

微瓴同业开放平台

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內容。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

项目空间管理相关接口

根据标签获取行政区划列表

告警状态列表查询

修改工作空间园区属性

通过城市id查询工作空间列表

平台通用服务相关接口

查询应用列表

查询边缘应用凭证

查询接口列表

获取租户下的空间列表

获取文件下载URL

文件上传接口

创建应用授权令牌

BIM空间服务相关接口

查询建筑列表

查询建筑三维模型

查询建筑信息

查询分页构件信息

查询构件树

查询构件属性（详情）

查询场景列表

查询指定空间设备编号列表

查询指定空间下设备与构件绑定关系列表

查询设备绑定的空间信息

查询设备绑定的空间层级关系

查询空间分类

查询租户楼栋数量和楼栋建筑面积

查询项目空间楼栋数量与建筑面积

孪生门户相关接口

查询租户组织部门列表

查询租户人员列表

查询项目空间人员列表

视频服务相关接口

云台控制

云录像接口

实时流接口

历史流接口

断流接口

获取视频扩展信息

物联相关接口

拉取设备的设备类型列表

产品列表查询

模型列表查询

设备列表查询

- 批量新增设备
- 批量修改设备名字
- 获取设备影子数据
- 设备状态获取
- 设备状态统计
- 设备控制（单个、批量控制）
- 标签列表查询
- 告警类型获取
- 告警级别枚举获取
- 告警列表查询
- 添加告警处理记录
- 变更告警状态
- 批量消警
- 联动规则列表查询
- 联动规则详情查询
- 事件列表查询
- 动作列表查询
- 上报应用消息
- 批量上报应用消息

数据结构

错误码

API 文档

更新历史

最近更新时间：2024-01-31 01:29:02

第 7 次发布

发布时间：2024-01-31 01:28:56

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeDeviceList](#)
 - 新增入参：IsActive, IsCamera

修改数据结构：

- [DeviceDataInfo](#)
 - 新增成员：DeviceStatus, Status

第 6 次发布

发布时间：2024-01-17 01:27:53

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [ControlDevice](#)
 - 新增入参：IsSynchronized

第 5 次发布

发布时间：2023-12-01 01:31:31

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeAlarmStatusList](#)
 - 新增入参：ApplicationToken, Workspaceld
 - 新增出参：Result

新增数据结构：

- [AlarmStatusData](#)
- [DescribeAlarmStatusListRes](#)

第 4 次发布

发布时间：2023-11-15 02:45:47

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [AddAlarmProcessRecord](#)
 - 新增入参：RecordSet, Workspaceld, ApplicationToken, ApplicationId, ExtendOne

- 新增出参: Result
- [BatchCreateDevice](#)
 - 新增入参: WorkspaceId, AddDeviceSet, ApplicationToken
 - 新增出参: Result
- [BatchKillAlarm](#)
 - 新增入参: BeginTime, EndTime, StatusSet, WorkspaceId, UserId, UserName, ApplicationToken, ProcessorId, AlarmTypeSet, LevelSet, WIDSet, IdSet, Desc
 - 新增出参: Result
- [BatchReportAppMessage](#)
 - 新增入参: WorkspaceId, ApplicationToken, ReportSet
 - 新增出参: Result
- [ChangeAlarmStatus](#)
 - 新增入参: Id, Status, ProcessTime, ProcessType, WorkspaceId, UserId, UserName, ApplicationToken, Processor, ProcessDescription, ProcessExtend, ExtendOne, ApplicationId
 - 新增出参: Result
- [ControlCameraPTZ](#)
 - 新增入参: WID, CMD, WorkspaceId, ApplicationToken
 - 新增出参: Result
- [ControlDevice](#)
 - 新增入参: WorkspaceId, WIDSet, ControlData, ApplicationToken
 - 新增出参: Result
- [CreateApplicationToken](#)
 - 新增入参: ApplicationId, Nonce, TenantId, RequestTime, Signature
 - 新增出参: Result
- [DescribeActionList](#)
 - 新增入参: WorkspaceId, PageNumber, PageSize, ApplicationToken, ActionType, IdSet
 - 新增出参: Result
- [DescribeAdministrationByTag](#)
 - 新增入参: ApplicationToken, WorkspaceId, Tag
 - 新增出参: Result
- [DescribeAlarmLevelList](#)
 - 新增入参: WorkspaceId, ApplicationToken
 - 新增出参: Result
- [DescribeAlarmList](#)
 - 新增入参: BeginTime, EndTime, PageNumber, PageSize, WorkspaceId, ApplicationToken, Statuses, AlarmTypeSet, LevelSet, IdSet, AppIdSet, WIDSet, SpaceCodeSet, ExtendOne, ExtendTwo, ProcessorSet, GroupIdSet
 - 新增出参: Result
- [DescribeAlarmTypeList](#)
 - 新增入参: WorkspaceId, ApplicationToken, ParentType
 - 新增出参: Result
- [DescribeBuildingList](#)
 - 新增入参: WorkspaceId, ApplicationToken, HasModel, SpaceCodes
 - 新增出参: Result
- [DescribeBuildingModel](#)
 - 新增入参: BuildingId, WorkspaceId, ApplicationToken
 - 新增出参: Result
- [DescribeBuildingProfile](#)
 - 新增入参: BuildingId, WorkspaceId, ApplicationToken
 - 新增出参: Result
- [DescribeCameraExtendInfo](#)
 - 新增入参: WID, WorkspaceId, ApplicationToken
 - 新增出参: Result
- [DescribeCityWorkspaceList](#)
 - 新增入参: AdministrativeCodeSet, ApplicationToken
 - 新增出参: Result
- [DescribeDeviceList](#)

