

企业微信汽车行业版

API 文档



腾讯云

【版权声明】

©2013-2024 腾讯云版权所有

本文档（含所有文字、数据、图片等内容）完整的著作权归腾讯云计算（北京）有限责任公司单独所有，未经腾讯云事先明确书面许可，任何主体不得以任何形式复制、修改、使用、抄袭、传播本文档全部或部分内容。前述行为构成对腾讯云著作权的侵犯，腾讯云将依法采取措施追究法律责任。

【商标声明】

及其它腾讯云服务相关的商标均为腾讯云计算（北京）有限责任公司及其关联公司所有。本文档涉及的第三方主体的商标，依法由权利人所有。未经腾讯云及有关权利人书面许可，任何主体不得以任何方式对前述商标进行使用、复制、修改、传播、抄录等行为，否则将构成对腾讯云及有关权利人商标权的侵犯，腾讯云将依法采取措施追究法律责任。

【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。

您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

【联系我们】

我们致力于为您提供个性化的售前购买咨询服务，及相应的技术售后服务，任何问题请联系 4009100100或95716。

文档目录

API 文档

更新历史

简介

API 概览

调用方式

请求结构

公共参数

签名方法 v3

签名方法

返回结果

参数类型

企业微信汽车行业版开放平台相关接口

新增渠道活码接口

新增标签库接口

线索回收接口

查询活动参与明细列表接口

查询活动列表接口

查询活动活码列表接口

查询渠道活码列表接口

查询会话存档列表接口

查询CRM统计数据接口

查询外部联系人SaaS使用明细数据接口

查询外部联系人详情接口

查询外部联系人列表接口

查询外部联系人事件明细列表接口

外部联系人转换接口

查询license信息接口

获取企业素材列表接口

查询小程序码列表接口

企业成员SaaS使用明细数据接口

查询企业成员信息列表接口

查询车系车型信息列表接口

通过时间段查询外部联系人详情列表

查询潜在客户档案列表接口

查询指定时间范围内发生过跟进的潜在客户信息

查询经销商信息列表接口

查询线索列表接口

查询指定时间范围内发生过到店的潜在客户到店信息

数据结构

错误码

API 文档

更新历史

最近更新时间：2023-07-18 01:59:38

第 17 次发布

发布时间：2023-07-18 01:59:28

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ClueInfoDetail](#)
 - 新增成员：UpdateTime

第 16 次发布

发布时间：2023-07-14 01:46:29

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [CustomerProfile](#)
 - 新增成员：LeadId, WxId, Position, IsBindWx, IsInvalid, InvalidType, InvalidTypeName, InvalidTime, InvalidRemark, IsLose, LoseType, LoseTypeName, LoseRemark
 - 修改成员：CustomerId, DealerCode, UnionId, CreateTime, UserName, Gender, Phone, AgeRangeName, JobTypeName, Address, LeadsProcessStatus, LeadType, SourceName, LeadsLevelCode, VehicleBrandCode, VehicleSeriesCode, VehicleTypeCode, VehiclePurpose, PurchaseConcern, SalesName, SalesPhone, RealArrivalTime, CompleteTestDriveTime, OrderTime, DeliveryTime, InvoiceTime, LoseTime, CreatedAtTime, ImportAtTime, DistributeTime, LeadCreateTime, Nickname, OrgIdList, Introducer, IntroducerPhone, FollowTime, NextFollowTime, EnterpriseTags, ChannelTags

第 15 次发布

发布时间：2023-07-11 01:55:43

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ClueInfoDetail](#)
 - 新增成员：CorpShopId

第 14 次发布

发布时间：2023-06-30 12:15:39

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [QueryArrivalList](#)
- [QueryFollowList](#)

新增数据结构：

- [ArrivalInfo](#)
- [FollowInfo](#)

修改数据结构：

- [ClueInfoDetail](#)
 - 新增成员：WxId, BrandCode, BuildTime, OrderTime, ArrivalTime, DeliveryTime, FollowTime, NextFollowTime, OrgId, OrgName, Introducer, IntroducerPhone, IsBindWx, IsMerge, IsInvalid, InvalidType, InvalidTypeName, InvalidRemark, InvalidTime, DealerName, ShopId, ShopName, Position
 - 修改成员：LeadUserType, LeadType, ChannelId, ChannelName, SourceChannelName, Gender, CreateTime, LeadStatus, LevelCode, ImportAtTime, DistributeTime, CreateAtTime
- [CorpUserInfo](#)
 - 新增成员：JobNumber

第 13 次发布

发布时间：2023-03-14 12:01:08

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [QueryCustomerProfileList](#)
- [QueryExternalContactDetailByDate](#)

新增数据结构：

- [ChannelTag](#)
- [CustomerProfile](#)
- [EnterpriseTag](#)
- [ExternalContactDetailPro](#)
- [FollowUserPro](#)
- [PurchaseConcern](#)
- [VehiclePurpose](#)

第 12 次发布

发布时间：2023-03-14 10:50:42

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [QueryClueInfoList](#)
 - 新增入参：BeginTime, EndTime

修改数据结构：

- [ClueInfoDetail](#)
 - 新增成员：LeadUserType, LeadType, ChannelId, ChannelName, SourceChannelName, Gender, CreateTime, LeadStatus, LevelCode, ImportAtTime, DistributeTime, CreateAtTime
- [DealerInfo](#)
 - 新增成员：ProvinceCode, CityCodeList, BrandIdList

第 11 次发布

发布时间：2022-10-12 07:10:28

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ClueInfoDetail](#)
 - 新增成员：UserName

第 10 次发布

发布时间：2022-04-04 07:39:05

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [QueryCrmStatistics](#)
- [QueryCustomerEventDetailStatistics](#)
- [QueryExternalUserEventList](#)
- [QueryMaterialList](#)
- [QueryStaffEventDetailStatistics](#)
- [QueryUserInfoList](#)

新增数据结构：

- [CorpUserInfo](#)
- [CrmStatisticsData](#)
- [CustomerActionEventDetail](#)
- [ExternalUserEventInfo](#)
- [MaterialInfo](#)
- [SalesActionEventDetail](#)

第 9 次发布

发布时间：2021-12-21 08:13:33

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateLead](#)
- [QueryDealerInfoList](#)
- [QueryVehicleInfoList](#)

新增数据结构：

- [DealerInfo](#)
- [VehicleInfo](#)

第 8 次发布

发布时间：2021-09-30 08:11:50

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [QueryClueInfoList](#)

新增数据结构：

- [ClueInfoDetail](#)

第 7 次发布

发布时间：2021-09-27 08:19:04

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ExternalContactSimpleInfo](#)
 - 新增成员：DepartmentIdList

第 6 次发布

发布时间：2021-07-29 08:00:14

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [QueryLicenseInfo](#)

新增数据结构：

- [LicenseInfo](#)

第 5 次发布

发布时间：2021-06-29 08:00:12

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [QueryExternalUserMappingInfo](#)

新增数据结构：

- [ExternalUserMappingInfo](#)

第 4 次发布

发布时间：2021-06-21 08:00:09

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ExternalContact](#)
 - 新增成员：Phone
- [ExternalContactSimpleInfo](#)
 - 新增成员：SalesName

第 3 次发布

发布时间：2021-06-17 08:00:11

本次发布包含了以下内容：

改善已有的文档。

修改数据结构：

- [ChatArchivingDetail](#)
 - 新增成员：BodyJson

第 2 次发布

发布时间：2021-06-15 08:00:03

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [CreateChannelCode](#)
- [CreateCorpTag](#)
- [QueryActivityJoinList](#)
- [QueryActivityList](#)
- [QueryChannelCodeList](#)
- [QueryChatArchivingList](#)
- [QueryExternalContactDetail](#)
- [QueryExternalContactList](#)
- [QueryMiniAppCodeList](#)

新增数据结构：

- [ActivityDetail](#)
- [ActivityJoinDetail](#)
- [ChannelCodeInnerDetail](#)
- [ChatArchivingDetail](#)
- [ChatArchivingMsgTypeVideo](#)

- [ExternalContact](#)
- [ExternalContactSimpleInfo](#)
- [ExternalContactTag](#)
- [FollowUser](#)
- [MiniAppCodeInfo](#)
- [TagDetailInfo](#)
- [TagGroup](#)
- [TagInfo](#)
- [WeComTagDetail](#)

第 1 次发布

发布时间：2021-06-07 14:33:31

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [QueryActivityLiveCodeList](#)

新增数据结构：

- [LiveCodeDetail](#)

简介

最近更新时间：2021-06-23 11:26:43

欢迎使用 企业微信汽车行业版 API 3.0 版本。全新的 API 接口文档更加规范和全面，统一的参数风格和公共错误码，统一的 SDK/CLI 版本与 API 文档严格一致，给您带来简单快捷的使用体验。支持全地域就近接入让您更快连接腾讯云产品。更多腾讯云 API 3.0 使用介绍请查看：[快速入门](#)

企业微信汽车行业版是基于企业微信的汽车行业私域流量运营平台。针对新车销售线索贵、售后客户流失严重等痛点，企业微信汽车行业版通过无缝植入企业微信的销售赋能工具、粉丝及售后会员社群运营、自动化任务推送等功能及服务，帮助车商全面赋能一线销售、最大化客户生命周期价值。

API 概览

最近更新时间：2023-06-30 12:15:53

企业微信汽车行业版开放平台相关接口

接口名称	接口功能	频率限制（次/秒）
CreateChannelCode	新增渠道活码接口	20
CreateCorpTag	新增标签库接口	20
CreateLead	线索回收接口	20
QueryActivityJoinList	查询活动参与明细列表接口	20
QueryActivityList	查询活动列表接口	20
QueryActivityLiveCodeList	查询活动活码列表接口	20
QueryChannelCodeList	查询渠道活码列表接口	20
QueryChatArchivingList	查询会话存档列表接口	20
QueryCrmStatistics	查询CRM统计数据接口	20
QueryCustomerEventDetailStatistics	查询外部联系人SaaS使用明细数据接口	20
QueryExternalContactDetail	查询外部联系人详情接口	20
QueryExternalContactList	查询外部联系人列表接口	20
QueryExternalUserEventList	查询外部联系人事件明细列表接口	20
QueryExternalUserMappingInfo	外部联系人转换接口	20
QueryLicenseInfo	查询license信息接口	20
QueryMaterialList	获取企业素材列表接口	20
QueryMiniAppCodeList	查询小程序码列表接口	20
QueryStaffEventDetailStatistics	企业成员SaaS使用明细数据接口	20
QueryUserInfoList	查询企业成员信息列表接口	20
QueryVehicleInfoList	查询车系车型信息列表接口	20
QueryArrivalList	查询指定时间范围内发生过到店的潜客到店信息	20
QueryClueInfoList	查询线索列表接口	20
QueryCustomerProfileList	查询潜客客户档案列表接口	20
QueryDealerInfoList	查询经销商信息列表接口	20
QueryExternalContactDetailByDate	通过时间段查询外部联系人详情列表	20
QueryFollowList	查询指定时间范围内发生过跟进的潜客信息	20

调用方式

请求结构

最近更新时间：2023-03-14 10:51:12

1. 服务地址

API 支持就近地域接入，本产品就近地域接入域名为 wav.tencentcloudapi.com ，也支持指定地域域名访问，例如广州地域的域名为 wav.ap-guangzhou.tencentcloudapi.com 。

推荐使用就近地域接入域名。根据调用接口时客户端所在位置，会自动解析到最近的某个具体地域的服务器。例如在广州发起请求，会自动解析到广州的服务器，效果和指定 wav.ap-guangzhou.tencentcloudapi.com 是一致的。

注意：对时延敏感的业务，建议指定带地域的域名。

注意：域名是 API 的接入点，并不代表产品或者接口实际提供服务的地域。产品支持的地域列表请在调用方式/公共参数文档中查阅，接口支持的地域请在接口文档输入参数中查阅。

目前支持的域名列表为：

接入地域	域名
就近地域接入（推荐，只支持非金融区）	wav.tencentcloudapi.com
华南地区(广州)	wav.ap-guangzhou.tencentcloudapi.com
华东地区(上海)	wav.ap-shanghai.tencentcloudapi.com
华北地区(北京)	wav.ap-beijing.tencentcloudapi.com
西南地区(成都)	wav.ap-chengdu.tencentcloudapi.com
西南地区(重庆)	wav.ap-chongqing.tencentcloudapi.com
港澳台地区(中国香港)	wav.ap-hongkong.tencentcloudapi.com
亚太东南(新加坡)	wav.ap-singapore.tencentcloudapi.com
亚太东南(曼谷)	wav.ap-bangkok.tencentcloudapi.com
亚太南部(孟买)	wav.ap-mumbai.tencentcloudapi.com
亚太东北(首尔)	wav.ap-seoul.tencentcloudapi.com
亚太东北(东京)	wav.ap-tokyo.tencentcloudapi.com
美国东部(弗吉尼亚)	wav.na-ashburn.tencentcloudapi.com
美国西部(硅谷)	wav.na-siliconvalley.tencentcloudapi.com
北美地区(多伦多)	wav.na-toronto.tencentcloudapi.com
欧洲地区(法兰克福)	wav.eu-frankfurt.tencentcloudapi.com

2. 通信协议

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

3. 请求方法

支持的 HTTP 请求方法:

- POST (推荐)
- GET

POST 请求支持的 Content-Type 类型:

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

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

4. 字符编码

均使用 UTF-8 编码。

公共参数

最近更新时间：2023-12-28 01:29:53

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

公共参数的具体内容会因您使用的签名方法版本不同而有所差异。

使用签名方法 v3 的公共参数

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

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

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

参数名称	类型	必选	描述
Action	String	是	HTTP 请求头：X-TC-Action。操作的接口名称。取值参考接口文档输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	HTTP 请求头：X-TC-Region。地域参数，用来标识希望操作哪个地域的数据。取值参考接口文档中输入参数章节关于公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	HTTP 请求头：X-TC-Timestamp。当前 UNIX 时间戳，可记录发起 API 请求的时间。例如 1529223702。 注意：如果与服务器时间相差超过5分钟，会引起签名过期错误。
Version	String	是	HTTP 请求头：X-TC-Version。操作的 API 的版本。取值参考接口文档中输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
Authorization	String	是	HTTP 标准身份认证头部字段，例如： TC3-HMAC-SHA256 Credential=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 为具体产品名，通常为域名前缀。例如，域名 cvm.tencentcloudapi.com 意味着产品名是 cvm。本产品取值为 wav； tc3_request 为固定字符串； - SignedHeaders：参与签名计算的头部信息，content-type 和 host 为必选头部； - Signature：签名摘要，计算过程详见 文档 。
Token	String	否	HTTP 请求头：X-TC-Token。即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	HTTP 请求头：X-TC-Language。指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

假设用户想要查询广州地域的云服务器实例列表中的前十个，接口参数设置为偏移量 Offset=0，返回数量 Limit=10，则其请求结构按照请求 URL、请求头部、请求体示例如下：

HTTP GET 请求结构示例：

```
https://cvm.tencentcloudapi.com/?Limit=10&Offset=0
```

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

HTTP POST (application/json) 请求结构示例:

```
https://cvm.tencentcloudapi.com/
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKID*****EXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: application/json
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou
```

```
{"Offset":0,"Limit":10}
```

HTTP POST (multipart/form-data) 请求结构示例 (仅特定的接口支持):

```
https://cvm.tencentcloudapi.com/
```

```
Authorization: TC3-HMAC-SHA256 Credential=AKID*****EXAMPLE/2018-05-30/cvm/tc3_request, SignedHeaders=content-type;host, Signature=582c400e06b5924a6f2b5d7d672d79c15b13162d9279b0855cfba6789a8edb4c
Content-Type: multipart/form-data; boundary=58731222010402
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1527672334
X-TC-Region: ap-guangzhou
```

```
--58731222010402
Content-Disposition: form-data; name="Offset"

0
--58731222010402
Content-Disposition: form-data; name="Limit"

10
--58731222010402--
```

