

tcecloud 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

最近更新时间: 2024-10-18 10:38:27

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 correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码 (TODO: 重复信息, 是否真的需要?)

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。

错误码	错误描述
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。

公共参数

最近更新时间: 2024-10-18 10:38:27

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。
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。tcecloud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

最近更新时间: 2024-10-18 10:38:27

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：2021-06-10 14:38:47。

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：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，本接口不需要传递此参数。
ProductId	是	否	String	TCE 产品Code
SubProductId	否	否	String	TCE 子产品Code
Regions	否	否	Array of String	待过滤的Region列表

3. 输出参数

参数名称	类型	描述
RegionCount	Int64	地域数量
RegionSet	RegionEx	地域信息
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

最近更新时间: 2024-10-18 10:38:27

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：2019-12-13 11:22:25。

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：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，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
RegionSet	RegionInfo	地域列表信息
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

最近更新时间: 2024-10-18 10:38:27

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：2019-12-04 16:17:41。

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，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
ZoneSet	ZoneInfo	可用区列表信息
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

最近更新时间: 2024-10-18 10:38:27

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)
RegionID	是	否	String	地域编号(国际化支持, 前端展示使用)
RegionName	是	否	String	地域名称
RegionState	是	否	String	地域状态(AVAILABLE/UNAVAILABLE)
RegionStateRemark	是	否	String	地域状态描述
ZoneCount	是	否	Int64	可用区数量
ZoneSet	是	否	Array of ZoneEx	可用区列表
RegionRole	否	否	String	地域角色(MR:主地域, SR:从地域, OR:其它地域)

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
Zone	是	否	String	可用区名称(固定为英文字符串, API使用)
ZoneID	是	否	String	可用区编号
ZoneName	是	否	String	可用区名称(国际化支持, 前端展示使用)
ZoneState	是	否	String	可用区状态(AVAILABLE/UNAVAILABLE)
ZoneStateRemark	是	否	String	可用区状态(原因描述)
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）

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-guangzhou-3
ZoneName	是	否	String	可用区描述，例如，广州三区
ZoneId	是	否	String	可用区ID
ZoneState	是	否	String	可用区ID

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-guangzhou
RegionName	是	否	String	地域描述，例如，华南地区(广州)
RegionState	是	否	String	地域是否可用状态

错误码

最近更新时间: 2024-10-18 10:38:27

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

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

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
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。
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。

错误码	说明
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

开放平台（open）
版本（2020-12-02）
API概览

最近更新时间: 2024-10-18 10:38:28

API版本

V3

open相关接口

接口名称	接口功能
CreateLdapIdp	CreateLdapIdp
GetLdapIdpConfig	GetLdapIdpConfig
ListIdentityProvider	ListIdentityProvider
TestLdap	TestLdap
UpdateLdapIdp	UpdateLdapIdp

调用方式
接口签名v1

最近更新时间: 2024-10-18 10:38:28

tcecloud 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tcecloud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
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	ap-guangzhou
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

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

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : '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=ap-guangzhou&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=ap-guangzhou&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=ap-guangzhou&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tcecloud 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
- JavaScript
- .NET

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： `http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&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 TceCloudAPIDemo {
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("http://imgcache.finance.cloud.tencent.com:80cvm.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)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如： params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
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", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
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' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
# resp = requests.get("http://imgcache.finance.cloud.tencent.com:80" + endpoint, params=data)
# print(resp.url)
```

接口签名v3

最近更新时间: 2024-10-18 10:38:28

tcecloud 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tcecloud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：

```
http://imgcache.finance.cloud.tencent.com:80cvm.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: ap-guangzhou
```

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 TceCloudAPITC3Demo {
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-urlencoded";
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 = "ap-guangzhou";
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" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

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

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

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("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 = "http://imgcache.finance.cloud.tencent.com:80" + host
region = "ap-guangzhou"
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

最近更新时间: 2024-10-18 10:38:28

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。tcecloud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

tcecloud 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

最近更新时间: 2024-10-18 10:38:28

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 correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码 (TODO: 重复信息, 是否真的需要?)

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。

错误码	错误描述
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。

公共参数

最近更新时间: 2024-10-18 10:38:29

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。
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。tcecloud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。

open相关接口

CreateLdapIdp

最近更新时间: 2024-10-18 10:38:29

1. 接口描述

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

创建ldap认证源

默认接口请求频率限制：20次/秒。

接口更新时间：2022-10-26 17:05:05。

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：CreateLdapIdp
Version	是	否	String	公共参数，本接口取值：2020-12-02
Region	是	否	String	公共参数，本接口不需要传递此参数。
AuthType	是	否	String	认证类型
IdpName	是	否	String	提供商名称
Remark	否	否	String	备注
LdapType	是	否	String	ldap类型
LdapUrl	是	否	String	地址
ConnectType	是	否	String	连接类型
Cert	否	否	String	证书
BaseDn	是	否	String	basedn
AdminAccount	是	否	String	管理账号
AdminPassword	是	否	String	管理账号密码
FilterCondition	是	否	String	过滤条件
UserAccountField	是	否	String	用户账号字段
UserMailField	是	否	String	邮箱字段
UserNicknameField	否	否	String	昵称字段
UserPhoneField	是	否	String	电话字段
IsSyncIdpUser	否	否	Int64	是否同步 idp 用户

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。

GetLdapIdpConfig

最近更新时间: 2024-10-18 10:38:29

1. 接口描述

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

获取ldap认证源

默认接口请求频率限制：20次/秒。

接口更新时间：2022-02-25 22:57:11。

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：GetLdapIdpConfig
Version	是	否	String	公共参数，本接口取值：2020-12-02
Region	是	否	String	公共参数，本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
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。

ListIdentityProvider

最近更新时间: 2024-10-18 10:38:29

1. 接口描述

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

获取认证源列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-02-09 17:40:06。

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：ListIdentityProvider
Version	是	否	String	公共参数，本接口取值：2020-12-02
Region	是	否	String	公共参数，本接口不需要传递此参数。

3. 输出参数

参数名称	类型	描述
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。

TestLdap

最近更新时间: 2024-10-18 10:38:29

1. 接口描述

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

测试ldap认证源

默认接口请求频率限制：20次/秒。

接口更新时间：2022-02-25 22:59:30。

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：TestLdap
Version	是	否	String	公共参数，本接口取值：2020-12-02
Region	是	否	String	公共参数，本接口不需要传递此参数。
AuthType	是	否	String	认证类型
IdpName	是	否	String	提供商名称
Remark	否	否	String	备注
LdapType	是	否	String	ldap类型
LdapUrl	是	否	String	地址
ConnectType	是	否	String	连接类型
Cert	否	否	String	证书
BaseDn	是	否	String	basedn
AdminAccount	是	否	String	管理账号
AdminPassword	是	否	String	管理账号密码
FilterCondition	是	否	String	过滤条件
UserAccountField	是	否	String	用户账号字段
UserMailField	是	否	String	邮箱字段
UserNicknameField	是	否	String	昵称字段
UserPhoneField	是	否	String	电话字段
Account	是	否	String	测试账号
Password	是	否	String	测试账号密码

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

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。

UpdateLdapIdp

最近更新时间: 2024-10-18 10:38:29

1. 接口描述

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

更新ldap认证源

默认接口请求频率限制：20次/秒。

接口更新时间：2022-10-26 17:05:47。

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：UpdateLdapIdp
Version	是	否	String	公共参数，本接口取值：2020-12-02
Region	是	否	String	公共参数，本接口不需要传递此参数。
AuthType	是	否	String	认证类型
IdpName	是	否	String	提供商名称
Remark	否	否	String	备注
LdapType	是	否	String	ldap类型
LdapUrl	是	否	String	地址
ConnectType	是	否	String	连接类型
Cert	否	否	String	证书
BaseDn	是	否	String	basedn
AdminAccount	是	否	String	管理账号
AdminPassword	是	否	String	管理账号密码
FilterCondition	是	否	String	过滤条件
UserAccountField	是	否	String	用户账号字段
UserMailField	是	否	String	邮箱字段
UserNicknameField	否	否	String	昵称字段
UserPhoneField	是	否	String	电话字段
Id	是	否	Uint64	Id
IsSyncIdpUser	否	否	Int64	是否同步 idp 用户

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

该接口暂无业务逻辑相关的错误码，其他错误码详见公共错误码。

数据结构

最近更新时间: 2024-10-18 10:38:29

ProviderItem

身份提供商配置

被如下接口引用：

名称	必选	允许NULL	类型	描述
Item	否	否	ProviderData	身份提供商配置

ProviderList

身份提供商列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
List	否	否	Array of Item	身份提供商列表

WechatUserAttr

企业微信用户属性

被如下接口引用：GetWorkWeixinOpenAppMember

名称	必选	允许NULL	类型	描述
WechatUserId	是	否	String	企业微信userId
WechatUserName	是	否	String	企业微信用户名

DepartmentAttr

企业微信部门信息

被如下接口引用：GetWorkWeixinOpenAppMember

名称	必选	允许NULL	类型	描述
DepartmentId	是	否	Uint64	部门id
DepartmentName	是	否	String	部门名称
ParentDepartmentId	是	否	Uint64	父部门id
UserList	是	是	Array of WechatUserAttr	部门成员列表

Item

企业认证身份配置

被如下接口引用：

名称	必选	允许NULL	类型	描述

名称	必选	允许NULL	类型	描述
Id	否	否	Int64	配置 id
Name	否	否	String	配置名称
OwnerUin	否	否	Int64	主账号 uin
ProviderType	否	否	Int64	身份提供商类型
Status	否	否	Int64	企业认证状态
Oauth	否	否	String	oauth协议映射
ModifyTime	否	否	String	修改时间
CreateTime	否	否	String	创建时间

ProviderData

身份提供商数据

被如下接口引用：

名称	必选	允许NULL	类型	描述
Data	否	否	ProviderList	身份提供商数据

错误码

最近更新时间: 2024-10-18 10:38:29

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

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

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
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。
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。

错误码	说明
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.InternalError	
InternalError.ErrorGetWorkWexinOpenAppConfigFail	

平台产品中心（tag）
版本（2018-08-13）
API概览

最近更新时间: 2024-10-18 10:38:41

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

最近更新时间: 2024-10-18 10:38:41

tcecloud 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tcecloud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
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	ap-guangzhou
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

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

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : '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=ap-guangzhou&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=ap-guangzhou&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=ap-guangzhou&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tcecloud 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
- JavaScript
- .NET

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： `http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&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 TceCloudAPIDemo {
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("http://imgcache.finance.cloud.tencent.com:80cvm.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)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如： params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
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", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
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' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
# resp = requests.get("http://imgcache.finance.cloud.tencent.com:80" + endpoint, params=data)
# print(resp.url)
```