- 新增入参: WorkspaceId, PageNumber, PageSize, ApplicationToken, DeviceTypeSet, ProductIdSet, TagIdSet, SpaceCodeSet, DeviceTagSet, WIDSet, Field, GroupIdSet
- 新增出参: Result
- [DescribeDeviceShadowList](#)
 - 新增入参: TagIdSet
- [DescribeDeviceStatusList](#)
 - 新增入参: WorkspaceId, ApplicationToken, PageNumber, PageSize, DeviceTypeSet, ProductIdSet, TagIdSet, SpaceCodeSet, WIDSet, DeviceTagSet, DeviceStatusSet, StatusSet, IsAlive
 - 新增出参: Result
- [DescribeDeviceStatusStat](#)
 - 新增入参: Level, WorkspaceId, ApplicationToken, SpaceCodeSet, DeviceTypeSet
 - 新增出参: Result
- [DescribeDeviceTagList](#)
 - 新增入参: WorkspaceId, PageNumber, PageSize, ApplicationToken
 - 新增出参: Result
- [DescribeDeviceTypeList](#)
 - 新增入参: WorkspaceId, ApplicationToken, Flag
 - 新增出参: Result
- [DescribeEdgeApplicationToken](#)
 - 新增入参: ApplicationToken, Refresh
 - 新增出参: Result
- [DescribeElementProfilePage](#)
 - 新增入参: BuildingId, PageNumber, PageSize, WorkspaceId, ApplicationToken, ParentElementIds, Level, SpaceTypeCode, EntityTypes, IncludeDelete, StartTime, EndTime
 - 新增出参: Result
- [DescribeElementProfileTree](#)
 - 新增入参: BuildingId, WorkspaceId, ApplicationToken, ElementId, Level, SpaceTypeCode
 - 新增出参: Result
- [DescribeEventList](#)
 - 新增入参: WorkspaceId, PageNumber, PageSize, ApplicationToken, TriggerType, IdSet
 - 新增出参: Result
- [DescribeFileDownloadURL](#)
 - 新增入参: WorkspaceId, FileId, ApplicationToken
 - 新增出参: Result
- [DescribeFileUploadURL](#)
 - 新增入参: WorkspaceId, FileName, FileSize, ApplicationToken, FileMD5, SaveType, FileExpireTime, NoExpireFlag
 - 新增出参: Result
- [DescribeInterfaceList](#)
 - 新增入参: ApplicationToken, PageNumber, PageSize, Keyword, Style, Type
 - 新增出参: Result
- [DescribeLinkRuleList](#)
 - 新增入参: WorkspaceId, PageNumber, PageSize, ApplicationToken, TriggerType, IdSet
 - 新增出参: Result
- [DescribeModelList](#)
 - 新增入参: WorkspaceId, PageNumber, PageSize, ApplicationToken, DeviceTypeSet, ProductIdSet, ModelIdSet
 - 新增出参: Result
- [DescribeProductList](#)
 - 新增入参: WorkspaceId, PageNumber, PageSize, ApplicationToken, DeviceTypeSet, ProductIdSet, ModelIdSet
 - 新增出参: Result
- [DescribePropertyList](#)
 - 新增入参: BuildingId, ElementId, WorkspaceId, ApplicationToken
 - 新增出参: Result
- [DescribeRuleDetail](#)
 - 新增入参: WorkspaceId, Id, ApplicationToken
 - 新增出参: Result
- [DescribeSceneList](#)
 - 新增入参: WorkspaceId, ApplicationToken