使用签名方法 v1 的公共参数

使用签名方法 v1（有时会称作 HmacSHA256 和 HmacSHA1），公共参数需要统一放到请求串中，完整介绍详见[文档](#)

参数名称	类型	必选	描述
Action	String	是	操作的接口名称。取值参考接口文档中输入参数章节关于公共参数 Action 的说明。例如云服务器的查询实例列表接口，取值为 DescribeInstances。
Region	String	-	地域参数，用来标识希望操作哪个地域的数据。接口接受的地域取值参考接口文档中输入参数公共参数 Region 的说明。 注意：某些接口不需要传递该参数，接口文档中会对此特别说明，此时即使传递该参数也不会生效。
Timestamp	Integer	是	当前 UNIX 时间戳，可记录发起 API 请求的时间。例如1529223702，如果与当前时间相差过大，会引起签名过期错误。
Nonce	Integer	是	随机正整数，与 Timestamp 联合起来，用于防止重放攻击。
SecretId	String	是	在 云API密钥 上申请的标识身份的 SecretId，一个 SecretId 对应唯一的 SecretKey，而 SecretKey 会用来生成请求签名 Signature。
Signature	String	是	请求签名，用来验证此次请求的合法性，需要用户根据实际的输入参数计算得出。具体计算方法参见 文档 。
Version	String	是	操作的 API 的版本。取值参考接口文档中入参公共参数 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	即 安全凭证服务 所颁发的临时安全凭证中的 Token，使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言，仅部分接口支持此参数。取值：zh-CN，en-US。zh-CN 返回中文，en-US 返回英文。

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

HTTP GET 请求结构示例：

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

Host: cvm.tencentcloudapi.com
```

HTTP POST 请求结构示例：

```
https://cvm.tencentcloudapi.com/

Host: cvm.tencentcloudapi.com
Content-Type: application/x-www-form-urlencoded

Action=DescribeInstances&Version=2017-03-12&SignatureMethod=HmacSHA256&Timestamp=1527672334&Signature=37ac2f4fde00b0ac9bd9eadeb459b1bbee224158d66e7ae5fcadb70b2d181d02&Region=ap-guangzhou&Nonce=23823223&SecretId=AKID*****EXAMPLE
```

地域列表

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

地域	取值
华南地区（广州）	ap-guangzhou

签名方法 v3

最近更新时间：2023-12-27 01:45:51

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

推荐使用 API Explorer

</> 点击调试

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

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

为什么要进行签名

签名通过以下方式帮助保护请求：

1. 验证请求者的身份

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

2. 保护传输中的数据

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

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

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

申请安全凭证

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

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

申请安全凭证的具体步骤如下：

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

签名版本 v3 签名过程

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

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

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

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

假设用户的 SecretId 和 SecretKey 分别是：AKIDz8krbsJ5yKBZQpn74WFkmLPx3***** 和 Gu5t9xGARNpq86cd98joQYCN3*****。用户想查看广州区云服务器名为“未命名”的主机状态，只返回一条数据。则请求可能为：

```
curl -X POST https://cvm.tencentcloudapi.com \
-H "Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3" \
-H "Content-Type: application/json; charset=utf-8" \
-H "Host: cvm.tencentcloudapi.com" \
-H "X-TC-Action: DescribeInstances" \
-H "X-TC-Timestamp: 1551113065" \
-H "X-TC-Version: 2017-03-12" \
-H "X-TC-Region: ap-guangzhou" \
-d '{"Limit": 1, "Filters": [{"Values": ["\u672a\u547d\u540d"], "Name": "instance-name"}]}'
```

下面详细解释签名计算过程。

1. 拼接规范请求串

按如下伪代码格式拼接规范请求串（CanonicalRequest）：

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

字段名称	解释
HTTPRequestMethod	HTTP 请求方法（GET、POST）。此示例取值为 POST。
CanonicalURI	URI 参数，API 3.0 固定为正斜杠 (/)。
CanonicalQueryString	发起 HTTP 请求 URL 中的查询字符串，对于 POST 请求，固定为空字符串""，对于 GET 请求，则为 URL 中问号（?）后面的字符串内容，例如：Limit=10&Offset=0。 注意：CanonicalQueryString 需要参考 RFC3986 进行 URLEncode 编码（特殊字符编码后需大写字母），字符集 UTF-8。推荐使用编程语言标准库进行编码。

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

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

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

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

2. 拼接待签名字符串

按如下格式拼接待签名字符串：

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

字段名称	解释
Algorithm	签名算法，目前固定为 TC3-HMAC-SHA256。

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

注意：

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

根据以上规则，示例中得到的待签名字符串如下：

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

3. 计算签名

1) 计算派生签名密钥，伪代码如下：

```
SecretKey = "Gu5t9xGARNpq86cd98joQYCN3*****"
SecretDate = HMAC_SHA256("TC3" + SecretKey, Date)
SecretService = HMAC_SHA256(SecretDate, Service)
SecretSigning = HMAC_SHA256(SecretService, "tc3_request")
```

派生出的密钥 SecretDate、SecretService 和 SecretSigning 是二进制的数，可能包含不可打印字符，将其转为十六进制字符串打印的输出分别为：f1cb4d518a0eda9d5cbbfdb7850983f1e603eeae484edea76e4dd8d8deb5556e，e7c609ce81bea53546bed2cc904778bef9ca14082e48e67883443ed64e227cd7，8aa8ab5755582f576e94bcfe383b8e29325b0ca90c3590d569221c6a63a091ed。

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

字段名称	解释
SecretKey	原始的 SecretKey，即 Gu5t9xGARNpq86cd98joQYCN3*****。
Date	即 Credential 中的 Date 字段信息。此示例取值为 2019-02-25。

字段名称	解释
Service	即 Credential 中的 Service 字段信息。此示例取值为 cvm。

2) 计算签名, 伪代码如下:

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

此示例计算结果是 be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3。

4. 拼接 Authorization

按如下格式拼接 Authorization:

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

字段名称	解释
Algorithm	签名方法, 固定为 TC3-HMAC-SHA256。
SecretId	密钥对中的 SecretId, 即 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****。
CredentialScope	见上文, 凭证范围。此示例计算结果是 2019-02-25/cvm/tc3_request。
SignedHeaders	见上文, 参与签名的头部信息。此示例取值为 content-type;host;x-tc-action。
Signature	签名值。此示例计算结果是 be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3。

根据以上规则, 示例中得到的值为:

```
TC3-HMAC-SHA256 Credential=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3
```

最终完整的调用信息如下:

```
POST https://cvm.tencentcloudapi.com/
Authorization: TC3-HMAC-SHA256 Credential=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****/2019-02-25/cvm/tc3_request, SignedHeaders=content-type;host;x-tc-action, Signature=be4f67d323c78ab9acb7395e43c0dbcf822a9cfac32fea2449a7bc7726b770a3
Content-Type: application/json; charset=utf-8
Host: cvm.tencentcloudapi.com
X-TC-Action: DescribeInstances
X-TC-Version: 2017-03-12
X-TC-Timestamp: 1551113065
X-TC-Region: ap-guangzhou

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

⚠ 注意：

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

签名演示

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

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

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

- [Signature Demo](#)

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

Java

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

public class TencentCloudAPITC3Demo {
    private final static Charset UTF8 = StandardCharsets.UTF_8;
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    private final static String SECRET_ID = System.getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    private final static String SECRET_KEY = System.getenv("TENCENTCLOUD_SECRET_KEY");
    private final static String CT_JSON = "application/json; charset=utf-8";

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

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

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

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

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

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

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

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

```
// ***** 步骤 4: 拼接 Authorization *****

String authorization = algorithm + " " + "Credential=" + SECRET_ID + "/" + credentialScope + ", "
+ "SignedHeaders=" + signedHeaders + ", " + "Signature=" + signature;
System.out.println(authorization);

TreeMap<String, String> headers = new TreeMap<String, String>();
headers.put("Authorization", authorization);
headers.put("Content-Type", CT_JSON);
headers.put("Host", host);
headers.put("X-TC-Action", action);
headers.put("X-TC-Timestamp", timestamp);
headers.put("X-TC-Version", version);
headers.put("X-TC-Region", region);

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

Python

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

# 密钥参数
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

service = "cvm"
host = "cvm.tencentcloudapi.com"
endpoint = "https://" + host
region = "ap-guangzhou"
action = "DescribeInstances"
version = "2017-03-12"
algorithm = "TC3-HMAC-SHA256"
#timestamp = int(time.time())
timestamp = 1551113065
```

```

date = datetime.utcfromtimestamp(timestamp).strftime("%Y-%m-%d")
params = {"Limit": 1, "Filters": [{"Values": [u"未命名"], "Name": "instance-name"}]}

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

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

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

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

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

```

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

Golang

```
package main

import (
    "crypto/hmac"
    "crypto/sha256"
    "encoding/hex"
    "fmt"
    "os"
    "strings"
    "time"
)

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

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

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")
    host := "cvm.tencentcloudapi.com"
    algorithm := "TC3-HMAC-SHA256"
    service := "cvm"
    version := "2017-03-12"
    action := "DescribeInstances"
    region := "ap-guangzhou"
    //var timestamp int64 = time.Now().Unix()
    var timestamp int64 = 1551113065

    // step 1: build canonical request string
    httpRequestMethod := "POST"
    canonicalURI := "/"
```

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

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

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

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

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

```
fmt.Println(curl)
}
```

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$host = "cvm.tencentcloudapi.com";
$service = "cvm";
$version = "2017-03-12";
$action = "DescribeInstances";
$region = "ap-guangzhou";
// $timestamp = time();
$timestamp = 1551113065;
$algorithm = "TC3-HMAC-SHA256";

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

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

```
.$timestamp."\n"
.$credentialScope."\n"
.$hashedCanonicalRequest;
echo $stringToSign.PHP_EOL;

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

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

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

Ruby

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

# 密钥参数
# 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKIDz8krbsj5yKBZQpn74WFkmLPx3*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

service = 'cvm'
host = 'cvm.tencentcloudapi.com'
endpoint = 'https://' + host
region = 'ap-guangzhou'
action = 'DescribeInstances'
```

```
version = '2017-03-12'
algorithm = 'TC3-HMAC-SHA256'
# timestamp = Time.now.to_i
timestamp = 1551113065
date = Time.at(timestamp).utc.strftime('%Y-%m-%d')

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

puts canonical_request

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

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

# ***** 步骤 4: 拼接 Authorization *****
authorization = "#{algorithm} Credential=#{secret_id}/#{credential_scope}, SignedHeaders=#{signed_headers}, Signature=
#{signature}"
```

```
puts authorization
```

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

DotNet

```
using System;  
using System.Collections.Generic;  
using System.Security.Cryptography;  
using System.Text;  
  
public class Application  
{  
    public static string SHA256Hex(string s)  
    {  
        using (SHA256 algo = SHA256.Create())  
        {  
            byte[] hashbytes = algo.ComputeHash(Encoding.UTF8.GetBytes(s));  
            StringBuilder builder = new StringBuilder();  
            for (int i = 0; i < hashbytes.Length; ++i)  
            {  
                builder.Append(hashbytes[i].ToString("x2"));  
            }  
            return builder.ToString();  
        }  
    }  
  
    public static byte[] HmacSHA256(byte[] key, byte[] msg)  
    {  
        using (HMACSHA256 mac = new HMACSHA256(key))  
        {  
            return mac.ComputeHash(msg);  
        }  
    }  
  
    public static Dictionary<String, String> BuildHeaders(string secretid,  
        string secretkey, string service, string endpoint, string region,  
        string action, string version, DateTime date, string requestPayload)  
    {  
        string datestr = date.ToString("yyyy-MM-dd");  
        DateTime startTime = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc);
```

```

long requestTimestamp = (long)Math.Round((date - startTime).TotalMilliseconds, MidpointRounding.AwayFromZero) / 1000;
// ***** 步骤 1: 拼接规范请求串 *****
string algorithm = "TC3-HMAC-SHA256";
string httpRequestMethod = "POST";
string canonicalUri = "/";
string canonicalQueryString = "";
string contentType = "application/json";
string canonicalHeaders = "content-type:" + contentType + "; charset=utf-8\n"
+ "host:" + endpoint + "\n"
+ "x-tc-action:" + action.ToLower() + "\n";
string signedHeaders = "content-type;host;x-tc-action";
string hashedRequestPayload = SHA256Hex(requestPayload);
string canonicalRequest = httpRequestMethod + "\n"
+ canonicalUri + "\n"
+ canonicalQueryString + "\n"
+ canonicalHeaders + "\n"
+ signedHeaders + "\n"
+ hashedRequestPayload;
Console.WriteLine(canonicalRequest);

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

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

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

Dictionary<string, string> headers = new Dictionary<string, string>();
headers.Add("Authorization", authorization);
headers.Add("Host", endpoint);
headers.Add("Content-Type", contentType + "; charset=utf-8");
headers.Add("X-TC-Timestamp", requestTimestamp.ToString());

```

```
headers.Add("X-TC-Version", version);
headers.Add("X-TC-Action", action);
headers.Add("X-TC-Region", region);
return headers;
}

public static void Main(string[] args)
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

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

    // 此处由于示例规范的原因, 采用时间戳2019-02-26 00:44:25, 此参数作为示例, 如果在项目中, 您应当使用:
    // DateTime date = DateTime.UtcNow;
    // 注意时区, 建议此时间统一采用UTC时间戳, 否则容易出错
    DateTime date = new DateTime(1970, 1, 1, 0, 0, 0, DateTimeKind.Utc).AddSeconds(1551113065);
    string requestPayload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}\"";

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

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

NodeJS

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

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

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

```
return hash.update(message).digest(encoding)
}

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

function main(){
  // 密钥参数
  // 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
  const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
  // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
  const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

  const endpoint = "cvm.tencentcloudapi.com"
  const service = "cvm"
  const region = "ap-guangzhou"
  const action = "DescribeInstances"
  const version = "2017-03-12"
  //const timestamp = getTime()
  const timestamp = 1551113065
  //时间处理, 获取世界时间日期
  const date = getDate(timestamp)

  // ***** 步骤 1: 拼接规范请求串 *****
  const payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}]}"

  const hashedRequestPayload = getHash(payload);
  const httpRequestMethod = "POST"
  const canonicalUri = "/"
  const canonicalQueryString = ""
  const canonicalHeaders = "content-type:application/json; charset=utf-8\\n"
  + "host:" + endpoint + "\\n"
  + "x-tc-action:" + action.toLowerCase() + "\\n"
  const signedHeaders = "content-type;host;x-tc-action"

  const canonicalRequest = httpRequestMethod + "\\n"
  + canonicalUri + "\\n"
  + canonicalQueryString + "\\n"
  + canonicalHeaders + "\\n"
  + signedHeaders + "\\n"
  + hashedRequestPayload
  console.log(canonicalRequest)

  // ***** 步骤 2: 拼接待签名字符串 *****
  const algorithm = "TC3-HMAC-SHA256"
```

```
const hashedCanonicalRequest = getHash(canonicalRequest);
const credentialScope = date + "/" + service + "/" + "tc3_request"
const stringToSign = algorithm + "\n" +
timestamp + "\n" +
credentialScope + "\n" +
hashedCanonicalRequest
console.log(stringToSign)

// ***** 步骤 3: 计算签名 *****
const kDate = sha256(date, 'TC3' + SECRET_KEY)
const kService = sha256(service, kDate)
const kSigning = sha256('tc3_request', kService)
const signature = sha256(stringToSign, kSigning, 'hex')
console.log(signature)