接口签名v3

最近更新时间: 2024-10-18 10:38:41

tcecloud 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tcecloud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：

```
http://imgcache.finance.cloud.tencent.com:80cvm.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: ap-guangzhou
```

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 TceCloudAPITC3Demo {
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-urlencoded";
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 = "ap-guangzhou";
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" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

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

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

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("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 = "http://imgcache.finance.cloud.tencent.com:80" + host
region = "ap-guangzhou"
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

最近更新时间: 2024-10-18 10:38:41

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。tcecloud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

tcecloud 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

最近更新时间: 2024-10-18 10:38:41

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 correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码 (TODO: 重复信息, 是否真的需要?)

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。

错误码	错误描述
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。

公共参数

最近更新时间: 2024-10-18 10:38:41

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。
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。tcecloud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

最近更新时间: 2024-10-18 10:38:41

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：2020-05-18 19:31:11。

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，本接口不需要传递此参数。
TagKey	是	否	String	标签键
TagValue	是	否	String	标签值
Resource	是	否	String	资源六段式描述

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	
LimitExceeded.TagKey	
LimitExceeded.TagValue	

创建标签

最近更新时间: 2024-10-18 10:38:41

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：2021-05-28 20:04:47。

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：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，本接口不需要传递此参数。
TagKey	是	否	String	标签键
TagValue	是	否	String	标签值

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	
LimitExceeded.TagValue	
InvalidParameterValue.ReservedTagKey	
InvalidParameterValue.TagKeyCharacterIllegal	
InvalidParameterValue.TagKeyEmpty	
InvalidParameterValue.TagKeyLengthExceeded	
InvalidParameterValue.TagValueCharacterIllegal	
InvalidParameterValue.TagValueLengthExceeded	
ResourceInUse.TagDuplicate	

标签解绑资源

最近更新时间: 2024-10-18 10:38:41

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：2020-05-18 19:41:43。

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，本接口不需要传递此参数。
TagKey	是	否	String	标签键
Resource	是	否	String	资源六段式描述

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	

删除标签

最近更新时间: 2024-10-18 10:38:41

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：2020-05-18 19:47:49。

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，本接口不需要传递此参数。
TagKey	是	否	String	需要删除的标签键
TagValue	是	否	String	需要删除的标签值

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	
ResourceNotFound.TagNonExist	

批量修改资源关联的标签

最近更新时间: 2024-10-18 10:38:42

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：2023-11-07 15:35:08。

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，本接口不需要传递此参数。
Resource	是	否	String	资源的六段式描述
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
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

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	
LimitExceeded.TagKey	
LimitExceeded.TagValue	
InvalidParameter.Tag	
InvalidParameterValue.DeleteTagsParamError	

读接口
查询资源关联标签

最近更新时间: 2024-10-18 10:38:42

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：2020-05-18 20:20:33。

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，本接口不需要传递此参数。
CreateUin	否	否	Uint64	创建者uin
ResourceRegion	否	否	String	资源所在地域
ServiceType	否	否	String	业务类型
ResourcePrefix	否	否	String	资源前缀
ResourceId	否	否	String	资源唯一标识
Offset	否	否	Uint64	数据偏移量，默认为 0, 必须为Limit参数的整数倍
Limit	否	否	Uint64	每页大小，默认为 15
CosResourceId	否	否	Uint64	是否是Cos的资源id

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	**此参数对外不可见。** 结果总数
Offset	Uint64	**此参数对外不可见。** 数据位移偏量
Limit	Uint64	**此参数对外不可见。** 每页大小
Rows	TagResource	**此参数对外不可见。** 资源标签
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

最近更新时间: 2024-10-18 10:38:42

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：2020-05-18 20:18:33。

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：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，本接口不需要传递此参数。
ServiceType	是	否	String	业务类型
ResourcePrefix	是	否	String	资源前缀
ResourceIds	是	否	Array of String	资源唯一标记
ResourceRegion	是	否	String	资源所在地域
Offset	否	否	Uint64	数据偏移量，默认为 0, 必须为Limit参数的整数倍
Limit	否	否	Uint64	每页大小，默认为 15

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	**此参数对外不可见。** 结果总数
Offset	Uint64	**此参数对外不可见。** 数据位移偏量
Limit	Uint64	**此参数对外不可见。** 每页大小
Tags	TagResource	**此参数对外不可见。** 标签列表
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	
InvalidParameterValue.ResourcePrefixInvalid	
InvalidParameterValue.ServiceTypeInvalid	

错误码	描述
InvalidParameterValue.UinInvalid	
InvalidParameterValue.ResourceIdSizeInvalid	

查询标签列表

最近更新时间: 2024-10-18 10:38:42

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：2020-05-18 20:15:39。

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，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
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
Offset	否	否	Uint64	数据偏移量，默认为 0, 必须为Limit参数的整数倍
Limit	否	否	Uint64	每页大小，默认为 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
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
ShowProject	否	否	Uint64	是否展现项目标签