- 新增出参: Result
- [DescribeSpaceDeviceIdList](#)
 - 新增入参: ElementIds, IsCascade, WorkspaceId, PageNumber, PageSize, ApplicationToken
 - 新增出参: Result
- [DescribeSpaceDeviceRelationList](#)
 - 新增入参: ElementIds, IsCascade, WorkspaceId, PageNumber, PageSize, ApplicationToken
 - 新增出参: Result
- [DescribeSpaceInfoByDeviceId](#)
 - 新增入参: DeviceId, WorkspaceId, ApplicationToken
 - 新增出参: Result
- [DescribeSpaceRelationByDeviceId](#)
 - 新增入参: DeviceId, WorkspaceId, ApplicationToken
 - 新增出参: Result
- [DescribeSpaceTypeList](#)
 - 新增入参: WorkspaceId, PageNumber, PageSize, ApplicationToken
 - 新增出参: Result
- [DescribeTenantBuildingCountAndArea](#)
 - 新增入参: WorkspaceIdList, ApplicationToken
 - 新增出参: Result
- [DescribeVideoCloudRecord](#)
 - 新增入参: WID, StartTime, EndTime, WorkspaceId, ApplicationToken
 - 新增出参: Result
- [DescribeVideoLiveStream](#)
 - 新增入参: WID, Protocol, WorkspaceId, ApplicationToken, StreamId, Env
 - 新增出参: Result
- [DescribeWorkSpaceBuildingCountAndArea](#)
 - 新增入参: WorkspaceIdList, ApplicationToken
 - 新增出参: Result
- [DescribeWorkspaceList](#)
 - 新增入参: ApplicationToken, WorkspaceId
 - 新增出参: Result
- [ModifyDeviceName](#)
 - 新增入参: WorkspaceId, Set, ApplicationToken
 - 新增出参: Result
- [ReportAppMessage](#)
 - 新增入参: WorkspaceId, Profile, ReportTs, Properties, ApplicationToken, EventSet, ServiceSet, ExtendTwo, Echo
 - 新增出参: Result
- [StopVideoStreaming](#)
 - 新增入参: Stream, WID, WorkspaceId, ApplicationToken
 - 新增出参: Result
- [UpdateWorkspaceParkAttributes](#)
 - 新增入参: WorkspaceId, ApplicationToken, ParkName, ParkNum
 - 新增出参: Result

新增数据结构:

- [Action](#)
- [ActionDetail](#)
- [ActionObj](#)
- [AddDeviceInfo](#)
- [AdministrationData](#)
- [AdministrativeDetail](#)
- [AlarmInfo](#)
- [AlarmLevelInfo](#)
- [AlarmTypeDetailInfo](#)
- [AlarmTypeInfo](#)
- [ApiContent](#)

- [ApilInfo](#)
- [ApilInfoList](#)
- [ApplicationTokenInfo](#)
- [BatchCreateDeviceRes](#)
- [BatchReportAppMessageRes](#)
- [BuildingListRes](#)
- [BuildingModel](#)
- [BuildingModelRes](#)
- [BuildingProfile](#)
- [BuildingProfileRes](#)
- [CameraExtendInfoRes](#)
- [ControlDeviceRes](#)
- [ControlDeviceSet](#)
- [CreateDeviceFailed](#)
- [CreateDeviceSucceeded](#)
- [CustomField](#)
- [CustomFieldInfo](#)
- [DescribeActionListRes](#)
- [DescribeAdministrationByTagRes](#)
- [DescribeAlarmLevelListRes](#)
- [DescribeAlarmListRes](#)
- [DescribeAlarmTypeListRes](#)
- [DescribeCityWorkspaceListRes](#)
- [DescribeDeviceListRes](#)
- [DescribeEventListRes](#)
- [DescribeLinkRuleListRes](#)
- [DeviceDataInfo](#)
- [DeviceLocation](#)
- [DeviceModifyInfo](#)
- [DeviceSpaceInfo](#)
- [DeviceSpaceInfoRes](#)
- [DeviceStatusInfo](#)
- [DeviceStatusRes](#)
- [DeviceStatusStatRes](#)
- [DeviceTagInfo](#)
- [DeviceTagRes](#)
- [DeviceType](#)
- [DeviceTypeOverview](#)
- [DeviceTypeSet](#)
- [ElementCoordinates](#)
- [ElementProfile](#)
- [ElementProfilePageRes](#)
- [ElementProfileTreeNode](#)
- [ElementProfileTreeRes](#)
- [ElementProperty](#)
- [ElementPropertyRes](#)
- [ElementTranslate](#)
- [EmptyRes](#)
- [Event](#)
- [EventDetail](#)
- [EventObj](#)
- [FileDownloadURL](#)
- [FileInfo](#)
- [FileUploadURL](#)
- [HandleRecordInfo](#)
- [HandlerPersonInfo](#)
- [LinkRule](#)