// ***** 步骤 4: 拼接 Authorization *****
const authorization = algorithm + " " +
"Credential=" + SECRET_ID + "/" + credentialScope + ", " +
"SignedHeaders=" + signedHeaders + ", " +
"Signature=" + signature
console.log(authorization)

const curlcmd = 'curl -X POST ' + "https://" + endpoint
+ ' -H "Authorization: ' + authorization + '"
+ ' -H "Content-Type: application/json; charset=utf-8"'
+ ' -H "Host: ' + endpoint + '"
+ ' -H "X-TC-Action: ' + action + '"
+ ' -H "X-TC-Timestamp: ' + timestamp.toString() + '"
+ ' -H "X-TC-Version: ' + version + '"
+ ' -H "X-TC-Region: ' + region + '"
+ " -d " + payload + "'
console.log(curlcmd)
}
main()
```

C++

```
#include <algorithm>
#include <cstdlib>
#include <iostream>
#include <iomanip>
#include <sstream>
#include <string>
#include <stdio.h>
#include <time.h>
#include <openssl/sha.h>
#include <openssl/hmac.h>

using namespace std;
```

```
string get_data(int64_t &timestamp)
{
    string utcDate;
    char buff[20] = {0};
    // time_t timenow;
    struct tm sttime;
    sttime = *gmtime(&timestamp);
    strftime(buff, sizeof(buff), "%Y-%m-%d", &sttime);
    utcDate = string(buff);
    return utcDate;
}

string int2str(int64_t n)
{
    std::stringstream ss;
    ss << n;
    return ss.str();
}

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

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

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

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

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

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

return (ss.str());
}

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

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

int main()
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    string SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    string SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");

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

```
int64_t timestamp = 1551113065;
string date = get_data(timestamp);

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

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

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

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

string curlcmd = "curl -X POST https://" + host + "\n"
+ " -H \"Authorization: \" + authorization + "\"\n"
+ " -H \"Content-Type: application/json; charset=utf-8\" + "\n"
+ " -H \"Host: \" + host + "\"\n"
+ " -H \"X-TC-Action: \" + action + "\"\n"
+ " -H \"X-TC-Timestamp: \" + RequestTimestamp + "\"\n"
```

```
+ " -H \"X-TC-Version: \" + version + "\\n"
+ " -H \"X-TC-Region: \" + region + "\\n"
+ " -d '" + payload + "'";
cout << curlcmd << endl;
return 0;
};
```

C

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

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

void sha256_hex(const char* str, char* result)
{
    char buf[3];
    unsigned char hash[SHA256_DIGEST_LENGTH];
    SHA256_CTX sha256;
    SHA256_Init(&sha256);
    SHA256_Update(&sha256, str, strlen(str));
    SHA256_Final(hash, &sha256);
    for(int i = 0; i < SHA256_DIGEST_LENGTH; i++)
    {
        snprintf(buf, sizeof(buf), "%02x", hash[i]);
        strcat(result, buf);
    }
}

void hmac_sha256(char* key, const char* input, char* result)
{
    unsigned char hash[32];

    HMAC_CTX *h;
    #if OPENSSL_VERSION_NUMBER < 0x10100000L
    HMAC_CTX hmac;
    HMAC_CTX_init(&hmac);
```

```
h = &hmac;
#else
h = HMAC_CTX_new();
#endif

HMAC_Init_ex(h, key, strlen(key), EVP_sha256(), NULL);
HMAC_Update(h, ( unsigned char* )input, strlen(input));
unsigned int len = 32;
HMAC_Final(h, hash, &len);

#ifdef OPENSSL_VERSION_NUMBER < 0x10100000L
HMAC_CTX_cleanup(h);
#else
HMAC_CTX_free(h);
#endif
strncpy(result, (const char*)hash, len);
}

void hex_encode(const char* input, char* output)
{
    static const char* const lut = "0123456789abcdef";
    size_t len = strlen(input);
    char add_out[128] = {0};
    char temp[2] = {0};
    for (size_t i = 0; i < len; ++i)
    {
        const unsigned char c = input[i];
        temp[0] = lut[c >> 4];
        strcat(add_out, temp);
        temp[0] = lut[c & 15];
        strcat(add_out, temp);
    }
    strncpy(output, add_out, 128);
}

void lowercase(const char * src, char * dst)
{
    for (int i = 0; src[i]; i++)
    {
        dst[i] = tolower(src[i]);
    }
}

int main()
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    const char* SECRET_ID = getenv("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    const char* SECRET_KEY = getenv("TENCENTCLOUD_SECRET_KEY");
```

```
const char* service = "cvm";
const char* host = "cvm.tencentcloudapi.com";
const char* region = "ap-guangzhou";
const char* action = "DescribeInstances";
const char* version = "2017-03-12";
int64_t timestamp = 1551113065;
char date[20] = {0};
get_utc_date(timestamp, date, sizeof(date));

// ***** 步骤 1: 拼接规范请求串 *****
const char* http_request_method = "POST";
const char* canonical_uri = "/";
const char* canonical_query_string = "";
char canonical_headers[100] = {"content-type:application/json; charset=utf-8\nhost:"};
strcat(canonical_headers, host);
strcat(canonical_headers, "\n-x-tc-action:");
char value[100] = {0};
lowercase(action, value);
strcat(canonical_headers, value);
strcat(canonical_headers, "\n");
const char* signed_headers = "content-type;host;x-tc-action";
const char* payload = "{\"Limit\": 1, \"Filters\": [{\"Values\": [\"\\u672a\\u547d\\u540d\"], \"Name\": \"instance-name\"}] }";
char hashed_request_payload[100] = {0};
sha256_hex(payload, hashed_request_payload);

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

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

// ***** 步骤 3: 计算签名 *****
char k_key[64] = {0};
sprintf(k_key, "%s%s", "TC3", SECRET_KEY);
char k_date[64] = {0};
```

```
hmac_sha256(k_key, date, k_date);
char k_service[64] = {0};
hmac_sha256(k_date, service, k_service);
char k_signing[64] = {0};
hmac_sha256(k_service, "tc3_request", k_signing);
char k_hmac_sha_sign[64] = {0};
hmac_sha256(k_signing, string_to_sign, k_hmac_sha_sign);

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

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

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

其他语言

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

签名失败

存在以下签名失败的错误码，请根据实际情况处理。

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

错误码	错误描述
AuthFailure.TokenFailure	临时证书 Token 错误。
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）。

签名方法

最近更新时间：2024-01-05 01:34:31

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

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

推荐使用 API Explorer

<> 点击调试

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

腾讯云 API 会对每个访问请求进行身份验证，即每个请求都需要在公共请求参数中包含签名信息（Signature）以验证请求者身份。签名信息由安全凭证生成，安全凭证包括 SecretId 和 SecretKey；若用户还没有安全凭证，请前往 [云API密钥页面](#) 申请，否则无法调用云 API 接口。

1. 申请安全凭证

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

安全凭证包括 SecretId 和 SecretKey：

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

申请安全凭证的具体步骤如下：

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

注意：每个账号最多可以拥有两对 SecretId/SecretKey。

2. 生成签名串

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

假设用户的 SecretId 和 SecretKey 分别是：

- SecretId: AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
- SecretKey: Gu5t9xGARNpq86cd98joQYCN3*****

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	AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
Timestamp	当前时间戳	1465185768
Nonce	随机正整数	11886

参数名称	中文	参数值
Region	实例所在区域	ap-guangzhou
InstanceId.0	待查询的实例 ID	ins-09dx96dg
Offset	偏移量	0
Limit	最大允许输出	20
Version	接口版本号	2017-03-12

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

2.1. 对参数排序

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

```
{
  'Action': 'DescribeInstances',
  'InstanceId.0': 'ins-09dx96dg',
  'Limit': 20,
  'Nonce': 11886,
  'Offset': 0,
  'Region': 'ap-guangzhou',
  'SecretId': 'AKIDz8krbsj5yKBZQpn74WFkmLPx3*****',
  'Timestamp': 1465185768,
  'Version': '2017-03-12',
}
```

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

2.2. 拼接请求字符串

此步骤生成请求字符串。

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

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

```
Action=DescribeInstances&InstanceId.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsj5yKBZQpn74WFkmLPx3*****&Timestamp=1465185768&Version=2017-03-12
```

2.3. 拼接签名原文字符串

此步骤生成签名原文字符串。

签名原文字符串由以下几个参数构成：

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

4. 请求字符串: 即上一步生成的请求字符串。

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

示例的拼接结果为:

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&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 = 'Gu5t9xGARNpq86cd98joQYCN3*****';
$srcStr = 'GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&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:

```
zmmjn35mikh6pM3V7sUEuX4wyYM=
```

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 zmmjn35mikh6pM3V7sUEuX4wyYM=, 最终得到的签名串请求参数 (Signature) 为: zmmjn35mikh6pM3V7sUEuX4wyYM%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 编码前需要以 UTF-8 进行编码。

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

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

4. 签名失败

根据实际情况, 存在以下签名失败的错误码, 请根据实际情况处理。

错误代码	错误描述
AuthFailure.SignatureExpire	签名过期
AuthFailure.SecretIdNotFound	密钥不存在

错误代码	错误描述
AuthFailure.SignatureFailure	签名错误
AuthFailure.TokenFailure	token 错误
AuthFailure.InvalidSecretId	密钥非法（不是云 API 密钥类型）

5. 签名演示

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

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

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

- [Signature Demo](#)

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

最终输出的 url 可能为：`https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****&Signature=zmmjn35mikh6pM3V7sUEuX4wyYM%3D&Timestamp=1465185768&Version=2017-03-12`。

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

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

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

Java

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

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

```

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

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

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

public static void main(String[] args) throws Exception {
    TreeMap<String, Object> params = new TreeMap<String, Object>(); // TreeMap可以自动排序
    // 实际调用时应当使用随机数, 例如: params.put("Nonce", new Random().nextInt(java.lang.Integer.MAX_VALUE));
    params.put("Nonce", 11886); // 公共参数
    // 实际调用时应当使用系统当前时间, 例如: params.put("Timestamp", System.currentTimeMillis() / 1000);
    params.put("Timestamp", 1465185768); // 公共参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    params.put("SecretId", System.getenv("TENCENTCLOUD_SECRET_ID")); // 公共参数
    params.put("Action", "DescribeInstances"); // 公共参数
    params.put("Version", "2017-03-12"); // 公共参数
    params.put("Region", "ap-guangzhou"); // 公共参数
    params.put("Limit", 20); // 业务参数
    params.put("Offset", 0); // 业务参数
    params.put("InstanceId", "ins-09dx96dg"); // 业务参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    params.put("Signature", sign(getStringToSign(params), System.getenv("TENCENTCLOUD_SECRET_KEY"), "HmacSHA1")); // 公共参数
    System.out.println(getUrl(params));
}

```

Python

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

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

import requests

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
secret_id = os.environ.get("TENCENTCLOUD_SECRET_ID")
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
secret_key = os.environ.get("TENCENTCLOUD_SECRET_KEY")

def get_string_to_sign(method, endpoint, params):
    s = method + endpoint + "/"
    query_str = "&".join("%s=%s" % (k, params[k]) for k in sorted(params))
    return s + query_str

def sign_str(key, s, method):
    hmac_str = hmac.new(key.encode("utf8"), s.encode("utf8"), method).digest()
    return base64.b64encode(hmac_str)

if __name__ == '__main__':
    endpoint = "cvm.tencentcloudapi.com"
    data = {
        'Action': 'DescribeInstances',
        'InstanceId.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'ap-guangzhou',
        'SecretId': secret_id,
        'Timestamp': 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用，成功后可能产生计费
    # resp = requests.get("https://" + endpoint, params=data)
    # print(resp.url)
```

Golang

```
package main
```

```
import (  
    "bytes"  
    "crypto/hmac"  
    "crypto/sha1"  
    "encoding/base64"  
    "fmt"  
    "os"  
    "sort"  
    "strconv"  
)  
  
func main() {  
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****  
    secretId := os.Getenv("TENCENTCLOUD_SECRET_ID")  
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****  
    secretKey := os.Getenv("TENCENTCLOUD_SECRET_KEY")  
    params := map[string]string{  
        "Nonce": "11886",  
        "Timestamp": strconv.Itoa(1465185768),  
        "Region": "ap-guangzhou",  
        "SecretId": secretId,  
        "Version": "2017-03-12",  
        "Action": "DescribeInstances",  
        "InstanceId.0": "ins-09dx96dg",  
        "Limit": strconv.Itoa(20),  
        "Offset": strconv.Itoa(0),  
    }  
  
    var buf bytes.Buffer  
    buf.WriteString("GET")  
    buf.WriteString("cvm.tencentcloudapi.com")  
    buf.WriteString("/")  
    buf.WriteString("?")  
  
    // sort keys by ascii asc order  
    keys := make([]string, 0, len(params))  
    for k, _ := range params {  
        keys = append(keys, k)  
    }  
    sort.Strings(keys)  
  
    for i := range keys {  
        k := keys[i]  
        buf.WriteString(k)  
        buf.WriteString("=")  
        buf.WriteString(params[k])  
        buf.WriteString("&")  
    }  
    buf.Truncate(buf.Len() - 1)
```

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

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

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
$secretId = getenv("TENCENTCLOUD_SECRET_ID");
// 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
$secretKey = getenv("TENCENTCLOUD_SECRET_KEY");
$params["Nonce"] = 11886;//rand();
$params["Timestamp"] = 1465185768;//time();
$params["Region"] = "ap-guangzhou";
$params["SecretId"] = $secretId;
$params["Version"] = "2017-03-12";
$params["Action"] = "DescribeInstances";
$params["InstanceId.0"] = "ins-09dx96dg";
$params["Limit"] = 20;
$params["Offset"] = 0;

ksort($params);

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

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

Ruby

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

```
require 'base64'

# 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
secret_id = ENV["TENCENTCLOUD_SECRET_ID"]
# 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
secret_key = ENV["TENCENTCLOUD_SECRET_KEY"]

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

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

DotNet

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

public class Application {
  public static string Sign(string signKey, string secret)
  {
    string signRet = string.Empty;
```

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

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

public static void Main(string[] args)
{
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    string SECRET_ID = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_ID");
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY, 值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    string SECRET_KEY = Environment.GetEnvironmentVariable("TENCENTCLOUD_SECRET_KEY");

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

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

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

NodeJS

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

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

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

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

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

function main(){
    // 密钥参数
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID，值为示例的 AKIDz8krbsJ5yKBZQpn74WFkmLPx3*****
    const SECRET_ID = process.env.TENCENTCLOUD_SECRET_ID
    // 需要设置环境变量 TENCENTCLOUD_SECRET_KEY，值为示例的 Gu5t9xGARNpq86cd98joQYCN3*****
    const SECRET_KEY = process.env.TENCENTCLOUD_SECRET_KEY

    const endpoint = "cvm.tencentcloudapi.com"
    const Region = "ap-guangzhou"
```

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

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

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

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

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

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

返回结果

最近更新时间：2024-01-09 01:30:52

云 API 3.0 接口默认返回 JSON 数据，返回非 JSON 格式的接口会在文档中做出说明。返回 JSON 数据时最大限制为 50 MB，如果返回的数据超过最大限制，请求会失败并返回内部错误。请根据接口文档中给出的过滤功能（例如时间范围）或者分页功能，控制返回数据不要过大。

注意：目前只要请求被服务端正常处理了，响应的 HTTP 状态码均为 200。例如返回的消息体里的错误码是签名失败，但 HTTP 状态码是 200，而不是 401。

正确返回结果

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

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

- Response 及其内部的 RequestId 是固定的字段，无论请求成功与否，只要 API 处理了，则必定会返回。
- RequestId 用于一个 API 请求的唯一标识，如果 API 出现异常，可以联系我们，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

返回结果中如果存在 Error 字段，则表示调用 API 接口失败。Error 中的 Code 字段表示错误码，所有业务都可能出现的错误码为公共错误码。完整的错误码列表请参考本产品“API 文档”目录下的“错误码”页面。