3. 输出参数

参数名称	类型	描述
TotalCount	Uint64	**此参数对外不可见。** 结果总数
Offset	Uint64	**此参数对外不可见。** 数据位移偏量
Limit	Uint64	**此参数对外不可见。** 每页大小
Tags	TagWithDelete	**此参数对外不可见。** 标签列表
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	

数据结构

最近更新时间: 2024-10-18 10:38:42

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
TagValue	否	否	Array of String	标签值数组 多个值的话是或的关系

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
TagValue	是	否	String	标签值
CanDelete	是	否	Uint64	是否可以删除

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
ResourcePrefix	是	否	Array of String	资源前缀

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
TagValue	否	否	String	标签值

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
TagKey	是	否	String	标签键
TagValue	是	否	String	标签值
ResourceId	是	否	String	资源ID
TagKeyMd5	是	否	String	标签键MD5值
TagValueMd5	是	否	String	标签值MD5值
ServiceType	是	是	String	资源类型

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
TagValue	是	否	String	标签值

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
TagKeyValues	是	否	String	标签键值对

错误码

最近更新时间: 2024-10-18 10:38:42

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

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

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
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。
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。

错误码	说明
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
LimitExceeded.TagKey	
InvalidParameterValue.ReservedTagKey	
ResourceInUse.TagDuplicate	
FailedOperation.TagAttachedResource	
InvalidParameterValue.ResourceIdSizeInvalid	
ResourceNotFound.AttachedTagKeyNotFound	
ResourceNotFound.TagNonExist	
InvalidParameterValue.TagFilters	
InvalidParameterValue.TagValueLengthExceeded	
InvalidParameterValue.DeleteTagsParamError	
InvalidParameterValue.TagKeyCharacterIllegal	
InvalidParameterValue.TagValueCharacterIllegal	
InvalidParameterValue.ResourceDescriptionError	
InvalidParameterValue.TagKeyLengthExceeded	
InvalidParameterValue.TagFiltersLengthExceeded	
InvalidParameterValue.ServiceTypeInvalid	
InvalidParameterValue.UinInvalid	
InvalidParameterValue.ResourcePrefixInvalid	
InvalidParameterValue.TagKeyEmpty	
InvalidParameterValue.RegionInvalid	
InvalidParameterValue.TagListEmpty	
LimitExceeded.TagValue	
InvalidParameter.Tag	

访问管理（cam）
版本（2019-01-16）
API概览

最近更新时间: 2024-10-18 10:38:46

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
GetServiceApiList	获取服务及其API列表
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
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

版权所有：腾讯云金融专区

接口名称	接口功能
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

最近更新时间: 2024-10-18 10:38:46

tcecloud 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tcecloud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
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	ap-guangzhou
InstanceIds.0	待查询的实例ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

2.1. 对参数排序

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

```
{
'Action' : 'DescribeInstances',
'InstanceIds.0' : 'ins-09dx96dg',
'Limit' : 20,
'Nonce' : 11886,
'Offset' : 0,
'Region' : 'ap-guangzhou',
'SecretId' : '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=ap-guangzhou&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=ap-guangzhou&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=ap-guangzhou&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tcecloud 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
- JavaScript
- .NET

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： `http://imgcache.finance.cloud.tencent.com:80cvm.finance.cloud.tencent.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&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 TceCloudAPIDemo {
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("http://imgcache.finance.cloud.tencent.com:80cvm.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)).append("&");
}
return url.toString().substring(0, url.length() - 1);
}

public static void main(String[] args) throws Exception {
TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
// 实际调用时应当使用随机数，例如：params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
params.put("Nonce", 11886); // 公共参数
// 实际调用时应当使用系统当前时间，例如： params.put("Timestamp", System.currentTimeMillis() / 1000);
params.put("Timestamp", 1465185768); // 公共参数
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", "ap-guangzhou"); // 公共参数
params.put("Limit", 20); // 业务参数
params.put("Offset", 0); // 业务参数
params.put("InstanceIds.0", "ins-09dx96dg"); // 业务参数
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' : 'ap-guangzhou',
'SecretId' : secret_id,
'Timestamp' : 1465185768, # int(time.time())
'Version': '2017-03-12'
}
s = get_string_to_sign("GET", endpoint, data)
data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
print(data["Signature"])
# 此处会实际调用，成功后可能产生计费
# resp = requests.get("http://imgcache.finance.cloud.tencent.com:80" + endpoint, params=data)
# print(resp.url)
```

接口签名v3

最近更新时间: 2024-10-18 10:38:46

tcecloud 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tcecloud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：

```
http://imgcache.finance.cloud.tencent.com:80cvm.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: ap-guangzhou
```

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 TceCloudAPITC3Demo {
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-urlencoded";
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 = "ap-guangzhou";
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" + canonicalQueryString + "\n"
+ canonicalHeaders + "\n" + signedHeaders + "\n" + hashedRequestPayload;
System.out.println(canonicalRequest);

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

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

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("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 = "http://imgcache.finance.cloud.tencent.com:80" + host
region = "ap-guangzhou"
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

最近更新时间: 2024-10-18 10:38:46

1. 服务地址

地域（Region）是指物理的数据中心的地理区域。tcecloud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