- [LinkRuleInfo](#)
- [MessageProfile](#)
- [ModelInfo](#)
- [ModelSet](#)
- [ProcessRecordInfo](#)
- [ProductInfo](#)
- [ProductSet](#)
- [RawInfo](#)
- [RecordInfo](#)
- [RelatedProduct](#)
- [ReportAppMessage](#)
- [ReportImg](#)
- [ReportMsgRes](#)
- [RuleDetailRes](#)
- [SceneInfo](#)
- [SceneListRes](#)
- [SpaceDataListStatsRes](#)
- [SpaceDataStats](#)
- [SpaceDataTotalStatsRes](#)
- [SpaceDeviceIdListRes](#)
- [SpaceDeviceRelation](#)
- [SpaceDeviceRelationRes](#)
- [SpaceInfo](#)
- [SpaceRelation](#)
- [SpaceRelationRes](#)
- [SpaceType](#)
- [SpaceTypeListRes](#)
- [StatDeviceType](#)
- [StatLevel](#)
- [VideoCloudRecordRes](#)
- [VideoRecordStreamRes](#)
- [WorkspaceInfo](#)
- [WorkspaceInfoList](#)

第 3 次发布

发布时间：2023-11-14 01:34:59

本次发布包含了以下内容：

改善已有的文档。

修改接口：

- [DescribeApplicationList](#)
 - 新增入参：WorkspaceId, ApplicationToken, ApplicationId, PageNumber, PageSize
 - 新增出参：Result
- [DescribeDeviceShadowList](#)
 - 新增入参：WorkspaceId, WIDSet, PageNumber, PageSize, ApplicationToken, DeviceTypeSet, ProductIdSet, SpaceCodeSet, DeviceTagSet
 - 新增出参：Result
- [DescribeVideoRecordStream](#)
 - 新增入参：WID, Protocol, StartTime, EndTime, PlayBackRate, WorkspaceId, ApplicationToken, Stream, Env

新增数据结构：

- [ApplicationInfo](#)
- [ApplicationList](#)
- [ApplicationLogo](#)
- [DeviceShadowInfo](#)

• [DeviceShadowRes](#)

第 2 次发布

发布时间：2023-11-10 01:30:47

本次发布包含了以下内容：

改善已有的文档。

新增接口：

- [AddAlarmProcessRecord](#)
- [BatchCreateDevice](#)
- [BatchKillAlarm](#)
- [BatchReportAppMessage](#)
- [ChangeAlarmStatus](#)
- [ControlCameraPTZ](#)
- [ControlDevice](#)
- [CreateApplicationToken](#)
- [DescribeActionList](#)
- [DescribeAdministrationByTag](#)
- [DescribeAlarmLevelList](#)
- [DescribeAlarmList](#)
- [DescribeAlarmStatusList](#)
- [DescribeAlarmTypeList](#)
- [DescribeBuildingList](#)
- [DescribeBuildingModel](#)
- [DescribeBuildingProfile](#)
- [DescribeCameraExtendInfo](#)
- [DescribeCityWorkspaceList](#)
- [DescribeDeviceList](#)
- [DescribeDeviceShadowList](#)
- [DescribeDeviceStatusList](#)
- [DescribeDeviceStatusStat](#)
- [DescribeDeviceTagList](#)
- [DescribeDeviceTypeList](#)
- [DescribeElementProfilePage](#)
- [DescribeElementProfileTree](#)
- [DescribeEventList](#)
- [DescribeFileDownloadURL](#)
- [DescribeFileUploadURL](#)
- [DescribeLinkRuleList](#)
- [DescribeModelList](#)
- [DescribeProductList](#)
- [DescribePropertyList](#)
- [DescribeRuleDetail](#)
- [DescribeSceneList](#)
- [DescribeSpaceDeviceIdList](#)
- [DescribeSpaceDeviceRelationList](#)
- [DescribeSpaceInfoByDeviceId](#)
- [DescribeSpaceRelationByDeviceId](#)
- [DescribeSpaceTypeList](#)
- [DescribeTenantBuildingCountAndArea](#)
- [DescribeTenantDepartmentList](#)
- [DescribeTenantUserList](#)
- [DescribeVideoCloudRecord](#)
- [DescribeVideoLiveStream](#)
- [DescribeVideoRecordStream](#)