参数类型

最近更新时间：2022-08-10 06:59:43

目前腾讯云 API 3.0 输入参数和输出参数支持如下几种数据格式：

- String: 字符串。
- Integer: 整型，上限为无符号64位整数。SDK 3.0 不同编程语言支持的类型有所差异，建议以所使用编程语言的最大整型定义，例如 Golang 的 uint64。
- Boolean: 布尔型。
- Float: 浮点型。
- Double: 双精度浮点型。
- Date: 字符串，日期格式。例如：2022-01-01。
- Timestamp: 字符串，时间格式。例如：2022-01-01 00:00:00。
- Timestamp ISO8601: ISO 8601 是由国际标准化组织（International Organization for Standardization，ISO）发布的关于日期和时间格式的国际标准，对应国标《GB/T 7408-2005数据元和交换格式信息交换日期和时间表示法》。建议以所使用编程语言的标准库进行格式解析。例如：2022-01-01T00:00:00+08:00。
- Binary: 二进制内容，需要以特定协议请求和解析。

企业微信汽车行业版开放平台相关接口

新增渠道活码接口

最近更新时间：2023-11-30 08:21:18

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

新增渠道活码接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateChannelCode。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Type	是	Integer	欢迎语类型:0普通欢迎语,1渠道欢迎语 示例值：1
UseUserId.N	是	Array of Integer	使用成员用户id集 示例值：[1386218784692961300]
UseUserOpenId.N	是	Array of String	使用成员企微账号id集 示例值：["544064e019b53e4791115020ffc68f33"]
AppIds	是	String	应用ID,字典表中的APP_TYPE值,多个已逗号分隔 示例值：DEFAULT
Source	否	String	渠道来源 示例值：1
SourceName	否	String	渠道来源名称 示例值：自然到店
Name	否	String	二维码名称 示例值：测试企业标签的活码
Tag.N	否	Array of WeComTagDetail	标签
SkipVerify	否	Integer	自动通过好友：0开启 1关闭, 默认开启 示例值：0
MsgId	否	Integer	欢迎语id (通过欢迎语新增返回的id) 示例值：1397103192673538050

参数名称	必选	类型	描述
Remark	否	String	备注 示例值：如图
SourceType	否	Integer	渠道类型 0 未知 1 公域 2 私域 示例值：1

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. 示例

示例1 渠道活码创建实例

输入示例

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

{
  "Remark": "如图",
  "AppIds": "DEFAULT",
  "Name": "测试企业标签的活码",
  "UseUserId": [
    1386218784692961300
  ],
  "SourceName": "自然到店",
  "MsgId": 1397103192673538050,
  "Source": "1",
  "Tag": [
    {
      "BizTagIdStr": "1394843909254754300",
      "BizTagId": 1394843909254754300,
      "BizGroupId": 1394843720913727500,
      "TagId": "etQIM2CwAA_dZ7TUzES2iRRIGXJrWY8w",
      "GroupName": "测试111",
      "TagName": "额嗡嗡嗡嗡",
      "Type": 1
    }
  ],
  "SkipVerify": 0,
  "UseUserOpenId": [
    "544064e019b53e4791115020ffc68f33"
  ],
}
```

```
"Type": 1,
"SourceType": 1
}
```

输出示例

```
{
  "Response": {
    "RequestId": "fea177dd-9fa6-4e76-9c8f-67f5a21f20bb"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

该接口暂无业务逻辑相关的错误码，其他错误码详见 [公共错误码](#)。

新增标签库接口

最近更新时间：2023-11-30 08:21:18

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

该接口用户设置标签库, 每个企业最多可配置3000个企业标签。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateCorpTag。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
GroupName	是	String	标签组名称，最长为15个字符 示例值：GROUP_NAME
Tags.N	是	Array of TagInfo	标签信息数组
Sort	否	Integer	标签组次序值。sort值大的排序靠前。有效的值范围是[0, 2^32) 示例值：1

3. 输出参数

参数名称	类型	描述
TagGroup	TagGroup	标签组信息
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. 示例

示例1 设置标签库

输入示例

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

<公共请求参数>

```
{
  "GroupName": "购车计划",
  "Sort": 1,
  "Tags": [
    {
      "TagName": "半年内",
      "Sort": 1
    },
    {
      "TagName": "一年内",
      "Sort": 2
    },
    {
      "TagName": "三年内",
      "Sort": 3
    }
  ]
}
```

输出示例

```
{
  "Response": {
    "RequestId": "79091a0a-ee46-49bb-9c22-a070b07cbe8f",
    "TagGroup": {
      "GroupName": "购车计划",
      "BizGroupId": "1407601951281504258",
      "CreateTime": 1624433431,
      "Sort": 1,
      "Tags": [
        {
          "TagName": "一年内",
          "CreateTime": 1624433431,
          "TagId": "etpqy2CAAAFrSM5y90F-pRGaiEkft--Q",
          "Sort": 2,
          "BizTagId": "1407601951327641602"
        },
        {
          "TagName": "三年内",
          "CreateTime": 1624433431,
          "TagId": "etpqy2CAAAGgssMTUt0l_JCvml-OKY8Q",
          "Sort": 3,
          "BizTagId": "1407601951327641603"
        },
        {
          "TagName": "半年内",
          "CreateTime": 1624433431,
          "TagId": "etpqy2CAAARruxu_UDmM5NuYkY27kBaw",

```

```
"Sort": 1,
"BizTagId": "1407601951327641601"
}
],
"GroupId": "etpqy2CAAAP0tGn_m9B-uTijYk7kktw"
}
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
AuthFailure.MissingAccessToken	令牌无效。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。

错误码	描述
MissingParameter	缺少参数错误。

线索回收接口

最近更新时间：2023-11-30 08:21:18

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

线索回收接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：CreateLead。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ChannelId	是	Integer	来源ID
ChannelName	是	String	来源名称 示例值：51QC
CreateTime	是	Integer	创建时间，单位毫秒 示例值：1638178594245
SourceType	是	Integer	线索类型：1-400呼入，2-常规留资 示例值：2
DealerId	是	Integer	经销商id 示例值：1438394065134600193
BrandId	是	Integer	品牌id 示例值：1373911438101237762
SeriesId	是	Integer	车系id 示例值：1376410380566495234
CustomerName	是	String	客户姓名 示例值：张三
CustomerPhone	是	String	客户手机号 示例值：134****1234
ModelId	否	Integer	车型id 示例值：1376759329958998019
CustomerSex	否	Integer	客户性别：0-未知，1-男，2-女 示例值：0

参数名称	必选	类型	描述
SalesName	否	String	销售姓名 示例值：李四
SalesPhone	否	String	销售手机号 示例值：134****6789
CcName	否	String	Cc坐席姓名 示例值：王五
Remark	否	String	备注 示例值：备注

3. 输出参数

参数名称	类型	描述
BusinessCode	Integer	线索处理状态码：0-表示创建成功，1-表示线索合并，2-表示线索重复 示例值：0
BusinessMsg	String	线索处理结果描述 示例值：创建成功
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. 示例

示例1 线索回收接口

输入示例

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

{
  "SeriesId": 1376410380566495234,
  "SalesName": "李四",
  "Remark": "备注",
  "ChannelName": "51QC",
  "DealerId": 1438394065134600193,
  "SourceType": 0,
  "CustomerSex": 0,
  "ChannelId": 1008,
  "BrandId": 1373911438101237762,
  "CustomerPhone": "手机号",
  "SalesPhone": "手机号",
  "CcName": "王五",
  "ModelId": 1376759329958998019,
  "CreateTime": 1638178594245,
```

```
"CustomerName": "张三"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "fea177dd-9fa6-4e76-9c8f-67f5a21f20bb",
    "BusinessCode": 0,
    "BusinessMsg": "创建成功"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
FailedOperation	操作失败。
FailedOperation.OpenPlatformError	开放平台访问错误。

错误码	描述
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RequestLimitExceeded	请求的次数超过了频率限制。
UnauthorizedOperation	未授权操作。

查询活动参与明细列表接口

最近更新时间：2023-11-30 08:21:18

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

根据游标拉取活动参与列表信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryActivityJoinList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ActivityId	是	Integer	活动id 示例值：1381612531432296450
Cursor	否	String	分页游标，对应结果返回的NextCursor,首次请求保持为空 示例值：bJbnlggwpL8gcUbAj2fc96iYzzeRd
Limit	否	Integer	单页数据限制 示例值：1000

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标 注意：此字段可能返回 null，表示取不到有效值。 示例值：1381458494719070210
PageData	Array of ActivityJoinDetail	活码列表响应参数 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询活动列表

输入示例

```
https://wav.tencentcloudapi.com/?Action=QueryActivityJoinList
&Limit=1
&ActivityId=1394233693086657654
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "NextCursor": "E9y1rxkoUxUW4i+Yx7ThkT7p0+bJm6IXDaCfF4etiyI=",
    "PageData": [
      {
        "ActivityData": "{\n\"cityInfo\":{\n\"cityCode\":\n\"420100\", \"cityName\":\n\"武汉市\", \"provinceCode\":\n\"420000\", \"provinceName\":\n\"湖北省\", \"carInfo\":{\n\"carId\":\n\"1334701452344614913\", \"carName\":\n\"A车型\", \"brandId\":1334701452336226306, \"tenantSeriesId\":1334701452340420610, \"baseBrandName\":\n\"大迪\", \"tenantSeriesName\":\n\"测试素材查看-车型\", \"shopInfo\":{\n\"shopId\":1392034008271540226, \"shopName\":\n\"武汉惠通陆华汽车服务有限公司\", \"cityCode\":\n\"420100\"}}},\n\"SalesName\": \"hc\", \"UserName\": \"lxc\", \"ActivityName\": \"测试未开始活动\", \"SalesPhone\": \"18976547654\", \"ActivityId\": 1394233693086657654, \"CreateTime\": 1624242265, \"LiveCodeId\": 0, \"JoinId\": 1406800137191108987, \"JoinState\": 1, \"DuplicateLeadId\": 0, \"UserPhone\": \"18045676789\", \"UpdateTime\": 1624242265, \"Duplicate\": 0, \"JoinTime\": 1624242264, \"LeadId\": 1406800137394126823\n}",
        "SalesName": "hc",
        "UserName": "lxc",
        "ActivityName": "测试未开始活动",
        "SalesPhone": "18976547654",
        "ActivityId": 1394233693086657654,
        "CreateTime": 1624242265,
        "LiveCodeId": 0,
        "JoinId": 1406800137191108987,
        "JoinState": 1,
        "DuplicateLeadId": 0,
        "UserPhone": "18045676789",
        "UpdateTime": 1624242265,
        "Duplicate": 0,
        "JoinTime": 1624242264,
        "LeadId": 1406800137394126823
      }
    ],
    "RequestId": "cee4598b-9eea-4e87-8544-7f584beb6235"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
FailedOperation.OpenPlatformError	开放平台访问错误。
InternalError	内部错误。

查询活动列表接口

最近更新时间：2023-11-30 08:21:18

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

根据游标拉取活动列表信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryActivityList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Cursor	否	String	分页游标，对应结果返回的NextCursor,首次请求保持为空 示例值：bJbnlggwpL8gcUbAj2fc96iYzzeRd
Limit	否	Integer	单页数据限制 示例值：1000

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标 注意：此字段可能返回 null，表示取不到有效值。 示例值：1381458494719070210
PageData	Array of ActivityDetail	活码列表响应参数 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询活动列表

输入示例

```
https://wav.tencentcloudapi.com/?Action=QueryActivityList
&Limit=1
```

&<公共请求参数>

输出示例

```
{
  "Response": {
    "NextCursor": "jQ+8f/xhRbMzAxSUUgYbgiYY28BzY+xFM6JuYjezhUQ=",
    "PageData": [
      {
        "ActivityState": 20,
        "ActivityName": "新建打圈互动111",
        "EndTime": 1627022340,
        "ActivityId": 1407589595607994444,
        "UpdateTime": 1624430486,
        "StartTime": 1624377600,
        "ActivityType": 101,
        "PrivacyAgreementId": "",
        "MainPhoto": "https://tmap-travel-didi-1255598736.cos.ap-guangzhou.myqcloud.com/media/92d8452b-e194-4d59-a28e-7300d0556976.jpg",
        "ActivityDataList": "{\\\"couponTypeList\\\":[1],\\\"couponInfoList\\\":[{\\\"couponId\\\":\\\"C1407254077494390786\\\",\\\"couponName\\\":\\\"自动化创建满减券161480\\\",\\\"thresholdAmount\\\":15000000,\\\"discountAmount\\\":15000000,\\\"discountPercentage\\\":0,\\\"couponType\\\":1,\\\"availableTimeType\\\":1,\\\"availableBeginTime\\\":0,\\\"availableEndTime\\\":10368000}]}\"
      }
    ],
    "RequestId": "d143e1fc-1106-4b3e-a9cd-854b09b51752"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
FailedOperation.OpenPlatformError	开放平台访问错误。
InternalError	内部错误。

查询活动活码列表接口

最近更新时间：2023-11-30 08:21:17

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

根据游标拉取活动活码列表信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryActivityLiveCodeList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
PageData	Array of LiveCodeDetail	活码列表响应参数 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询活码列表

输入示例

```
https://wav.tencentcloudapi.com/?Action=QueryActivityLiveCodeList
&Cursor=eKMxI/XqG5yQmGrx3qXBRthr/F8JGlknuYJTKHZaks=
```

```
&Limit=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "NextCursor": "eKMXI/XqG5yQmGrx3qXBRrthr/F8JGiknuYJTKHZaks=",
    "PageData": [
      {
        "LiveCodePreview": "https://tmap-travel-didi-1255598736.cos.ap-guangzhou.myqcloud.com/liveCode/ww81084a80e7663baf/1405537314006573058.jpg",
        "ActivityName": "10日券",
        "ActivityId": 1405537288119328800,
        "CreateTime": 1623941184,
        "ShortChainAddress": "https://aggretrip.map.qq.com/wework_saas/dist/index.html?wexin_target=weixin%3A%2F%2Fdl%2Fbusiness%2F%3Ft%3DreO6Co5q6ko",
        "UpdateTime": 1623941184,
        "LiveCodeData": "",
        "LiveCodeState": 0,
        "LiveCodeName": "22",
        "LiveCodeId": 1405537311796174800
      }
    ],
    "RequestId": "e520d8eb-969f-4554-bf5f-169b344935e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
InternalError	内部错误。

查询渠道活码列表接口

最近更新时间：2023-11-30 08:21:17

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

根据游标拉取渠道活码列表信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryChannelCodeList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：1397103193143300098=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
PageData	Array of ChannelCodeInnerDetail	活码列表响应参数 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询活码列表

输入示例

```
https://wav.tencentcloudapi.com/?Action=QueryChannelCodeList
&Limit=1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "NextCursor": "iKJ7mI3Itna2Hs+BMN02l6F6RojbCkeHgCBKcY0Qe+s=",
    "PageData": [
      {
        "Friends": 0,
        "QrCodeUrl": "https://wework.qpic.cn/wwpic/182442_klur6CpKTWe-bVG_1623808245/0",
        "SourceName": "公域车迅达",
        "MsgId": 1404979725786083311,
        "Source": "1001",
        "Remark": "",
        "Name": "erwef",
        "Type": 1,
        "UseUserOpenIdList": [
          "ShiHao"
        ],
        "AppId": "LXC",
        "UseUserIdList": [
          1372482008858415111
        ],
        "TagList": [],
        "ConfigId": "f25675c3b271f41f28274c1c7e729f16",
        "Id": 1404979727371530211,
        "SkipVerify": 0,
        "RecStatus": 0
      }
    ],
    "RequestId": "801fbb9d-00ad-4990-b0c9-fdb3e4e48073"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
InternalServerError	内部错误。

查询会话存档列表接口

最近更新时间：2023-11-30 08:21:17

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

根据游标拉取会话存档列表信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryChatArchivingList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
PageData	Array of ChatArchivingDetail	会话存档列表响应参数 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询会话存档列表

输入示例

```
https://wav.tencentcloudapi.com/?Action=QueryChatArchivingList
&Cursor=+1H24tK0tELjSiTOR10DzA==
```

&Limit=10
&<公共请求参数>

输出示例