tcecloud 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

最近更新时间: 2024-10-18 10:38:46

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 correct."
},
"RequestId": "ed93f3cb-f35e-473f-b9f3-0d451b8b79c6"
}
}
```

- Error 的出现代表着该请求调用失败。Error 字段连同其内部的 Code 和 Message 字段在调用失败时是必定返回的。
- Code 表示具体出错的错误码，当请求出错时可以先根据该错误码在公共错误码和当前接口对应的错误码列表里面查找对应原因和解决方案。
- Message 显示出了这个错误发生的具体原因，随着业务发展或体验优化，此文本可能会经常保持变更或更新，用户不应依赖这个返回值。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，并提供该 ID 来解决问题。

公共错误码 (TODO: 重复信息, 是否真的需要?)

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。

错误码	错误描述
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。

公共参数

最近更新时间: 2024-10-18 10:38:46

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。
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。tcecloud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

最近更新时间: 2024-10-18 10:38:46

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，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)
RoleName	否	否	String	角色名称(与角色ID必传一项)
PolicyId	否	否	Array of Uint64	策略ID list(与策略名 list必传一项)
PolicyName	否	否	Array of String	策略名 list(与策略ID list必传一项)

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	
InvalidParameter.OperatePoliciesOverLimit	
InvalidParameter.PasswordLengthTooShort	
InternalError.SystemError	
InvalidParameter.ParamError	
InvalidParameter.PolicyIdNotExist	

绑定权限策略到角色

最近更新时间: 2024-10-18 10:38:46

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，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
AttachRoleId	否	否	String	角色ID，用于指定角色，入参 AttachRoleId 与 AttachRoleName 二选一
AttachRoleName	否	否	String	角色名称，用于指定角色，入参 AttachRoleId 与 AttachRoleName 二选一
PolicyName	否	否	String	策略名，入参PolicyId与PolicyName二选一

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	
InternalError.SystemError	
InvalidParameter.RoleNotExist	
InvalidParameter.AttachmentFull	

绑定多个角色到策略

最近更新时间: 2024-10-18 10:38:46

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，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)
RoleName	否	否	Array of String	角色名称(与角色ID必传一项)
PolicyId	否	否	Uint64	策略ID(与策略名必传一项)
PolicyName	否	否	String	策略名(与策略ID必传一项)

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	
InvalidParameter.OperatePoliciesOverLimit	
InvalidParameter.PasswordLengthTooShort	
InternalError.SystemError	
InvalidParameter.ParamError	
InvalidParameter.PolicyIdNotExist	

创建策略

最近更新时间: 2024-10-18 10:38:46

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，本接口不需要传递此参数。
PolicyName	是	否	String	策略名
Description	否	否	String	策略描述
PolicyDocument	是	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.cloud.tencent.com","cls.cloud.tencent.com"]}}}，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（http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/document/product/598/36221）输出参数RoleInfo
CreateMode	否	否	Uint64	创建模式

3. 输出参数

参数名称	类型	描述
PolicyId	Uint64	**此参数对外不可见。** 策略id
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	
ResourceNotFound.UserNotExist	
FailedOperation.PolicyFull	

错误码	描述
InternalError.SystemError	
InvalidParameter.ActionNotExist	

创建角色

最近更新时间: 2024-10-18 10:38:46

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，本接口不需要传递此参数。
RoleName	是	否	String	角色名称
PolicyDocument	是	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.cloud.tencent.com","cls.cloud.tencent.com"]}}]}，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（http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/document/product/598/36221）输出参数RoleInfo
Description	否	否	String	角色描述
ConsoleLogin	否	否	Uint64	是否允许登录 1 为允许 0 为不允许
SessionDuration	否	否	Uint64	申请角色临时密钥的最长有效期限制(范围：0~43200)
RoleType	否	否	String	角色类型(system|user)

3. 输出参数

参数名称	类型	描述
RoleId	String	**此参数对外不可见。** 角色ID
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	
InvalidParameter.ConditionError	
InvalidParameter.DescriptionLengthOverlimit	
InvalidParameter.PrincipalError	

错误码	描述
InvalidParameter.RoleFull	
InvalidParameter.RoleNameError	
InvalidParameter.RoleNameInUse	
InvalidParameter.UserNotExist	
InternalError.SystemError	
InvalidParameter.ParamError	

删除策略

最近更新时间: 2024-10-18 10:38:46

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：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：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，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

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	
InvalidParameter.ParamError	
InvalidParameter.PolicyIdError	
InvalidParameter.PolicyIdNotExist	
ResourceNotFound.NotFound	
ResourceNotFound.PolicyIdNotFound	

删除角色

最近更新时间: 2024-10-18 10:38:46

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，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
RoleName	否	否	String	角色名称，用于指定角色，入参 RoleId 与 RoleName 二选一

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	
InvalidParameter.RoleNotExist	
InvalidParameter.ParamError	

获取角色列表

最近更新时间: 2024-10-18 10:38:46

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：2022-12-02 16:09:25。

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：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，本接口不需要传递此参数。
Page	是	否	Uint64	页码，从1开始
Rp	是	否	Uint64	每页行数，不能大于200
Service	否	否	String	按角色的服务账号载体过滤
Keyword	否	否	Array of String	按角色名或角色描述过滤

3. 输出参数

参数名称	类型	描述
List	RoleInfo	**此参数对外不可见。** 角色详情列表。
TotalNum	Uint64	**此参数对外不可见。** 角色总数
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	
InvalidParameter.ParamError	

解除绑定多个策略到用户组

最近更新时间: 2024-10-18 10:38:46

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：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：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，本接口不需要传递此参数。
GroupId	是	否	Uint64	用户组ID
PolicyId	是	否	Array of Uint64	策略ID list

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	
InvalidParameter.PolicyIdNotExist	
InternalError.SystemError	
InvalidParameter.PolicyIdError	

解除绑定策略到多个用户组

最近更新时间: 2024-10-18 10:38:46

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：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：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，本接口不需要传递此参数。
GroupId	是	否	Array of Uint64	用户组ID list
PolicyId	是	否	Uint64	策略ID

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	
InvalidParameter.PolicyIdNotExist	
InternalError.SystemError	
InvalidParameter.PolicyIdError	

解除绑定策略到多个用户

最近更新时间: 2024-10-18 10:38:46

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：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：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，本接口不需要传递此参数。
TargetUin	是	否	Array of Uint64	目标用户ID list
PolicyId	是	否	Uint64	策略ID

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	
InvalidParameter.PolicyIdNotExist	
FailedOperation.PolicyFull	
InternalError.SystemError	

查看策略详情

最近更新时间: 2024-10-18 10:38:46

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：2022-08-11 19:03:58。

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，本接口不需要传递此参数。
PolicyId	是	否	Uint64	策略Id

3. 输出参数

参数名称	类型	描述
PolicyName	String	**此参数对外不可见。** 策略名
Description	String	**此参数对外不可见。** 策略描述
Type	Uint64	**此参数对外不可见。** 1 表示自定义策略，2 表示预设策略
AddTime	Datetime	**此参数对外不可见。** 创建时间
UpdateTime	Datetime	**此参数对外不可见。** 最近更新时间
PolicyDocument	String	**此参数对外不可见。** 策略文档
PresetAlias	String	**此参数对外不可见。** 备注
IsServiceLinkedRolePolicy	Uint64	**此参数对外不可见。** 是否服务相关策略
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	
InvalidParameter.ParamError	
InternalError.SystemError	
InvalidParameter.PolicyIdError	

获取角色详情

最近更新时间: 2024-10-18 10:38:46

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：2022-08-11 19:03:58。

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，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
RoleName	否	否	String	角色名，用于指定角色，入参 RoleId 与 RoleName 二选一

3. 输出参数

参数名称	类型	描述
RoleInfo	RoleInfo	**此参数对外不可见。** 角色详情
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	
InvalidParameter.ParamError	
InvalidParameter.RoleNotExist	

获取服务及其API列表

最近更新时间: 2024-10-18 10:38:46

1. 接口描述

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

获取服务及其API列表

默认接口请求频率限制：20次/秒。

接口更新时间：2022-12-02 17:17:57。

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：GetServiceApiList
Version	是	否	String	公共参数，本接口取值：2019-01-16
Region	是	否	String	公共参数，本接口不需要传递此参数。
IsReturnApi	是	否	Uint64	是否返回api

3. 输出参数

参数名称	类型	描述
List	ServiceApiInfo	**此参数对外不可见。** 服务以及API列表信息
Condition	String	**此参数对外不可见。** 策略条件
ServiceList	ServiceApiInfo	**此参数对外不可见。** 服务列表
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	
InvalidParameter.ParamError	
InvalidParameter.ServiceTypeError	

获取服务角色信息

最近更新时间: 2024-10-18 10:38:46

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：2022-12-27 15:52:05。

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：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，本接口不需要传递此参数。
RoleName	否	否	String	角色名
PolicyName	否	否	Array of String	策略名列表

3. 输出参数

参数名称	类型	描述
RoleName	String	**此参数对外不可见。** 角色名
ServiceType	String	**此参数对外不可见。** 业务类型
ServiceTypeEn	String	**此参数对外不可见。** 英文业务类型
RoleDesc	String	**此参数对外不可见。** 角色描述
RoleDescEn	String	**此参数对外不可见。** 角色英文描述
PolicyName	String	**此参数对外不可见。** 预设策略名
Remark	String	**此参数对外不可见。** 描述
EnRemark	String	**此参数对外不可见。** 英文描述
PolicyList	RolePolicyList	**此参数对外不可见。** 策略列表
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	
InternalError.SystemError	

查询用户组关联的策略列表

最近更新时间: 2024-10-18 10:38:46

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：2022-08-11 19:03:58。

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，本接口不需要传递此参数。
TargetGroupId	是	否	Uint64	用户组ID
Page	是	否	Uint64	页码，默认值是 1，从 1 开始
Rp	是	否	Uint64	每页大小，默认值是 20

3. 输出参数

参数名称	类型	描述
TotalNum	Uint64	**此参数对外不可见。** 策略总数
List	AttachPolicyInfo	**此参数对外不可见。** 策略列表
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	
InternalError.SystemError	

获取角色绑定的策略列表

最近更新时间: 2024-10-18 10:38:46

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：2022-08-11 19:03:58。

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，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
RoleName	否	否	String	角色名。用于指定角色，入参 RoleId 与 RoleName 二选一
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
Page	是	否	Uint64	页码，从 1 开始
Rp	是	否	Uint64	每页行数，不能大于200

3. 输出参数

参数名称	类型	描述
List	AttachedPolicyOfRole	**此参数对外不可见。** 角色关联的策略列表
TotalNum	Uint64	**此参数对外不可见。** 角色关联的策略总数
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	
InvalidParameter.ParamError	

查询策略关联的实体列表

最近更新时间: 2024-10-18 10:38:46

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：2022-08-11 19:03:58。

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，本接口不需要传递此参数。
PolicyId	是	否	Uint64	策略 id
Page	否	否	Uint64	页码，默认值是 1，从 1 开始
Rp	否	否	Uint64	每页大小，默认值是 20
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'

3. 输出参数

参数名称	类型	描述
TotalNum	Uint64	**此参数对外不可见。** 实体总数
List	AttachEntityOfPolicy	**此参数对外不可见。** 实体列表
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	
InvalidParameter.ParamError	
InternalError.SystemError	

查询策略列表

最近更新时间: 2024-10-18 10:38:46

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：2022-12-02 16:37:36。

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，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
Page	否	否	Uint64	页码，默认值是 1，从 1开始，不能大于 200
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'
Keyword	否	否	String	按策略名匹配
TargetUin	否	否	Uint64	按Uin匹配
TargetGroupId	否	否	Uint64	按组Id匹配
TargetRoleId	否	否	Uint64	按角色Id匹配
ServiceType	否	否	String	按产品Id匹配，如cvm
Client	否	否	String	调用来源，控制台为"console"
FlagUin	否	否	Uint64	按Uin标记关联
FlagGroupId	否	否	Uint64	按GroupId标记关联
FlagRoleId	否	否	Uint64	按角色Id标记关联
CreateMode	否	否	Uint64	创建类型：1.按产品功能或项目权限创建；2.按策略语法创建；3.按策略生成器创建；4.按标签授权创建；5.按权限边界规则创建

3. 输出参数

参数名称	类型	描述
TotalNum	Uint64	**此参数对外不可见。** 策略总数

参数名称	类型	描述
List	StrategyInfo	**此参数对外不可见。** 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
ServiceTypeList	String	**此参数对外不可见。** 服务列表
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	
InvalidParameter.UinError	
InternalError.SystemError	
InvalidParameter.ScopeError	
InvalidParameter.KeywordError	
InvalidParameter.GroupIdError	
InvalidParameter.ServiceTypeError	

修改角色信任策略

最近更新时间: 2024-10-18 10:38:46

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，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
RoleName	否	否	String	角色名称，用于指定角色，入参 RoleId 与 RoleName 二选一
PolicyDocument	是	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.cloud.tencent.com","cls.cloud.tencent.com"]}}]}，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（http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/document/product/598/36221 ）输出参数RoleInfo

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	
InvalidParameter.VersionError	
InternalError.SystemError	
InvalidParameter.AttachmentFull	
InvalidParameter.PrincipalError	
InvalidParameter.RoleNotExist	

错误码	描述
InvalidParameter.ParamError	

更新策略

最近更新时间: 2024-10-18 10:38:46

版权所有：腾讯云金融专区

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，本接口不需要传递此参数。
PolicyId	是	否	Uint64	策略ID
PolicyName	否	否	String	策略名
Description	否	否	String	策略描述
PolicyDocument	否	否	String	策略文档，示例：{"version":"2.0","statement":[{"action":"name/sts:AssumeRole","effect":"allow","principal":{"service":["cloudaudit.cloud.tencent.com","cls.cloud.tencent.com"]}}}，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（http://imgcache.finance.cloud.tencent.com:80cloud.tencent.com/document/product/598/36221）输出参数RoleInfo

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	
InvalidParameter.PrincipalError	
InvalidParameter.ConditionError	