```
{
  "Response": {
    "PageData": [
      {
        "MsgId": "1312616990135121111",
        "Action": "send",
        "MsgType": "video",
        "From": "YangYan",
        "ToList": [
          "YangGuo"
        ],
        "RoomId": "100001",
        "MsgTime": "1616990135",
        "Video": {
          "PlayLength": 4,
          "FileSize": 645990,
          "CosKey": "http://test"
        }
      }
    ],
    "NextCursor": "+1H24tK0tELjSiTOR10DzA==",
    "RequestId": "4d48312c-a062-4875-a5d5-69f0f11baf96"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
InternalError	内部错误。

查询CRM统计数据接口

最近更新时间：2023-11-30 08:21:17

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

通过接口拉取租户/指定成员/部门在指定日期范围内的CRM跟进统计数据

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryCrmStatistics。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
BeginTime	是	Integer	查询开始时间，单位秒 示例值：1647187200
EndTime	是	Integer	查询结束时间，单位秒 示例值：1647273600
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100
SalesId	否	String	请求的企业成员id，为空时默认全租户 示例值：1323253932850728968
OrgId	否	Integer	请求的部门id，为空时默认全租户 示例值：3

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。
PageData	Array of CrmStatisticsData	CRM统计响应数据 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询CRM统计数据接口

输入示例

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

{
  "SalesId": "xx",
  "Cursor": "xx",
  "Limit": 0,
  "BeginTime": 1,
  "OrgId": 1,
  "EndTime": 1
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "+1H24tK0tELjSiTOR10DzA==",
    "PageData": [
      {
        "LeadCnt": 0,
        "BuildCnt": 0,
        "InvitedCnt": 0,
        "OrderedCnt": 0,
        "DeliveredCnt": 0,
        "DefeatCnt": 0,
        "NewContactCnt": 0,
        "StatisticalTime": "String"
      }
    ],
    "RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

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

查询外部联系人SaaS使用明细数据接口

最近更新时间：2023-11-30 08:21:16

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

通过接口拉取SaaS内C端外部联系人在指定时间范围内的行为事件明细。此接口提供的数据以天为维度，查询的时间范围为[start_time,end_time]，即前后均为闭区间，支持的最大查询跨度为365天。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryCustomerEventDetailStatistics。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
BeginTime	是	Integer	查询开始时间，单位秒 示例值：1647273600
EndTime	是	Integer	查询结束时间，单位秒 示例值：1647573600
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5l2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。
PageData	Array of CustomerActionEventDetail	外部联系人SaaS使用明细统计响应数据 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询外部联系人SaaS使用明细数据接口

输入示例

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

{
  "Cursor": "xx",
  "Limit": 0,
  "BeginTime": 1647273600,
  "EndTime": 1647573600
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "+1H24tK0tELjSiTOR10DzA==",
    "PageData": [
      {
        "EventCode": "1000",
        "EventType": 1,
        "EventSource": 1,
        "MaterialType": 1,
        "MaterialId": 1348495383617208321,
        "ExternalUserId": "wmQIM2CwAAL1ZKqK-icQG9XAMa3AhP-A",
        "EventTime": 1647573600,
        "SalesId": 1323253932850728968
      }
    ],
    "RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

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

查询外部联系人详情接口

最近更新时间：2023-11-30 08:21:16

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

企业可通过此接口，根据外部联系人的userid，拉取外部联系人详情

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryExternalContactDetail。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ExternalUserId	是	String	外部联系人的userid，注意不是企业成员的账号 示例值：woAJ2GCAAAXtWyujaWJHDDGi0mACHAAA
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填。当客户在企业内的跟进人超过500人时需要使用cursor参数进行分页获取 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	当前接口Limit不需要传参，保留Limit只是为了保持向后兼容性，Limit默认值为500，当返回结果超过500时，NextCursor才有返回值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Customer	ExternalContact	客户信息 注意：此字段可能返回 null，表示取不到有效值。
FollowUser	Array of FollowUser	添加了此外部联系人的企业成员信息 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询外部联系人详情

通过外部联系人ID获取相应的详情信息

输入示例

```
https://wv.tencentcloudapi.com/?Action=QueryExternalContactDetail
&ExternalUserId=2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw
&Cursor=sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxIm/oLU=
&Limit=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "NextCursor": "abc",
    "Customer": {
      "ExternalUserId": "abc",
      "Gender": 0,
      "Name": "abc",
      "Type": 1,
      "UnionId": "abc",
      "Phone": "abc"
    },
    "FollowUser": [
      {
        "UserId": "abc",
        "Remark": "abc",
        "Description": "abc",
        "CreateTime": 1,
        "AddWay": 0,
        "OperUserId": "abc",
        "Tags": [
          {
            "GroupName": "abc",
            "TagName": "abc",
            "Type": 1,
            "TagId": "abc"
          }
        ]
      },
      {
        "RequestId": "abc"
      }
    ]
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
FailedOperation	操作失败。
FailedOperation.OpenPlatformError	开放平台访问错误。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RequestLimitExceeded	请求的次数超过了频率限制。

查询外部联系人列表接口

最近更新时间：2023-11-30 08:21:15

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

企业可通过此接口基于外部联系人获取指定成员添加的客户列表。客户是指配置了客户联系功能的成员所添加的外部联系人。没有配置客户联系功能的成员，所添加的外部联系人将不会作为客户返回。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryExternalContactList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5l2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
PageData	Array of ExternalContactSimpleInfo	外部联系人信息列表 注意：此字段可能返回 null，表示取不到有效值。
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5l2MVw
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. 示例

示例1 查询外部联系人列表

输入示例

```
https://wav.tencentcloudapi.com/?Action=QueryExternalContactList
&Cursor=sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxlm/oLU=
&Limit=1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "NextCursor": "sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxlm/oLU=",
    "PageData": [
      {
        "SalesName": "杨过",
        "UserId": "YangGuo",
        "ExternalUserId": "wmpqy2CAAATGwpQTxuU1IUfoiOFH2cXA",
        "DepartmentIdList": [
          0,
          1
        ]
      }
    ],
    "RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

• [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

查询外部联系人事件明细列表接口

最近更新时间：2023-11-30 08:21:15

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

通过接口拉取租户在指定时间范围内的外部联系人添加/删除明细，此接口提供的数据以天为维度，查询的时间范围为[StarTime, EndTime]，即前后均为闭区间，支持的最大查询跨度为365天；

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryExternalUserEventList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
BeginTime	是	Integer	查询开始时间，单位秒 示例值：1647187200
EndTime	是	Integer	查询结束时间，单位秒 示例值：1647273600
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
PageData	Array of ExternalUserEventInfo	外部联系人事件信息响应数据 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询外部联系人事件明细列表接口

输入示例

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

{
  "Cursor": "+1H24tK0tELjSiTOR10DzA==",
  "Limit": 100,
  "BeginTime": 1647187200,
  "EndTime": 1647273600
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "+1H24tK0tELjSiTOR10DzA==",
    "PageData": [
      {
        "EventCode": "ADD_EXTERNAL_CUSTOMER",
        "ExternalUserId": "woQIM2CwAA8BPLSVDDDBZUYIiviX3_9DQ",
        "SalesId": "1323253932850728968",
        "EventTime": 1614923098
      }
    ],
    "RequestId": "4d48312c-a062-4875-a5d5-69f0f11baf96"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [GitHub](#) [Gitee](#)
- Tencent Cloud SDK 3.0 for Java: [GitHub](#) [Gitee](#)

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
AuthFailure.MissingAccessToken	令牌无效。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TimeSpanLimitExceeded	参数值时间跨度超出接口限制。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
RequestLimitExceeded	请求的次数超过了频率限制。

外部联系人转换接口

最近更新时间：2023-11-30 08:21:15

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

企业可通过此接口将企业主体对应的外部联系人id转换为乐销车应用主体对应的外部联系人。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryExternalUserMappingInfo。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
CorpExternalUserIdList.N	是	Array of String	企业主体对应的外部联系人id列表，列表长度限制最大为50。 示例值：["wmQIM2CwAAywHqH07voRw9aacKXf09eQ"]

3. 输出参数

参数名称	类型	描述
ExternalUserIdMapping	Array of ExternalUserMappingInfo	外部联系人映射信息，只返回映射成功的记录 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 外部联系人转换接口

输入示例

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

```
{
  "CorpExternalUserIdList": [
    "wmQIM2CwAAyWqH07voRw9aacKXf09eQ",
    "wmAA2CCwAAyWqH07voRw9aacAFEewf"
  ]
}
```

输出示例

```
{
  "Response": {
    "ExternalUserIdMapping": [
      {
        "CorpExternalUserId": "wmabc2CAAATGwpQTxuU1laaiOFH2cXA",
        "ExternalUserId": "wmpqy2CAAATGwpQTxuU1IufioFH2cXA"
      }
    ],
    "RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

查询license信息接口

最近更新时间：2023-11-30 08:21:15

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

该接口获取license对应的详细信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryLicenseInfo。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
License	是	String	license编号 示例值：afeafefefe1

3. 输出参数

参数名称	类型	描述
LicenseInfo	LicenseInfo	license响应信息 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询license信息接口

输入示例

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

{
```

```
"License": "A1c0tVAAAGSB1"
}
```

输出示例

```
{
  "Response": {
    "LicenseInfo": {
      "DestroyTime": "2022-07-15 00:00:00",
      "ResourceEndTime": "2022-07-01 00:00:00",
      "Extra": "",
      "IsolationDeadline": "2022-07-08 00:00:00",
      "License": "A1c0tVAAAGSB1",
      "LicenseEdition": 1,
      "ResourceStartTime": "2021-07-01 00:00:00",
      "Status": 1
    },
    "RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

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

获取企业素材列表接口

最近更新时间：2023-11-30 08:21:14

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

通过接口按类型拉取租户当前的素材列表及关键信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryMaterialList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
MaterialType	是	Integer	素材类型：0-图片，1-视频，3-文章，10-车型，11-名片 示例值：0
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
PageData	Array of MaterialInfo	企业素材列表响应数据 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 获取企业素材列表接口

输入示例

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

{
  "Cursor": "+1H24tK0tELjSiTOR10DzA==",
  "Limit": 100,
  "MaterialType": 0
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "+1H24tK0tELjSiTOR10DzA==",
    "PageData": [
      {
        "MaterialId": 0,
        "MaterialName": "素材名称",
        "Status": 0
      }
    ],
    "RequestId": "4d48312c-a062-4875-a5d5-69f0f11baf96"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
AuthFailure.MissingAccessToken	令牌无效。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TimeSpanLimitExceeded	参数值时间跨度超出接口限制。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
RequestLimitExceeded	请求的次数超过了频率限制。

查询小程序码列表接口

最近更新时间：2023-11-30 08:21:14

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

查询小程序码列表接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryMiniAppCodeList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
PageData	Array of MiniAppCodeInfo	小程序码列表响应数据 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询小程序码列表

输入示例

```
https://wav.tencentcloudapi.com/?Action=QueryMiniAppCodeList
&Cursor=+1H24tK0tELjSiTOR10DzA==
```

```
&Limit=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "NextCursor": "+1H24tK0tELjSiTOR10DzA==",
    "PageData": [
      {
        "MiniAdminUrl": "https://weworksaas-web-dev.map.qq.com/",
        "State": 1,
        "CreateTime": 1618556502,
        "UpdateTime": 1618556507,
        "MiniAppName": "官方小程序",
        "MiniAppLogo": "https://tmap-travel-didi-1255598736.cos.ap-guangzhou.myqcloud.com/media/9b2bfb85-9d3a-41d2-a6cb-2ac95b11af68.jpg",
        "Id": 10
      }
    ],
    "RequestId": "4d48312c-a062-4875-a5d5-69f0f11baf96"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
AuthFailure.MissingAccessToken	令牌无效。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。

企业成员SaaS使用明细数据接口

最近更新时间：2023-11-30 08:21:14

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

通过接口拉取SaaS内企业成员在指定时间范围内的行为事件明细。此接口提供的数据以天为维度，查询的时间范围为[start_time,end_time]，即前后均为闭区间，支持的最大查询跨度为365天。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryStaffEventDetailStatistics。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
BeginTime	是	Integer	查询开始时间，单位秒 示例值：1647273600
EndTime	是	Integer	查询结束时间，单位秒 示例值：1647573600
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5l2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。
PageData	Array of SalesActionEventDetail	企业成员SaaS使用明细统计响应数据 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询企业成员SaaS使用明细数据接口

输入示例

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

{
  "Cursor": "xx",
  "Limit": 0,
  "BeginTime": 1647273600,
  "EndTime": 1647573600
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "+1H24tK0tELjSiTOR10DzA==",
    "PageData": [
      {
        "EventCode": "1000",
        "EventType": 1,
        "EventSource": 1,
        "MaterialType": 1,
        "MaterialId": 1348495383617208321,
        "EventTime": 1647573600,
        "SalesId": 1323253932850728968
      }
    ],
    "RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

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

查询企业成员信息列表接口

最近更新时间：2023-11-30 08:21:14

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

查询企业成员信息列表接口

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryUserInfoList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	否	String	公共参数 ，本接口不需要传递此参数。
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，再下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
PageData	Array of CorpUserInfo	企业成员信息列表响应数据 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询企业成员信息列表

查询企业成员信息列表

输入示例

```
https://wav.tencentcloudapi.com/?Action=QueryUserInfoList
&Cursor=+1H24tK0tELjSiTOR10DzA==
&Limit=10
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "NextCursor": "+1H24tK0tELjSiTOR10DzA==",
    "PageData": [
      {
        "UserId": 0,
        "UserName": "杨过",
        "UserOpenId": "YangGuo",
        "DealerId": 0,
        "ShopId": 0,
        "Phone": "134****1234",
        "OrgIds": "2,3",
        "MainDepartment": "1",
        "IsLeaderInDept": "1,0",
        "Status": 0,
        "JobNumber": "12345678"
      }
    ],
    "RequestId": "4d48312c-a062-4875-a5d5-69f0f11baf96"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
AuthFailure.MissingAccessToken	令牌无效。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。

查询车系车型信息列表接口

最近更新时间：2023-11-30 08:21:14

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

企业可通过此接口获取企微SaaS平台上的车系车型信息。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryVehicleInfoList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
PageData	Array of VehicleInfo	车系车型信息列表 注意：此字段可能返回 null，表示取不到有效值。
NextCursor	String	分页游标，下次调用带上该值，则从当前的位置继续往后拉取新增的数据，以实现增量拉取。 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw
HasMore	Integer	是否还有更多数据。0-否；1-是。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询车系车型信息列表接口

输入示例

```
https://wav.tencentcloudapi.com/?Action=QueryVehicleInfoList
&Cursor=sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxlm/oLU=
&Limit=1
&<公共请求参数>
```

输出示例

```
{
  "Response": {
    "NextCursor": "sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxlm/oLU=",
    "HasMore": 1,
    "PageData": [
      {
        "BrandId": 1348081115398452226,
        "BrandName": "品牌名称",
        "SeriesId": 1348080105398452333,
        "SeriesName": "车系名称",
        "ModelId": 1348084565398452121,
        "ModelName": "车型名称"
      }
    ],
    "RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

• [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。

通过时间段查询外部联系人详情列表

最近更新时间：2023-11-30 08:21:16

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

企业可通过传入起始和结束时间，获取该时间段的外部联系人详情列表

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryExternalContactDetailByDate。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
BeginTime	是	Integer	查询结束时间，单位秒 示例值：1672502400
EndTime	是	Integer	查询结束时间，单位秒 示例值：1672588800
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，下次请求时填写以获取之后分页的记录，如果已经没有更多的数据则返回空 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
PageData	Array of ExternalContactDetailPro	外部联系人详细信息
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. 示例