InvalidParameter.ParamError	
InvalidParameter.PolicyDocumentError	
InvalidParameter.ActionError	
ResourceNotFound.PolicyIdNotFound	

错误码	描述
InvalidParameter.NotSupportProduct	
InvalidParameter.PolicyDocumentLengthOverLimit	
ResourceNotFound.UserNotExist	
InternalError.SystemError	
InvalidParameter.StatementError	
InvalidParameter.VersionError	
InvalidParameter.EffectError	
InvalidParameter.PolicyIdNotExist	
InvalidParameter.PolicyIdError	
InvalidParameter.ResourceError	
InvalidParameter.UserNotExist	
ResourceNotFound.GroupNotExist	
ResourceNotFound.NotFound	
FailedOperation.PolicyNameInUse	
InvalidParameter.AttachmentFull	
UnauthorizedOperation	
InvalidParameter.DescriptionLengthOverlimit	

版权所有：腾讯云金融专区

用户相关接口
获取CAM密码规则

最近更新时间: 2024-10-18 10:38:46

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：2022-08-11 19:06:02。

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，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	**此参数对外不可见。** 密码规则列表
UpdateTime	String	**此参数对外不可见。** 更新时间
Modifier	String	**此参数对外不可见。** 更新用户
BlackList	String	**此参数对外不可见。** 黑名单字符串列表json string
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。

更新CAM密码规则

最近更新时间: 2024-10-18 10:38:47

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，本接口不需要传递此参数。
Rules	是	否	PasswordRules	密码设置规则
BlackList	否	否	String	黑名单字符串列表json string

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

最近更新时间: 2024-10-18 10:38:47

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，本接口不需要传递此参数。
Name	是	否	String	身份提供商（企业）名称
Desc	否	否	String	备注
ClientId	是	否	String	注册应用的id
ClientSecret	是	否	String	注册应用的密钥
AuthorizeUri	是	否	String	oauth验证授权信息url
AccessTokenUri	是	否	String	获取access_token url
GetUserInfoUri	是	否	String	获取用户信息url
UserNameField	是	否	String	登录账号对应字段名称
NickNameField	否	否	String	昵称对应字段名称
PhoneNumField	是	否	String	手机号对应字段名称
EmailField	是	否	String	邮箱对应字段名称
IsSyncIdpUser	否	否	Int64	是否同步 idp 用户

3. 输出参数

参数名称	类型	描述
Id	Int64	**此参数对外不可见。** id
Name	String	**此参数对外不可见。** 名称

参数名称	类型	描述
SAMLProviderArn	String	**此参数对外不可见。** SAMLProviderArn
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	

获取用户oauth标识

最近更新时间: 2024-10-18 10:38:47

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，本接口不需要传递此参数。
UserAuthCode	是	否	String	auth code授权码
OpenAccessToken	否	否	String	第三方access token，复杂授权使用。

3. 输出参数

参数名称	类型	描述
AppId	String	**此参数对外不可见。** app id
UserOpenId	String	**此参数对外不可见。** 第三方openId
UserUnionId	String	**此参数对外不可见。** 第三方unionId
UserAccessToken	String	**此参数对外不可见。** 第三方access token
ExpiresAt	Int64	**此参数对外不可见。** 过期时间
UserRefreshToken	String	**此参数对外不可见。** refresh token
Scope	String	**此参数对外不可见。** 授权范围
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	
InternalError.SystemError	
FailedOperation.AuthCodeError	

刷新用户userAccessToken

最近更新时间: 2024-10-18 10:38:47

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，本接口不需要传递此参数。
UserRefreshToken	是	否	String	用户刷新token
UserOpenId	是	否	String	用户openId

3. 输出参数

参数名称	类型	描述
UserAccessToken	String	**此参数对外不可见。** 第三方access_token
ExpiresAt	Int64	**此参数对外不可见。** 过期时间
AppId	String	**此参数对外不可见。** appId
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	
InvalidParameter.ParamError	
InternalError.SystemError	

更新Oauth配置信息

最近更新时间: 2024-10-18 10:38:47

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，本接口不需要传递此参数。
Desc	否	否	String	备注
Name	是	否	String	身份提供商（企业）名称
Id	是	否	Int64	id
OwnerUin	是	否	Int64	OwnerUin
ClientId	是	否	String	注册应用的id
ClientSecret	是	否	String	注册应用的密钥
AuthorizeUri	是	否	String	oauth验证授权信息url
AccessTokenUri	是	否	String	获取access_token url
GetUserInfoUri	是	否	String	获取用户信息url
UserNameField	是	否	String	登录账号对应字段名称
NickNameField	否	否	String	昵称对应字段名称
PhoneNumField	是	否	String	手机号对应字段名称
EmailField	是	否	String	邮箱对应字段名称
IsSyncIdpUser	否	否	Int64	是否同步 idp 用户数据

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

最近更新时间: 2024-10-18 10:38:47

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，本接口不需要传递此参数。
UserAccessToken	是	否	String	access token
UserOpenId	是	否	String	open id

3. 输出参数

参数名称	类型	描述
UserOpenId	String	**此参数对外不可见。** 第三方平台openId
UserUnionId	String	**此参数对外不可见。** 第三方平台unionId
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	
InvalidParameter.ParamError	
InternalError.SystemError	

数据结构

最近更新时间: 2024-10-18 10:38:47

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
Attribute	否	是	String	属性
AttributeValue	否	是	String	属性值

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
Uin	是	是	Uint64	账户唯一id
Name	是	是	String	用户名
IsReceiverOwner	是	是	Int64	是否是主账户
Group	是	是	Array of GroupMeta	用户组信息

ActionLoginFlag

登录操作敏感标识

被如下接口引用：AddSubAccount

名称	必选	允许NULL	类型	描述
Phone	否	否	String	电话
Stoken	否	否	String	软Token
Token	否	否	String	硬Token
Ukey	否	否	String	ukey

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
ConsoleLogin	否	否	String	是否是控制台登陆，1-是

版权所有：腾讯云金融专区

名称	必选	允许NULL	类型	描述
CountryCode	否	否	String	国家编码
Name	否	否	String	用户名
NeedResetPassword	否	否	String	是否需要重置密码，1-是
PhoneNum	否	否	String	手机号
Remark	否	否	String	备注
SystemType	否	否	String	账户类型
Uid	否	否	String	接收者用户ID
Uin	否	否	String	账户唯一id
Password	否	否	String	密码
WxzsStatus	否	否	Int64	微信消息状态
UserType	否	否	Int64	用户类型
Email	否	否	String	联系邮箱
Account	否	否	String	用户名
Lang	否	否	String	语言
NickName	否	否	String	昵称

AccountAttributeAndValue

扩展属性值

被如下接口引用：ListSubAccounts

名称	必选	允许NULL	类型	描述
AttributeName	否	否	String	属性名称
Attribute	否	否	String	属性
AttributeId	否	否	Int64	属性id
ValueId	否	否	Int64	值id
Uin	否	否	Int64	uin
Value	否	否	String	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
ResourceName	否	否	String	中文名
ResourceType	否	否	String	资源类型
ServiceType	否	否	String	服务类型

版权所有：腾讯云金融专区

OwnerInfo

主账号信息

被如下接口引用：ListSubAccounts

名称	必选	允许NULL	类型	描述
Uin	是	是	Uint64	主帐号Uin
UserName	是	是	String	用户名
CheckStatus	是	否	Uint64	校验状态

ServiceAndPerm

业务信息与相关的接口信息

被如下接口引用：DescribeCamServiceAndPerm

名称	必选	允许NULL	类型	描述
ServiceType	是	是	String	业务ID
ServiceName	是	是	String	业务名
IsDisZone	是	是	String	是否分地域、分区
IsDisProject	是	是	String	是否分项目
IsAllowDefProj	是	是	String	是否允许关联默认项目
WhiteKey	是	是	String	灰度白名单 key
PermList	否	是	Array of PermList	相关API列表

GroupMemberInfo

用户组用户信息

被如下接口引用：DescribeGroups、GetGroupList、GetSubsGroup、ListGroups

名称	必选	允许NULL	类型	描述
Uid	是	否	Uint64	子用户 Uid。
Uin	是	否	Uint64	子用户 Uin。
Name	是	否	String	子用户名称。
PhoneNum	是	否	String	手机号。
CountryCode	是	否	String	手机区域代码。
PhoneFlag	是	否	Uint64	是否已验证手机。
Email	是	否	String	邮箱地址。
EmailFlag	是	否	Uint64	是否已验证邮箱。
UserType	是	否	Uint64	用户类型。
CreateTime	是	否	String	创建时间。
IsReceiverOwner	是	否	Uint64	是否为主消息接收人。
QywxUserId	否	否	String	企业微信id

版权所有：腾讯云金融专区

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
GroupName	否	是	String	用户组名称

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
ServiceType	是	否	String	服务ID
ArnDocument	是	是	String	服务介绍文档链接
ApiList	否	是	Array of ServiceApiListInfo	API信息列表