示例1 查询指定时间内外部联系人详情

通过时间段查询外部联系人详情

输入示例

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

{
  "Cursor": "2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=",
  "Limit": 100,
  "BeginTime": 1647187200,
  "EndTime": 1647273600
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=",
    "PageData": [
      {
        "Customer": {
          "ExternalUserId": "wmQIM2CwAAE-JSOVGMzsRayI3ow5kGzQ",
          "Gender": 1,
          "Name": "文",
          "Type": 1,
          "UnionId": "ozynqsulJFCZ2z1aYeS8h-nuasdAAA",
          "Phone": "134****3232"
        },
        "FollowUser": [
          {
            "UserId": "rocky",
            "Remark": "备注",
            "Description": "描述",
            "CreateTime": 1618322621,
            "AddWay": 0,
            "OperUserId": "wmQIM2CwAAE-JSOVGMzsRayI3ow5kGzQ",
            "Tags": [
              {
                "GroupName": "标签分组名称",
                "TagName": "标签名称",
                "Type": 1,
                "TagId": "etAJ2GCAAAXtWyujaWJHDDGi0mACHAAA"
              }
            ]
          }
        ]
      }
    ]
  }
}
```

```
"SalesName": "张三",
"DepartmentIdList": [
  0
]
}
]
}
],
"RequestId": "fea177dd-9fa6-4e76-9c8f-67f5a21f20bb"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
InternalError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

错误码	描述
LimitExceeded	超过配额限制。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RequestLimitExceeded	请求的次数超过了频率限制。

查询潜在客户档案列表接口

最近更新时间：2023-11-30 08:21:16

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

通过接口拉取租户已有潜在客户档案列表信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryCustomerProfileList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Limit	是	Integer	分页，预期请求的数据量，取值范围 1 ~ 1000 示例值：100
BeginTime	是	Integer	查询开始时间，单位秒 示例值：1647187200
EndTime	是	Integer	查询结束时间，单位秒 示例值：1647273600
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，下次调用带上该值，则从当前的位置继续往后拉，以实现增量拉取。 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw
PageData	Array of CustomerProfile	潜在客户存档信息 注意：此字段可能返回 null，表示取不到有效值。
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询潜在客户存档信息

用于查询潜客客户存档信息

输入示例

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

{
  "Cursor": "2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=",
  "Limit": 100,
  "BeginTime": 1647187200,
  "EndTime": 1647273600
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=",
    "PageData": [
      {
        "CustomerId": 1234567890,
        "DealerCode": "abc",
        "UnionId": "ozynqsulJFCZ2z1aYeS8h",
        "CreateTime": "2023-01-01 00:00:00",
        "UserName": "张三",
        "Gender": 1,
        "Phone": "123***4567",
        "AgeRangeName": "26-30",
        "JobTypeName": "abc",
        "Address": "地址1",
        "LeadsProcessStatus": 0,
        "LeadType": 1,
        "SourceName": "自然到店",
        "LeadsLevelCode": "101",
        "VehicleBrandCode": "品牌编号",
        "VehicleSeriesCode": "车系编号",
        "VehicleTypeCode": "车型编号",
        "VehiclePurpose": {
          "VehiclePurposeCode": "abc",
          "VehiclePurposeName": "abc"
        },
        "PurchaseConcern": [
          {
            "Code": "abc",
            "Description": "abc"
          }
        ]
      }
    ]
  }
}
```

```
],
"SalesName": "张三",
"SalesPhone": "123****4567",
"RealArrivalTime": 1618322621,
"CompleteTestDriveTime": "2023-01-01 00:00:00",
"OrderTime": 1618322621,
"DeliveryTime": 1618322621,
"InvoiceTime": 1618322621,
"LoseTime": 1618322621,
"CreatedAtTime": 1618322621,
"ImportAtTime": 1618322621,
"DistributeTime": 1618322621,
"LeadCreateTime": 1618322621,
"Nickname": "jack",
"OrgIdList": [
  "1"
],
"Introducer": "李四",
"IntroducerPhone": "123****4567",
"FollowTime": 1618322621,
"NextFollowTime": 1618322621,
"EnterpriseTags": [
  {
    "GroupName": "标签分组名称",
    "TagName": "标签名称",
    "TagId": "etAJ2GCAAAXtWyujaWJHDDGi0mACHAAA"
  }
],
"ChannelTags": [
  {
    "TagName": "abc",
    "TagId": "etAJ2GCAAAXtWyujaWJHDDGi0mACHAAA"
  }
],
"LeadId": 1348080105398452200,
"Wxid": "李四_1234",
"Position": "销售",
"IsBindWx": 1,
"IsInvalid": 1,
"InvalidType": "001",
"InvalidTypeName": "无效",
"InvalidTime": 1618322621,
"InvalidRemark": "无效",
"IsLose": 1,
"LoseType": "001",
"LoseTypeName": "战败",
"LoseRemark": "战败"
},
"RequestId": "fea177dd-9fa6-4e76-9c8f-67f5a21f20bb"
```

```
}  
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
FailedOperation.OpenPlatformError	开放平台访问错误。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RequestLimitExceeded	请求的次数超过了频率限制。
UnknownParameter	未知参数错误。

查询指定时间范围内发生过跟进的潜在客户信息

最近更新时间：2023-11-30 08:21:15

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

查询指定时间范围内发生过跟进的潜在客户信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryFollowList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Limit	是	Integer	分页，预期请求的数据量，取值范围 1 ~ 1000 示例值：100
BeginTime	是	Integer	查询开始时间，单位秒 示例值：1647187200
EndTime	是	Integer	查询结束时间，单位秒 示例值：1647273600
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，下次调用带上该值，则从当前的位置继续往后拉，以实现增量拉取。 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw
PageData	Array of FollowInfo	潜在客户存档信息 注意：此字段可能返回 null，表示取不到有效值。
HasMore	Integer	是否还有更多数据。0-否；1-是。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询指定时间范围内发生过跟进的潜在客户信息

查询指定时间范围内发生过跟进的潜在客户信息

输入示例

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

{
  "Cursor": "sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxIm/oLU=",
  "Limit": 100,
  "BeginTime": 1647187200,
  "EndTime": 1647187200
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxIm/oLU=",
    "PageData": [
      {
        "ClueId": 1348080105398452200,
        "CustomerId": 1348080105398452200,
        "UserName": "张三",
        "Phone": "134xxxx1234",
        "IsOverdue": 0,
        "OverdueTime": 1618322621,
        "EventType": 1,
        "EventTypeName": "逾期",
        "FollowWayType": "1",
        "FollowWayName": "自然逾期",
        "FollowTime": 1618322621,
        "NextFollowTime": 1618322621,
        "FollowRecord": "abc"
      }
    ],
    "HasMore": 1,
    "RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
FailedOperation.OpenPlatformError	开放平台访问错误。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RequestLimitExceeded	请求的次数超过了频率限制。
UnknownParameter	未知参数错误。

查询经销商信息列表接口

最近更新时间：2023-11-30 08:21:16

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

企业可通过此接口获取录入在企微SaaS平台上的经销商信息。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryDealerInfoList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100

3. 输出参数

参数名称	类型	描述
PageData	Array of DealerInfo	经销商信息列表 注意：此字段可能返回 null，表示取不到有效值。
NextCursor	String	分页游标，下次调用带上该值，则从当前的位置继续往后拉取新增的数据，以实现增量拉取。 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw
HasMore	Integer	是否还有更多数据。0-否；1-是。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询经销商信息列表接口

查询经销商信息列表

输入示例

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

{
  "Cursor": "sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxIm/oLU=",
  "Limit": 100
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxIm/oLU=",
    "HasMore": 1,
    "PageData": [
      {
        "DealerId": 1348080105398452200,
        "DealerName": "经销商名称",
        "ProvinceCode": "110100",
        "CityCodeList": [
          "110100"
        ],
        "BrandIdList": [
          "1"
        ]
      }
    ],
    "RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。

查询线索列表接口

最近更新时间：2023-11-30 08:21:17

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

企业可通过此接口获取线索列表。

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryClueInfoList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=
Limit	否	Integer	返回的最大记录数，整型，最大值100，默认值50，超过最大值时取最大值 示例值：100
BeginTime	否	Integer	查询开始时间，单位秒 示例值：1647187200
EndTime	否	Integer	查询结束时间，单位秒 示例值：1647273600

3. 输出参数

参数名称	类型	描述
PageData	Array of ClueInfoDetail	线索信息列表 注意：此字段可能返回 null，表示取不到有效值。
NextCursor	String	分页游标，下次调用带上该值，则从当前的位置继续往后拉，以实现增量拉取。 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw
HasMore	Integer	是否还有更多数据。0-否；1-是。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询线索列表接口

用于查询线索列表

输入示例

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

{
  "Cursor": "sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxIm/oLU=",
  "Limit": 100,
  "BeginTime": 1647187200,
  "EndTime": 1647273600
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "sHmJbhJZhCwwMYmSot50dl2Hs9qQvbP+pCUVxIm/oLU=",
    "HasMore": 1,
    "PageData": [
      {
        "DealerId": "1,3,5",
        "ClueId": "1348080105398452226",
        "EnquireTime": 1618556502,
        "UnionId": "wmpqy2CAAATGwpQTxuU1IUfoiOFH2cXA",
        "Name": "微信昵称",
        "UserName": "李四",
        "Phone": "134xxxx1234",
        "SeriesCode": "车系编号",
        "ModelCode": "车型编号",
        "ProvinceCode": "省份编号",
        "CityCode": "城市编号",
        "SalesName": "顾问名称",
        "SalesPhone": "134xxxx6789",
        "Remark": "备注",
        "TagList": [
          "标签1",
          "标签2"
        ],
        "LeadUserType": 0,
        "LeadType": 1,
        "ChannelId": 1,
        "ChannelName": "渠道1",
      }
    ]
  }
}
```

```
"SourceChannelName": "渠道1",
"Gender": 1,
"CreateTime": "2023-01-01 00:00:00",
"UpdateTime": "2023-01-01 00:00:00",
"LeadStatus": 101,
"LevelCode": "101",
"ImportAtTime": 1618322621,
"DistributeTime": 1618322621,
"CreateAtTime": 1618322621,
"WxId": "李四_1234",
"BrandCode": "abc",
"BuildTime": 1618322621,
"OrderTime": 1618322621,
"ArrivalTime": 1618322621,
"DeliveryTime": 1618322621,
"FollowTime": 1618322621,
"NextFollowTime": 1618322621,
"OrgId": 1,
"OrgName": "org1",
"Introducer": "introducer",
"IntroducerPhone": "134xxxx1234",
"IsBindWx": 0,
"IsMerge": 0,
"IsInvalid": 0,
"InvalidType": "1",
"InvalidTypeName": "空错号",
"InvalidRemark": "无效记录",
"InvalidTime": 1618322621,
"DealerName": "dealer1",
"ShopId": 1234567890,
"CorpShopId": "S1234",
"ShopName": "shop1",
"Position": "销售"
}
],
"RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
FailedOperation.OpenPlatformError	开放平台访问错误。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
LimitExceeded	超过配额限制。

查询指定时间范围内发生过到店的潜客到店信息

最近更新时间：2023-11-30 08:21:17

1. 接口描述

接口请求域名：wav.tencentcloudapi.com。

查询指定时间范围内发生过到店的潜客到店信息

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：QueryArrivalList。
Version	是	String	公共参数 ，本接口取值：2021-01-29。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Limit	是	Integer	分页，预期请求的数据量，取值范围 1 ~ 1000 示例值：100
BeginTime	是	Integer	查询开始时间，单位秒 示例值：1647187200
EndTime	是	Integer	查询结束时间，单位秒 示例值：1647273600
Cursor	否	String	用于分页查询的游标，字符串类型，由上一次调用返回，首次调用可不填 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw=

3. 输出参数

参数名称	类型	描述
NextCursor	String	分页游标，下次调用带上该值，则从当前的位置继续往后拉，以实现增量拉取。 注意：此字段可能返回 null，表示取不到有效值。 示例值：2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw
PageData	Array of ArrivalInfo	潜客客户存档信息 注意：此字段可能返回 null，表示取不到有效值。
HasMore	Integer	是否还有更多数据。0-否；1-是。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RequestId	String	唯一请求 ID，每次请求都会返回。定位问题时需要提供该次请求的 RequestId。

4. 示例

示例1 查询指定时间范围内发生过到店的潜客到店信息

查询指定时间范围内发生过到店的潜客到店信息

输入示例

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

{
  "Cursor": "2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw",
  "Limit": 100,
  "BeginTime": 1618322621,
  "EndTime": 1618322621
}
```

输出示例

```
{
  "Response": {
    "NextCursor": "2Kgdu51quLaeNJSIwb79im883bftBE9d9Emju5I2MVw",
    "PageData": [
      {
        "ClueId": 1348080105398452200,
        "CustomerId": 1348080105398452200,
        "UserName": "张三",
        "Phone": "134xxxx4321",
        "FirstArrival": 1,
        "ArrivalTime": 1618322621,
        "EventType": 1,
        "EventTypeName": "首次到店",
        "FollowRecord": ""
      }
    ],
    "HasMore": 1,
    "RequestId": "b1a024bf-4d74-4b5d-a5bd-bbec330520e8"
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

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

命令行工具

- [Tencent Cloud CLI 3.0](#)

6. 错误码

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

错误码	描述
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
FailedOperation.OpenPlatformError	开放平台访问错误。
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
MissingParameter	缺少参数错误。
OperationDenied	操作被拒绝。
RequestLimitExceeded	请求的次数超过了频率限制。
UnknownParameter	未知参数错误。

数据结构

最近更新时间：2023-08-17 06:34:02

ActivityDetail

活动详情

被如下接口引用：QueryActivityList。

名称	类型	描述
ActivityId	Integer	活动id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1381619428653346818
ActivityName	String	活动名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：留资活动
ActivityState	Integer	活动状态，10:未开始状态、20:已开始（进行中）状态、30:已结束状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：10
ActivityType	Integer	活动类型，100:留资活动 注意：此字段可能返回 null，表示取不到有效值。 示例值：100
StartTime	Integer	活动开始时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618243200
EndTime	Integer	活动结束时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618329600
MainPhoto	String	活动主图 注意：此字段可能返回 null，表示取不到有效值。 示例值：https://tmap-travel-didi-1255598736.cos.ap-guangzhou.myqcloud.co
PrivacyAgreementId	String	协议编号 注意：此字段可能返回 null，表示取不到有效值。 示例值：1380815131099852801
UpdateTime	Integer	活动更新时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1621500202
ActivityDataList	String	活动数据列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"optionalCityCodeList": [

ActivityJoinDetail

活动参与详情

被如下接口引用：QueryActivityJoinList。

名称	类型	描述
----	----	----

名称	类型	描述
ActivityId	Integer	活动id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1381619428653346818
ActivityName	String	活动名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：留资活动
SalesName	String	销售姓名 注意：此字段可能返回 null，表示取不到有效值。 示例值：销售1
SalesPhone	String	销售电话 注意：此字段可能返回 null，表示取不到有效值。 示例值：13000000000
JoinId	Integer	参与id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1381612531432296450
LiveCodeId	Integer	活码id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1381612531432296450
UserPhone	String	用户电话 注意：此字段可能返回 null，表示取不到有效值。 示例值：13000000000
UserName	String	用户姓名 注意：此字段可能返回 null，表示取不到有效值。 示例值：用户1
ActivityData	String	活动数据 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"cityInfo":{"cityCode":"420100","cityName":"武汉市"},"p
LeadId	Integer	线索id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1381612531432296450
JoinTime	Integer	参与时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618329600
Duplicate	Integer	线索是否是重复创建， 0：新建、1：合并、2：重复， 默认为0 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
DuplicateLeadId	Integer	重复线索id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1391954669730611201
JoinState	Integer	是否为参与多次活动， 1：参与一次、2：参与多次，默认为0 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
CreateTime	Integer	创建时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618329600

名称	类型	描述
UpdateTime	Integer	更新时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618329600