ConditionKeyList	否	是	Array of String	条件规则列表

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
Uin	是	否	Uint64	账号唯一序列号
Name	是	否	String	用户名称
PhoneNum	是	否	String	电话号码
CountryCode	是	否	String	区号
PhoneFlag	是	否	Int64	电话认证标志
Email	是	否	String	邮箱地址
EmailFlag	是	否	Int64	邮箱是否认证
UserType	是	否	Int64	用户类型
CreateTime	是	否	String	创建时间
WechatFlag	是	否	Int64	微信标识
SystemType	是	否	String	账号系统类型
IsReceiverOwner	是	否	Int64	是否为主账号

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
MustContain	是	否	String	最少包含
ForcePasswordChange	是	否	Int64	密码有效期
ReusePasswordLimit	是	否	Int64	密码重复次数
RetryPasswordLimit	否	否	Int64	登陆最大密码失败次数
OnlyAdminCanResetPassword	否	否	Int64	是否只有admin可以重置密码

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
PolicyName	是	否	String	策略名称
AddTime	是	否	String	绑定时间
PolicyType	是	是	String	策略类型，User表示自定义策略，QCS表示预设策略
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

Filter

过滤条件组合

被如下接口引用：ListSubAccounts

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

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
SecretKey	否	否	String	密钥Key
CreateTime	否	否	Uint64	创建时间(时间戳)
Status	否	否	Uint64	状态(2:有效, 3:禁用)
Source	否	否	Uint64	默认0

名称	必选	允许NULL	类型	描述
Remark	否	是	String	描述

UserInfo

用户信息

被如下接口引用：AddSubAccount

名称	必选	允许NULL	类型	描述
CanLogin	否	否	String	子账号类型
CountryCode	否	否	String	区号
Detail	否	否	AccountDetail	详情
Name	否	否	String	名称
PhoneNum	否	否	String	电话号码
SystemType	否	否	String	系统类型
Email	否	否	String	安全邮箱
NickName	否	否	String	昵称
Remark	否	否	String	备注
WxzsStatus	否	否	Int64	微信登陆状态
ContactMail	否	否	String	联系邮箱
IsReceiverOwner	否	否	Int64	是否是主账号
IdentifyType	否	否	Int64	身份类型

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

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
RoleName	是	否	String	角色名称
PolicyDocument	是	否	String	角色的策略文档
Description	是	否	String	角色描述
AddTime	是	否	String	角色的创建时间

版权所有：腾讯云金融专区

名称	必选	允许NULL	类型	描述
UpdateTime	是	否	String	角色的最近一次时间
DeletionTaskId	是	是	String	兼容公有云字段，无含义
ConsoleLogin	是	否	Uint64	角色是否允许登录
RoleType	否	是	String	角色类型，取user、system或service_linked
SessionDuration	否	是	Uint64	有效时间

FilterItem

过滤条件数据结构

被如下接口引用：ListSubAccounts

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

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
CreateUin	是	否	Uint64	创建账户uin
OwnerUin	是	否	Uint64	主账户uin
Name	是	否	String	名称
Desc	是	否	String	描述
ProviderType	是	否	Int64	provider类型
Status	是	否	Int64	状态
ModifyTime	是	否	String	修改时间
CreateTime	是	否	String	创建时间
SAMLMetadata	是	否	String	SAML元数据
SAMLEntityId	是	否	String	SAML实例id
SAMLSingleSignOn	是	否	String	SAML登陆跳转
SAMLSingleLogout	是	否	String	SAML登出跳转
SAMLKeys	是	否	String	SAML关键字
Cas	是	否	String	Cas
CasRoot	是	否	String	cas根地址
CasLoginUrl	是	否	String	cas登陆url

版权所有：腾讯云金融专区

名称	必选	允许NULL	类型	描述
CasValidateUrl	是	否	String	cas校验url
CasLogoutUrl	是	否	String	cas登出url
Oauth	是	否	String	oauth配置

AccountDetail

账号详情

被如下接口引用：AddSubAccount

名称	必选	允许NULL	类型	描述
ActionFlag	否	否	ActionLoginFlag	敏感操作标识
ConsoleLogin	否	否	String	是否允许控制台登录
LoginFlag	否	否	ActionLoginFlag	登录保护
NeedResetPassword	否	否	String	是否需要重置密码
Password	否	否	String	用户密码
UseApi	否	否	String	使用Api
TokenType	否	否	Int64	分配到设备类型

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
PolicyName	是	是	String	策略名称
AddTime	是	是	Datetime	创建时间
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。

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
Name	是	否	String	名字
Remark	是	否	String	备注
PhoneNumber	是	否	String	手机号码
PhoneFlag	是	否	Int64	手机号码是否验证
Email	是	否	String	邮箱

版权所有：腾讯云金融专区

名称	必选	允许NULL	类型	描述
EmailFlag	是	否	Int64	邮箱是否验证
IsReceiverOwner	是	否	Int64	是否主联系人
WechatFlag	是	否	Int64	是否允许微信接收通知
Uin	是	否	Uint64	账号uin

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
GroupName	否	否	String	分组名称

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	String	策略ID
PolicyName	否	否	String	策略名
Description	否	否	String	策略描述
AddTime	否	否	String	创建时间
StrategyType	否	否	String	策略类型(1表示自定义策略，2表示预设策略)
CreateMode	否	否	String	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)
Groups	否	是	Array of AttachedUserPolicyGroupInfo	随组关联信息

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
PolicyId	否	否	Uint64	策略Id
PolicyName	否	否	String	策略名

ApiEnNameList

API英文名信息列表

被如下接口引用：DescribeCamServiceAndPerm

名称	必选	允许NULL	类型	描述
Name	是	是	String	API名称
IsNeedObject	是	是	String	是否需要关联对象
IsSeen	是	是	String	是否可见

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
GroupName	是	否	String	用户组名称
GroupNum	是	否	Int64	用户组成员数量
Channel	是	否	Int64	创建渠道
GroupMem	否	否	Array of Uint64	组成员uid

PrincipalServiceRoleCases

服务角色案例

被如下接口引用：GetPrincipalService

名称	必选	允许NULL	类型	描述
RoleName	是	否	String	角色名
ServiceName	是	否	String	服务载体名
CaseName	是	否	String	场景中文名
CaseEnName	是	否	String	场景英文名
RoleDesc	是	否	String	角色描述
RoleEnDesc	是	否	String	角色英文描述

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
GroupId	是	否	Uint64	用户组id
Uid	是	否	Uint64	用户id
GroupName	否	否	String	用户组名称
Channel	否	否	Int64	息接收渠道 0:无 1: 短信 2：邮件 3：短信+邮件

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
ArnDocument	否	否	String	ArnDocument
ColConf	否	是	String	ColConf
DefAddr	否	是	String	DefAddr
DefaultStrategyList	否	否	String	默认策略
IsAllowDefProj	否	否	String	IsAllowDefProj
IsDisProject	否	否	String	IsDisProject
IsDisZone	否	否	String	IsDisZone
IsSeen	否	否	String	是否可见
Online	否	否	String	Online
QueryAddr	否	否	String	QueryAddr
QueryInterface	否	否	String	QueryInterface
ServiceEnName	否	否	String	服务英文名
ServiceName	否	否	String	服务名
ServiceType	否	否	String	服务类型
SynInterface	否	否	String	SynInterface
UpdateTime	否	否	Datetime	变更时间
Weight	否	否	String	Weight
WhiteKey	否	否	String	WhiteKey
Writter	否	否	String	创建人
ResourceTypeList	否	否	ResourceTypeItem	资源类型数组
Type	否	否	String	类型

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
Name	是	是	String	实体名称
Uin	是	是	Uint64	实体Uin
RelatedType	是	否	Uint64	关联类型。1 用户关联 ； 2 用户组关联

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
ApiAddr	否	否	String	ApiAddr
ApiZhName	否	否	String	中文描述
AuthFunction	否	否	String	鉴权接口
CWildcardName	否	否	String	CWildcardName
InterfaceEnName	否	否	String	接口名
InterfaceLevel	否	否	String	鉴权粒度，0:接口级别、1:资源级别
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
IsNeedObject	否	否	String	IsNeedObject
IsSeen	否	否	String	IsSeen
IsSeenAtGenerator	否	否	String	策略生成器是否可见
IsSpResource	否	否	String	IsSpResource
IsUserSet	否	否	String	IsUserSet
PermId	否	否	String	Id
ReadWriteDetail	否	否	String	接口类别
ResourceType	否	是	String	资源类别
UpdateTime	否	否	String	更新时间
Weight	否	否	String	Weight
Writter	否	否	String	操作者
ServiceName	否	否	String	服务名
ProductShortCode	否	否	String	ProductShortCode
ProductShortName	否	否	String	ProductShortName
ServiceType	否	否	String	服务类型

PermList

API信息列表

被如下接口引用：DescribeCamServiceAndPerm

名称	必选	允许NULL	类型	描述
PermId	是	是	String	Id
ApiZhName	是	是	String	API中文名
ApiEnName	是	是	Array of ApiEnNameList	API英文名信息列表
IsNeedObject	是	是	String	是否需要关联对象
IsSpResource	是	是	String	是否支持实例级鉴权
PmService	是	是	String	pmService

PrinciPalService

服务账号详情

被如下接口引用：GetPrincipalService

名称	必选	允许NULL	类型	描述
Domain	是	是	String	域名
Name	是	是	Name	名称
ServiceLinkedRole	是	是	Uint64	是否存在服务相关角色
RoleCases	是	是	PrincipalServiceRoleCases	角色场景信息
ServiceType	是	是	String	业务模块名

ProviderList

cas身份提供商列表

被如下接口引用：

名称	必选	允许NULL	类型	描述
Id	是	否	Uint64	cas Id号
CreateUin	是	否	Uint64	创建者uin
OwnerUin	是	否	Uint64	主账号
Name	是	否	String	身份提供商名称
Desc	是	否	String	身份提供商描述
ProviderType	是	否	Uint64	身份提供商类型
Status	是	否	Uint64	身份提供商状态
ModifyTime	是	否	Datetime	修改时间
CreateTime	是	否	Datetime	创建时间
Cas	是	否	String	Cas属性

StrategyInfo

策略信息

被如下接口引用：ListPolicies、ListPoliciesByAction

名称	必选	允许NULL	类型	描述
PolicyId	是	否	Uint64	策略ID。
PolicyName	是	否	String	策略名称。
AddTime	是	是	Datetime	策略创建时间。
Type	是	否	Uint64	策略类型。1 表示自定义策略，2 表示预设策略。
Description	是	是	String	策略描述。
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。
Attachments	否	否	Uint64	关联的用户数

名称	必选	允许NULL	类型	描述
ServiceType	是	是	String	策略关联的产品
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
Deactived	否	是	Uint64	是否已下线
DeactivedDetail	否	是	Array of String	已下线产品列表
IsCheck	是	是	Uint64	是否进行安全性校验

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
AppId	是	否	String	app id
OpenName	是	否	String	app name
OpenLogo	是	否	String	open logo
OpenHome	是	否	String	第三方平台主页
OpenType	是	否	Int64	授权类型
Uin	是	否	Uint64	申请账号
Status	是	否	Int64	状态
Domain	是	否	String	第三方平台域名
State	是	否	Int64	冻结状态，0-非冻结，1-冻结
Modifier	是	否	String	修改人
ModifyTime	是	否	String	更新时间
CreateTime	是	否	String	创建时间
Memo	是	否	String	备注

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
NeedResetStoken	是	否	Int64	需要重置mfa的stoken

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
Uin	是	否	Uint64	子用户Uin
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

Name

名称

被如下接口引用：GetPrincipalService

名称	必选	允许NULL	类型	描述
Zh	是	是	String	中文
En	是	否	String	英文

SubAccountFilter

带过滤条件的子帐号信息

被如下接口引用：ListSubAccounts

名称	必选	允许NULL	类型	描述
Uid	是	是	Uint64	子用户Uid
Uin	是	是	Uint64	用户Uin
Name	是	是	String	用户Name
Remark	是	是	String	备注
CanLogin	是	是	Uint64	是都允许登录
PhoneNum	是	是	String	电话号码
CountryCode	是	是	String	区号
PhoneFlag	是	是	Int64	电话号码是否验证
Email	是	是	String	邮箱
EmailFlag	是	是	Int64	邮箱是否验证
UserType	是	是	Int64	用户类型
CreateTime	是	是	String	创建时间
IsReceiverOwner	是	是	Int64	是否消息接收人
SystemType	是	是	String	类型
NeedResetPassword	是	是	Int64	是否需要重置密码
ConsoleLogin	是	是	Int64	是否允许控制台登录
WxzsStatus	是	是	Int64	微信公众号关注状态
PermType	是	是	Array of String	权限类型
NickName	是	是	String	昵称
QywxUserId	否	否	String	企业微信用户id

名称	必选	允许NULL	类型	描述
UserAttributeAndValues	否	否	Array of AccountAttributeAndValue	扩展属性

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
GroupId	是	否	String	组id

GroupInfo

用户组信息

被如下接口引用：DescribeGroups、GetGroupList、GetSubsGroup、ListGroups

名称	必选	允许NULL	类型	描述
GroupId	否	否	Uint64	组id
GroupName	否	否	String	组名称
Channel	否	是	Int64	息接收渠道 0:无 1: 短信 2：邮件 3：短信+邮件
Remark	否	是	String	备注
CreateTime	否	是	String	创建时间
UserInfo	否	是	Array of GroupMemberInfo	用户组成员信息
GroupType	否	是	Int64	用户组类型，0-自定义，1-预设

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
IsNeedObject	是	是	String	是否需要关联对象
Desc	是	是	String	描述
ReadWriteDetail	是	是	Uint64	接口类别：0.读取，1.写入，2.标记，3.列表
InterfaceLevel	是	是	Uint64	授权粒度：0.接口级，1.资源级
ResourceExample	是	是	String	资源六段式范例

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
SubAccountUin	是	否	String	子账号uin

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
Token	否	否	String	token
Password	否	否	String	密码
SecretId	否	否	String	secretid
SecretKey	否	否	String	secretKey
NickName	否	否	String	昵称
Uin	否	否	Uint64	uin

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
CreateTime	否	否	Uint64	创建时间(时间戳)
Status	否	否	Uint64	状态(2:有效, 3:禁用)

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：
TotalNum	否	否	Uint64	策略数
Id	否	否	Uint64	入参Type=1时表示uin，2时表示groupId

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。
PolicyName	否	否	String	策略名称。
AddTime	否	是	Datetime	策略创建时间。
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。
Description	否	是	String	策略描述。

GroupUserInfo

用户组下用户信息

被如下接口引用：GetGroupsSubAccount、ListUsersForGroup

名称	必选	允许NULL	类型	描述
Uid	是	是	Uint64	接收者用户id
Uin	是	是	Uint64	账户唯一id
Name	是	是	String	用户名
PhoneNum	是	是	String	手机号
CountryCode	是	是	String	国家编码
PhoneFlag	是	是	String	手机号标识
Email	是	是	String	邮箱
EmailFlag	是	是	String	邮箱标识
UserType	是	是	Int64	用户类型
CreateTime	是	是	String	创建时间
IsReceiverOwner	是	是	String	是否是主账户
SystemType	是	是	String	账户类型
NickName	是	是	String	昵称

版权所有：腾讯云金融专区

错误码

最近更新时间: 2024-10-18 10:38:47

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

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

Error 中的 Code 表示错误码，Message 表示该错误的具体信息。

错误码列表

公共错误码

错误码	说明
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。
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。

错误码	说明
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.UnknownAccountSettingGroup	
FailedOperation.AccountSettingConfigError	
InvalidParameter.ParamError	
FailedOperation.AccountGroupNameNotMatch	
FailedOperation.SecretIdExist	
InvalidParameter.ServiceTypeError	
InvalidParameter.ResourceError	
ResourceNotFound.IdentityNotExist	
FailedOperation.IdentityExist	
InvalidParameter.PolicyDocumentError	
InvalidParameter.ActionNotExist	
InvalidParameter.PolicyNameError	
FailedOperation.Accesskey	
FailedOperation.UnknownAccountSettingKey	
InvalidParameter.PrincipalError	
FailedOperation.AccountSettingValueCalculateError	
InvalidParameter.OperatePoliciesOverLimit	
ResourceUnavailable.IDPMaxLimit	
ResourceNotFound.PolicyIdNotFound	
InvalidParameter.PasswordLengthTooShort	
FailedOperation.PolicyFull	
FailedOperation.InAsyncModifyError	
InvalidParameter.DescriptionLengthOverlimit	
FailedOperation.NoneValue	
UnauthorizedOperation	
FailedOperation.AddIdentityError	
InvalidParameter.KeywordError	

版权所有：腾讯云金融专区

错误码	说明
InvalidParameter.ScopeError	
InvalidParameter.GroupIdError	
FailedOperation.RefreshTokenError	
InvalidParameter.ActionError	
ResourceNotFound.UserNotExist	
FailedOperation.SkeyExpired	
ResourceNotFound.URLError	
InvalidParameter.RoleNotExist	
InvalidParameter.PolicyDocumentLengthOverLimit	
InvalidParameter.ConditionError	
InvalidParameter.PolicyIdError	
FailedOperation.PolicyNameInUse	
InternalError.SystemError	
InvalidParameter.PolicyIdNotExist	
InvalidParameter.NotSupportProduct	
FailedOperation.AuthCodeError	
FailedOperation.SetLoginRuleFail	
ResourceNotFound.NotFound	
FailedOperation.UserrAccessTokenError	
InvalidParameter.UserNotExist	
InvalidParameter.VersionError	
ResourceNotFound.GroupNotExist	
InvalidParameter.EffectError	
InvalidParameter.AttachmentFull	
InvalidParameter.RoleNameError	
InvalidParameter.RoleFull	
InvalidParameter.RoleNameInUse	
InvalidParameter.UinError	
InvalidParameter.StatementError	
FailedOperation.SubAccountHasKey	

版权所有：腾讯云金融专区