ArrivalInfo

发生过到店的潜客到店信息

被如下接口引用：QueryArrivalList。

名称	类型	描述
Clueld	Integer	线索id 示例值：1348080105398452226
CustomerId	Integer	客户id 示例值：1348080105398452226
UserName	String	客户姓名 示例值：张三
Phone	String	客户的手机号 示例值：134xxxx4321
FirstArrival	Integer	是否首次到店，0否，1是 示例值：1
ArrivalTime	Integer	到店时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
EventType	Integer	发生事件 示例值：1
EventTypeName	String	发生事件名称 示例值：首次到店
FollowRecord	String	跟进记录

ChannelCodeInnerDetail

渠道活码详情

被如下接口引用：QueryChannelCodeList。

名称	类型	描述
Id	Integer	渠道活码id 示例值：1397103193143300098
Type	Integer	欢迎语类型，0：普通欢迎语、1:渠道欢迎语 示例值：1
Source	String	渠道来源 示例值：1
SourceName	String	渠道来源名称 示例值：自然到店

名称	类型	描述
Name	String	二维码名称 示例值：测试企业标签的活码
UseUserIdList	Array of Integer	使用成员用户id集 示例值：[1386218784692961300]
UseUserOpenIdList	Array of String	使用成员企微账号id集 示例值：["544064e019b53e4791115020ffc68f33"
TagList	Array of WeComTagDetail	标签 示例值：[{ "GroupName": "测试
SkipVerify	Integer	自动通过好友，0：开启、1：关闭，默认0开启 示例值：0
Friends	Integer	添加好友人数 示例值：0
Remark	String	备注 示例值：备注
MsgId	Integer	欢迎语id（通过欢迎语新增返回的id） 示例值：1397103192673538050
ConfigId	String	联系我config_id 示例值：13ef26feea8c92e2b0c895db45987fb9
QrCodeUrl	String	联系我二维码地址 示例值： https://wework.qpic.cn/wwwpic/142302_fc9KKS9RSyS58cp_1621930333/0
RecStatus	Integer	记录状态，0：有效、1：无效 示例值：0
AppId	String	应用ID 示例值：LXC

ChannelTag

客户渠道标签

被如下接口引用：QueryCustomerProfileList。

名称	类型	描述
TagName	String	该客户档案当前已成功关联的渠道标签名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：标签名称
TagId	String	该客户档案当前已成功关联的渠道标签的id 注意：此字段可能返回 null，表示取不到有效值。 示例值：etAJ2GCAAAXtWyujaWJHDDGi0mACHAAA

ChatArchivingDetail

会话存档数据详情

被如下接口引用：QueryChatArchivingList。

名称	类型	描述
MsgId	String	消息id
Action	String	动作名称，switch表示切换企微账号，send表示企微普通消息 示例值：switch, send
MsgType	String	消息类型，当Action != "switch"时存在，例如video, text, voice 等，和企微开放文档一一对应 https://open.work.weixin.qq.com/api/doc/90000/90135/91774 注意：此字段可能返回 null，表示取不到有效值。 示例值：video, text, voice
From	String	消息发送人 注意：此字段可能返回 null，表示取不到有效值。
ToList	Array of String	消息接收人列表，注意接收人可能只有一个 注意：此字段可能返回 null，表示取不到有效值。 示例值：["xx"]
RoomId	String	如果是群消息，则不为空 注意：此字段可能返回 null，表示取不到有效值。 示例值：roomId
MsgTime	Integer	消息发送的时间戳，单位为秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1621496513
Video	ChatArchivingMsgTypeVideo	MsgType=video时的消息体，忽略此字段，见BodyJson字段 注意：此字段可能返回 null，表示取不到有效值。
BodyJson	String	根据MsgType的不同取值，解析内容不同，参考： https://open.work.weixin.qq.com/api/doc/90000/90135/91774 注意：此字段可能返回 null，表示取不到有效值。 示例值：{}

ChatArchivingMsgTypeVideo

会话存档的视频消息类型

被如下接口引用：QueryChatArchivingList。

名称	类型	描述
PlayLength	Integer	视频时长，单位秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：4
FileSize	Integer	文件大小 注意：此字段可能返回 null，表示取不到有效值。 示例值：645990
CosKey	String	视频资源对象Cos下载地址 示例值：http://xx

ClueInfoDetail

线索信息详情

被如下接口引用：QueryClueInfoList。

名称	类型	描述
ClueId	String	线索id，线索唯一识别编码 示例值：1348080105398452226
DealerId	String	接待客户经销商顾问所属经销商code 示例值：P1234
EnquireTime	Integer	线索获取时间，用户添加企业微信时间，单位是秒 示例值：1618556502
UnionId	String	客户在微信生态中唯一识别码 示例值：ozynqsulJFCZ2z1aYeS8h
Name	String	微信昵称 示例值：文
Phone	String	联系方式 示例值：134xxxx3232
SeriesCode	String	车系编号
ModelCode	String	车型编号
ProvinceCode	String	省份编号
CityCode	String	城市编号
SalesName	String	顾问名称 示例值：张三
SalesPhone	String	顾问电话 示例值：134xxxx4321
Remark	String	备注 示例值：备注
TagList	Array of String	标签 示例值：["标签1","标签2"]
UserName	String	客户姓名 注意：此字段可能返回 null，表示取不到有效值。 示例值：李四
LeadUserType	Integer	线索属性，0：个人，1：企业 示例值：0
LeadType	Integer	线索来源类型，1：线上，2：线下 示例值：1
ChannelId	Integer	线索渠道对应ID 示例值：1
ChannelName	String	线索渠道类型，与线索来源对应的渠道名称 示例值：渠道1
SourceChannelName	String	线索渠道名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：渠道1
Gender	Integer	0：未知，1：男，2：女 示例值：1

名称	类型	描述
CreateTime	String	线索创建时间戳，单位：秒 示例值：1618322621
UpdateTime	String	线索创建时间戳，单位：秒
LeadStatus	Integer	线索所处状态，101-待分配 201-待建档 301-已建档 401-已邀约 501-跟进中 601-已下订单 701-已成交 801-战败申请中 901-已战败 1001-未知状态 1101-转移申请中 1201-已完成 示例值：101
LevelCode	String	线索意向等级 示例值：101
ImportAtTime	Integer	线索成功导入的时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
DistributeTime	Integer	完成线索分配的时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
CreateAtTime	Integer	获取线索的时间戳，单位：秒 示例值：1618322621
WxId	String	客户微信id
BrandCode	String	意向车型对应品牌code
BuildTime	Integer	建档时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
OrderTime	Integer	下订时间，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
ArrivalTime	Integer	到店时间，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
DeliveryTime	Integer	交车时间，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
FollowTime	Integer	上次跟进时间，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
NextFollowTime	Integer	下次跟进时间，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
OrgId	Integer	线索所属组织id 示例值：1
OrgName	String	线索所属组织名称 示例值：org1
Introducer	String	介绍人姓名 示例值：introducer

名称	类型	描述
IntroducerPhone	String	介绍人电话 示例值：134xxxx4321
IsBindWx	Integer	是否关联微信 1 是 0 否 示例值：1
IsMerge	Integer	是否经过合并 1 是 0 否 示例值：1
IsInvalid	Integer	是否无效 1 是 0 否 示例值：0
InvalidType	String	无效类型 示例值：1
InvalidTypeName	String	无效类型枚举： 无意向购买、空错号、未接听、其他 示例值：空错号
InvalidRemark	String	由顾问手动输入的无效原因文字
InvalidTime	Integer	无效时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
DealerName	String	经销商名称 示例值：dealer1
ShopId	Integer	经销商下级门店ID 示例值：1234567890
ShopName	String	经销商下级门店名称 示例值：shop1
Position	String	职位 示例值：销售
CorpShopId	String	自定义的门店id 示例值：S1234

CorpUserInfo

企业成员信息

被如下接口引用：QueryUserInfoList。

名称	类型	描述
UserId	Integer	企业成员UserId 示例值：1323253932850728968
UserName	String	企业成员在SaaS名片内填写的姓名 注意：此字段可能返回 null，表示取不到有效值。 示例值：杨过
UserOpenId	String	企业成员在企微原生通讯录内的id 注意：此字段可能返回 null，表示取不到有效值。 示例值：YangGuo

名称	类型	描述
DealerId	Integer	成员所属经销商id，可为空 注意：此字段可能返回 null，表示取不到有效值。 示例值：1438427185617956866
ShopId	Integer	成员所属门店id，可为空 注意：此字段可能返回 null，表示取不到有效值。 示例值：1381564121059749890
Phone	String	企业成员手机号 注意：此字段可能返回 null，表示取不到有效值。 示例值：134****1234
OrgIds	String	成员所属部门id列表，仅返回该应用有查看权限的部门id；成员授权模式下，固定返回根部门id，即固定为1；多个部门使用逗号分割 注意：此字段可能返回 null，表示取不到有效值。 示例值：3,5
MainDepartment	String	主部门，仅当应用对主部门有查看权限时返回 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
IsLeaderInDept	String	是否为部门负责人，第三方应用可为空。与orgIds值一一对应，多个部门使用逗号隔开，0-否，1-是 注意：此字段可能返回 null，表示取不到有效值。 示例值：0,1
Status	Integer	激活状态: 0=已激活，1=已禁用，-1=退出企业" 示例值：0
JobNumber	String	工号 注意：此字段可能返回 null，表示取不到有效值。 示例值：12345678

CrmStatisticsData

CRM统计数据响应

被如下接口引用：QueryCrmStatistics。

名称	类型	描述
LeadCnt	Integer	新增线索 示例值：10
BuildCnt	Integer	新增建档 示例值：5
InvitedCnt	Integer	新增到店 示例值：1
OrderedCnt	Integer	新增下订 示例值：5
DeliveredCnt	Integer	新增成交 示例值：10
DefeatCnt	Integer	新增战败 示例值：3
NewContactCnt	Integer	新增好友 示例值：2

名称	类型	描述
StatisticalTime	String	统计时间, 单位: 天 示例值: 2022-11-11

CustomerActionEventDetail

外部联系人SaaS使用明细数据

被如下接口引用: QueryCustomerEventDetailStatistics。

名称	类型	描述
EventCode	String	事件码 示例值: 1000
EventType	Integer	事件类型 示例值: 1
EventSource	Integer	事件来源 示例值: 1
ExternalUserId	String	外部联系人id 示例值: wmQIM2CwAAL1ZKqK-icQG9XAMa3AhP-A
SalesId	Integer	销售顾问id 示例值: 1323253932850728968
MaterialType	Integer	素材类型 示例值: 1
MaterialId	Integer	素材编号id 示例值: 1348495383617208321
EventTime	Integer	事件上报时间, 单位: 秒 示例值: 1647423519

CustomerProfile

潜在客户档案信息

被如下接口引用: QueryCustomerProfileList。

名称	类型	描述
CustomerId	Integer	客户档案id, 客户唯一识别编码 示例值: 1234567890
DealerCode	String	所属经销商id
UnionId	String	客户在微信生态中唯一识别码 示例值: ozynqsulJFCZ2z1aYeS8h
CreateTime	String	档案创建时间戳, 单位: 秒 示例值: 2023-01-01 00:00:00
UserName	String	客户姓名 示例值: 张三
Gender	Integer	0未知, 1: 男, 2: 女 示例值: 1

名称	类型	描述
Phone	String	客户联系手机号 示例值: 123****4567
AgeRangeName	String	客户年龄段名称 示例值: 26-30
JobTypeName	String	客户行业类型名称信息, 如教师、医生
Address	String	客户居住地址 示例值: 地址1
LeadsProcessStatus	Integer	客户所处状态 0:已分配 1:未分配 1 待建档, 2 已建档, 3 已到店 4 已经试驾 5 战败申请中 6 已战败 7 已成交 示例值: 0
LeadType	Integer	客户来源类型, 1: 线上, 2: 线下 示例值: 1
SourceName	String	与客户来源类型对应的渠道名称 示例值: 自然到店
LeadsLevelCode	String	客户购车的意向等级 示例值: 101
VehicleBrandCode	String	客户意向品牌编号 示例值: 品牌编号
VehicleSeriesCode	String	客户意向车系编号 示例值: 车系编号
VehicleTypeCode	String	客户意向车型编号 示例值: 车型编号
VehiclePurpose	VehiclePurpose	购车用途信息
PurchaseConcern	Array of PurchaseConcern	购车关注点信息
SalesName	String	客户所属顾问姓名 示例值: 张三
SalesPhone	String	客户所属顾问手机号 示例值: 123****4567
RealArrivalTime	Integer	客户实际到店时间戳, 单位: 秒 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1618322621
CompleteTestDriveTime	String	客户到店完成试乘试驾时间戳, 单位: 秒 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2023-01-01 00:00:00
OrderTime	Integer	客户完成下订的时间戳, 单位: 秒 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1618322621
DeliveryTime	Integer	客户成交的时间戳, 单位: 秒 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1618322621

名称	类型	描述
InvoiceTime	Integer	开票时间戳 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
LoseTime	Integer	完成对此客户战败审批的时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
CreatedAtTime	Integer	线索成功获取的时间戳，单位：秒 示例值：1618322621
ImportAtTime	Integer	线索成功导入的时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
DistributeTime	Integer	完成线索分配的时间戳，单位：秒 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
LeadCreateTime	Integer	线索成功创建的时间戳，单位：秒 示例值：1618322621
Nickname	String	线索关联微信昵称 示例值：jack
OrgIdList	Array of String	线索归属部门节点 示例值：["1", "2"]
Introducer	String	客户的介绍人姓名 注意：此字段可能返回 null，表示取不到有效值。 示例值：李四
IntroducerPhone	String	客户的介绍人手机号码 注意：此字段可能返回 null，表示取不到有效值。 示例值：123****4567
FollowTime	Integer	最近一次完成跟进的时间戳，单位：秒 示例值：1618322621
NextFollowTime	Integer	最近一次计划跟进的时间戳，单位：秒 示例值：1618322621
EnterpriseTags	Array of EnterpriseTag	已为该客户添加的企业标签信息 注意：此字段可能返回 null，表示取不到有效值。
ChannelTags	Array of ChannelTag	已为该客户添加的渠道标签信息 注意：此字段可能返回 null，表示取不到有效值。
LeadId	Integer	关联线索id
WxId	String	客户微信id
Position	String	顾问职位
IsBindWx	Integer	是否关联微信 1 是 0 否
IsInvalid	Integer	是否无效
InvalidType	String	无效类型
InvalidTypeName	String	无效类型名称

名称	类型	描述
InvalidTime	Integer	无效时间 注意：此字段可能返回 null，表示取不到有效值。
InvalidRemark	String	由顾问手动输入的无效原因文字
IsLose	Integer	线索是否战败
LoseType	String	战败类型
LoseTypeName	String	战败类型名称
LoseRemark	String	战败申请原因

DealerInfo

经销商信息

被如下接口引用：QueryDealerInfoList。

名称	类型	描述
DealerId	Integer	企微SaaS平台经销商id 示例值：1438393844623261698
DealerName	String	经销商名称 示例值：深圳神车经销商
ProvinceCode	String	所属省份编号 注意：此字段可能返回 null，表示取不到有效值。 示例值：110100
CityCodeList	Array of String	所属城市编号列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[110100]
BrandIdList	Array of String	业务覆盖品牌id列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[1,2]

EnterpriseTag

客户企业标签

被如下接口引用：QueryCustomerProfileList。

名称	类型	描述
GroupName	String	该客户档案当前已成功关联的企业标签分组名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：标签分组名称
TagName	String	该客户档案当前已成功关联的企业标签名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：标签名称
TagId	String	该客户档案当前已成功关联的企业标签的id 注意：此字段可能返回 null，表示取不到有效值。 示例值：etAJ2GCAAAXtWyujaWJHDDGi0mACHAAA

ExternalContact

客户信息

被如下接口引用：QueryExternalContactDetail, QueryExternalContactDetailByDate。

名称	类型	描述
ExternalUserId	String	外部联系人的userId 示例值：wmQIM2CwAAE-JSOVGMzsRayl3ow5kGzQ
Gender	Integer	外部联系人性别 0-未知 1-男性 2-女性 示例值：1
Name	String	外部联系人的名称 示例值：文
Type	Integer	外部联系人的类型，1表示该外部联系人是微信用户，2表示该外部联系人是企业微信用户 示例值：1
UnionId	String	外部联系人在微信开放平台的唯一身份标识（微信unionid），通过此字段企业可将外部联系人与公众号/小程序用户关联起来。仅当联系人类型是微信用户，且企业或第三方服务商绑定了微信开发者ID有此字段。 注意：此字段可能返回 null，表示取不到有效值。 示例值：ozynqsulJFCZ2z1aYeS8h-nuasdAAA
Phone	String	外部联系人联系电话 注意：此字段可能返回 null，表示取不到有效值。 示例值：134****3232

ExternalContactDetailPro

具备更多信息的外部联系人详细信息

被如下接口引用：QueryExternalContactDetailByDate。

名称	类型	描述
Customer	ExternalContact	客户信息 注意：此字段可能返回 null，表示取不到有效值。
FollowUser	Array of FollowUserPro	添加了此外部联系人的企业成员信息 注意：此字段可能返回 null，表示取不到有效值。

ExternalContactSimpleInfo

外部联系人简短信息

被如下接口引用：QueryExternalContactList。

名称	类型	描述
ExternalUserId	String	外部联系人的userId 示例值：woAJ2GCAAAXtWyujaWJHDDGi0mACAAA
UserId	String	添加了此外部联系人的企业成员userId 示例值：yangGuo
SalesName	String	添加了此外部联系人的企业成员的姓名 注意：此字段可能返回 null，表示取不到有效值。 示例值：销售名称

名称	类型	描述
DepartmentIdList	Array of Integer	添加了此外部联系人的企业成员的归属部门id列表 示例值: [0,1]

ExternalContactTag

外部联系人标签

被如下接口引用: QueryExternalContactDetail, QueryExternalContactDetailByDate。

名称	类型	描述
GroupName	String	该成员添加此外部联系人所打标签的分组名称（标签功能需要企业微信升级到2.7.5及以上版本） 注意：此字段可能返回 null，表示取不到有效值。 示例值：标签分组名称
TagName	String	该成员添加此外部联系人所打标签名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：标签名称
Type	Integer	该成员添加此外部联系人所打标签类型, 1-企业设置, 2-用户自定义 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
TagId	String	该成员添加此外部联系人所打企业标签的id，仅企业设置（type为1）的标签返回 注意：此字段可能返回 null，表示取不到有效值。 示例值：etAJ2GCAAAXtWyujaWJHDDGi0mACHAAA

ExternalUserEventInfo

外部联系人事件信息

被如下接口引用: QueryExternalUserEventList。

名称	类型	描述
EventCode	String	事件编码, 添加外部联系人(ADD_EXTERNAL_CUSTOMER)/成员删除外部联系人(DELETE_EXTERNAL_CUSTOMER)/外部联系人删除成员(DELETE_FOLLOW_USER) 示例值: ADD_EXTERNAL_CUSTOMER
ExternalUserId	String	外部联系人id 示例值: woQIM2CwAA8BPLSVDDDBZUYIlviX3_9DQ
SalesId	String	企微SaaS的成员id 示例值: 1323253932850728968
EventTime	Integer	事件上报时间戳, 单位: 秒 示例值: 1614923098

ExternalUserMappingInfo

外部联系人映射信息

被如下接口引用: QueryExternalUserMappingInfo。

名称	类型	描述
----	----	----

名称	类型	描述
CorpExternalUserId	String	企业主体对应的外部联系人userId 示例值：wmQIM2CwAAywHqH07voRw9iuAAA09eQ
ExternalUserId	String	乐销车应用主体对应的外部联系人, 当不存在好友关系时, 该字段值为空 注意：此字段可能返回 null，表示取不到有效值。 示例值：wmQIM2CwAAywHqH07voRw9feafe9eQ

FollowInfo

发生过跟进的潜在客户信息

被如下接口引用：QueryFollowList。

名称	类型	描述
Clueld	Integer	线索id 示例值：1348080105398452226
CustomerId	Integer	客户档案id 示例值：1348080105398452226
UserName	String	客户姓名 示例值：张三
Phone	String	客户的手机号 示例值：134xxxx4321
IsOverdue	Integer	是否逾期 示例值：0
OverdueTime	Integer	逾期时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
EventType	Integer	发生事件 示例值：1
EventTypeName	String	发生事件名称 示例值：再次到店
FollowWayType	String	跟进方式 示例值：1
FollowWayName	String	跟进方式名称 示例值：线上跟进
FollowTime	Integer	本次跟进时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
NextFollowTime	Integer	下次跟进时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1618322621
FollowRecord	String	跟进记录

FollowUser

添加了此外部联系人的企业成员信息

被如下接口引用：QueryExternalContactDetail。

名称	类型	描述
UserId	String	添加了此外部联系人的企业成员userid 示例值：rocky
Remark	String	该成员对此外部联系人的备注 注意：此字段可能返回 null，表示取不到有效值。 示例值：备注
Description	String	该成员对此外部联系人的描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：描述
CreateTime	Integer	该成员添加此外部联系人的时间戳，单位为秒 示例值：1618322621
AddWay	Integer	该成员添加此客户的来源，具体含义详见 来源定义 示例值：1
OperUserId	String	发起添加的userid，如果成员主动添加，为成员的userid；如果是客户主动添加，则为客户的外部联系人userid；如果是内部成员共享/管理员分配，则为对应的成员/管理员userid 示例值：wmQIM2CwAAE-JSOVGMzsRayl3ow5kGzQ
Tags	Array of ExternalContactTag	该成员添加此外部联系人所打标签信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]

FollowUserPro

具备更多信息的添加了此外部联系人的企业成员信息

被如下接口引用：QueryExternalContactDetailByDate。

名称	类型	描述
UserId	String	添加了此外部联系人的企业成员userid 示例值：rocky
Remark	String	该成员对此外部联系人的备注 注意：此字段可能返回 null，表示取不到有效值。 示例值：备注
Description	String	该成员对此外部联系人的描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：描述
CreateTime	Integer	该成员添加此外部联系人的时间戳，单位为秒 示例值：1618322621
AddWay	Integer	该成员添加此客户的来源，具体含义详见 来源定义 示例值：0
OperUserId	String	发起添加的userid，如果成员主动添加，为成员的userid；如果是客户主动添加，则为客户的外部联系人userid；如果是内部成员共享/管理员分配，则为对应的成员/管理员userid 示例值：wmQIM2CwAAE-JSOVGMzsRayl3ow5kGzQ
Tags	Array of ExternalContactTag	该成员添加此外部联系人所打标签信息 注意：此字段可能返回 null，表示取不到有效值。

名称	类型	描述
SalesName	String	添加了此外部联系人的企业成员的姓名 注意：此字段可能返回 null，表示取不到有效值。 示例值：张三
DepartmentIdList	Array of Integer	企业成员的归属部门id列表 示例值：[1,2,3]

LicenseInfo

license相关信息

被如下接口引用：QueryLicenseInfo。

名称	类型	描述
License	String	license编号 示例值：A1c0tVAAAGSB1
LicenseEdition	Integer	license版本；1-基础版，2-标准版，3-增值版 示例值：1
ResourceStartTime	String	生效开始时间, 格式yyyy-MM-dd HH:mm:ss 示例值：2021-07-01 00:00:00
ResourceEndTime	String	生效结束时间, 格式yyyy-MM-dd HH:mm:ss 示例值：2022-07-01 00:00:00
IsolationDeadline	String	隔离截止时间, 格式yyyy-MM-dd HH:mm:ss 注意：此字段可能返回 null，表示取不到有效值。 示例值：2022-07-08 00:00:00
DestroyTime	String	资源计划销毁时间, 格式yyyy-MM-dd HH:mm:ss 注意：此字段可能返回 null，表示取不到有效值。 示例值：2022-07-15 00:00:00
Status	Integer	资源状态，1.正常，2.隔离，3.销毁 示例值：1
Extra	String	扩展信息 注意：此字段可能返回 null，表示取不到有效值。

LiveCodeDetail

活动活码详情

被如下接口引用：QueryActivityLiveCodeList。

名称	类型	描述
LiveCodeId	Integer	活码id 示例值：1381612531432296450
LiveCodeName	String	活码名称 示例值：留资活动活码
ShortChainAddress	String	短链url 注意：此字段可能返回 null，表示取不到有效值。 示例值：https://aggretrip.map.qq.com/wework_saas/dist/index.html?wexin_t

名称	类型	描述
LiveCodePreview	String	活码二维码 注意：此字段可能返回 null，表示取不到有效值。 示例值：https://aggretrip.map.qq.com/wework_saas/dist/index.html?wexin_t
ActivityId	Integer	活动id 示例值：1381619428653346818
ActivityName	String	活动名称 示例值：留资活动名称
LiveCodeState	Integer	活码状态，-1：删除，0：启用，1禁用，默认为0 示例值：0
LiveCodeData	String	活码参数，每个活码参数都是不一样的， 这个的值对应的是字符串json类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"channel": {"chan
CreateTime	Integer	创建时间戳，单位为秒 示例值：1621496513
UpdateTime	Integer	更新时间戳，单位为秒 示例值：1621496513

MaterialInfo

素材信息响应体

被如下接口引用：QueryMaterialList。

名称	类型	描述
MaterialId	Integer	素材id 示例值：1348481184631488514
MaterialName	String	素材名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：图片名称
Status	Integer	素材状态，-1: 删除 0: 启用 1: 禁用 示例值：0

MiniAppCodeInfo

小程序码信息

被如下接口引用：QueryMiniAppCodeList。

名称	类型	描述
Id	Integer	主键id 示例值：30
MiniAppName	String	小程序名称 示例值：小程序名称
MiniAppLogo	String	小程序logo 示例值：https://xxx.jpg

名称	类型	描述
MiniAdminUrl	String	小程序管理端地址 示例值: https://www.xxx.com
State	Integer	状态: 0正常, 1删除 示例值: 0
CreateTime	Integer	创建时间戳, 单位为秒 示例值: 1621496513
UpdateTime	Integer	更新时间戳, 单位为秒 示例值: 1621496513

PurchaseConcern

购车关注点

被如下接口引用: QueryCustomerProfileList。

名称	类型	描述
Code	String	购车关注点code
Description	String	购车关注点描述

SalesActionEventDetail

企业成员SaaS使用明细数据

被如下接口引用: QueryStaffEventDetailStatistics。

名称	类型	描述
EventCode	String	事件码 示例值: 1000
EventType	Integer	事件类型 示例值: 1
EventSource	Integer	事件来源 示例值: 1
SalesId	Integer	销售顾问id 示例值: 1323253932850728968
MaterialType	Integer	素材类型 示例值: 1
MaterialId	Integer	素材编号id 示例值: 1348495383617208321
EventTime	Integer	事件上报时间, 单位: 秒 示例值: 1647423519

TagDetailInfo

标签详细信息

被如下接口引用: CreateCorpTag。

名称	类型	描述
TagName	String	标签名称 示例值：标签名称
BizTagId	String	标签业务ID 示例值：1394843909254754300
TagId	String	企微标签ID 示例值：etQIM2CwAA6_HurtDtzSyK68jfiaX0UQ
Sort	Integer	标签排序的次序值，sort值大的排序靠前。有效的值范围是[0, 2^32) 示例值：0
CreateTime	Integer	标签创建时间,单位为秒 示例值：1621500202

TagGroup

标签组信息

被如下接口引用：CreateCorpTag。

名称	类型	描述
GroupId	String	企微标签组id 示例值：etQIM2CwAAbJ-lpFWKlyEHrWD1Z6qYkw
BizGroupId	String	标签组业务id 示例值：1348107208961921026
GroupName	String	企微标签组名称，不能超过15个字符 示例值：组名
Sort	Integer	标签组次序值。sort值大的排序靠前。有效的值范围是[0, 2^32) 示例值：1
CreateTime	Integer	标签组创建时间,单位为秒 示例值：1621500202
Tags	Array of TagDetailInfo	标签组内的标签列表, 上限为20

TagInfo

标签信息

被如下接口引用：CreateCorpTag。

名称	类型	必选	描述
TagName	String	是	标签名称, 最大长度限制15个字符 示例值：标签名称
Sort	Integer	否	标签组排序,值越大,排序越靠前 示例值：1

VehicleInfo

车型车系信息

被如下接口引用：QueryVehicleInfoList。

名称	类型	描述
BrandId	Integer	品牌id 示例值：1334702033519960066
BrandName	String	品牌名称 示例值：品牌名称
SeriesId	Integer	车系id 示例值：1334702033528348674
SeriesName	String	车系名称 示例值：车系名称
ModelId	Integer	车型id 示例值：1334740794876203009
ModelName	String	车型名称 示例值：车型名称

VehiclePurpose

购车用途详细信息

被如下接口引用：QueryCustomerProfileList。

名称	类型	描述
VehiclePurposeCode	String	购车用途code
VehiclePurposeName	String	购车用途名称

WeComTagDetail

企微个人标签信息,渠道活码使用

被如下接口引用：CreateChannelCode, QueryChannelCodeList。

名称	类型	必选	描述
GroupName	String	否	标签分组名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：测试111
BizGroupId	String	否	标签分组业务ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：1394843720913727500
TagName	String	否	标签名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：涛涛涛涛涛涛涛涛涛涛涛涛涛涛11
TagId	String	否	标签ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：etQIM2CwAA6_HurtDtzSyK68jfiaX0UQ
BizTagId	String	否	标签业务ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：1394843909254754300

名称	类型	必选	描述
Type	Integer	否	标签分类，1：企业设置、2：用户自定义 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
BizTagIdStr	String	否	标签业务ID字符串格式 注意：此字段可能返回 null，表示取不到有效值。 示例值：1394843909254754300

错误码

最近更新时间：2022-04-04 07:39:17

功能说明

如果返回结果中存在 Error 字段，则表示调用 API 接口失败。例如：

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

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

错误码列表

公共错误码

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

错误码	说明
InvalidRequest	请求 body 的 multipart 格式错误。
IpInBlacklist	IP地址在黑名单中。
IpNotInWhitelist	IP地址不在白名单中。
LimitExceeded	超过配额限制。
MissingParameter	缺少参数。
NoSuchProduct	产品不存在
NoSuchVersion	接口版本不存在。
RequestLimitExceeded	请求的次数超过了频率限制。
RequestLimitExceeded.GlobalRegionUinLimitExceeded	主账号超过频率限制。
RequestLimitExceeded.IPLimitExceeded	IP限频。
RequestLimitExceeded.UinLimitExceeded	主账号限频。
RequestSizeLimitExceeded	请求包超过限制大小。
ResourceInUse	资源被占用。
ResourceInsufficient	资源不足。
ResourceNotFound	资源不存在。
ResourceUnavailable	资源不可用。
ResponseSizeLimitExceeded	返回包超过限制大小。
ServiceUnavailable	当前服务暂时不可用。
UnauthorizedOperation	未授权操作。
UnknownParameter	未知参数错误，用户多传未定义的参数会导致错误。
UnsupportedOperation	操作不支持。
UnsupportedProtocol	http(s) 请求协议错误，只支持 GET 和 POST 请求。
UnsupportedRegion	接口不支持所传地域。

业务错误码

错误码	说明
AuthFailure	CAM签名/鉴权错误。
AuthFailure.InvalidAuthorizationCode	授权码无效。
AuthFailure.MissingAccessToken	令牌无效。
FailedOperation.OpenPlatformError	开放平台访问错误。
InvalidParameterValue.TimeSpanLimitExceeded	参数值时间跨度超出接口限制。
OperationDenied	操作被拒绝。