- [DescribeWorkspaceBuildingCountAndArea](#)
- [ModifyDeviceName](#)
- [ReportAppMessage](#)
- [StopVideoStreaming](#)
- [UpdateWorkspaceParkAttributes](#)

新增数据结构:

- [SsoDepartment](#)
- [SsoDepartmentsResult](#)
- [SsoUser](#)
- [SsoUserResult](#)

第 1 次发布

发布时间: 2023-09-12 09:30:48

本次发布包含了以下内容:

改善已有的文档。

新增接口:

- [DescribeApplicationList](#)
- [DescribeEdgeApplicationToken](#)
- [DescribeInterfaceList](#)
- [DescribeWorkspaceList](#)
- [DescribeWorkspaceUserList](#)

新增数据结构:

- [SsoTeamUser](#)
- [SsoTeamUserResult](#)

简介

最近更新时间：2023-09-12 09:31:05

欢迎使用 微瓴同业开放平台 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](#)

微瓴同业开放平台（WeilingWith）承袭微瓴的技术能力，为园区/楼宇提供智慧空间数字底座及物联操作系统。响应行业数字化诉求，让建筑/园区如同电脑/手机般灵活加载和自由扩展软件应用。自带开箱即用的园区/楼宇运维管理基础应用，助力伙伴快速满足用户的基础诉求。

API 概览

最近更新時間：2023-11-27 00:30:34

項目空間管理相關接口

接口名稱	接口功能	頻率限制（次/秒）
DescribeAdministrationByTag	根據標籤獲取行政區劃列表	20
DescribeAlarmStatusList	告警狀態列表查詢	20
UpdateWorkspaceParkAttributes	修改工作空間園區屬性	20
DescribeCityWorkspaceList	通過城市id查詢工作空間列表	20

平台通用服務相關接口

接口名稱	接口功能	頻率限制（次/秒）
DescribeApplicationList	查詢應用列表	20
DescribeEdgeApplicationToken	查詢邊緣應用憑證	20
DescribeInterfaceList	查詢接口列表	20
DescribeWorkspaceList	獲取租戶下的空間列表	20
DescribeFileDownloadURL	獲取文件下載URL	20
DescribeFileUploadURL	文件上傳接口	20
CreateApplicationToken	創建應用授權令牌	20

BIM空間服務相關接口

接口名稱	接口功能	頻率限制（次/秒）
DescribeBuildingList	查詢建築列表	20
DescribeBuildingModel	查詢建築三維模型	20
DescribeBuildingProfile	查詢建築信息	20
DescribeElementProfilePage	查詢分頁構件信息	20
DescribeElementProfileTree	查詢構件樹	20
DescribePropertyList	查詢構件屬性（詳情）	20
DescribeSceneList	查詢場景列表	20
DescribeSpaceDeviceIdList	查詢指定空間設備編號列表	20
DescribeSpaceDeviceRelationList	查詢指定空間下設備與構件綁定關係列表	20
DescribeSpaceInfoByDeviceId	查詢設備綁定的空間信息	20
DescribeSpaceRelationByDeviceId	查詢設備綁定的空間層級關係	20
DescribeSpaceTypeList	查詢空間分類	20
DescribeTenantBuildingCountAndArea	查詢租戶樓棟數量和樓棟建築面積	20
DescribeWorkSpaceBuildingCountAndArea	查詢項目空間樓棟數量與建築面積	20

孿生門戶相關接口

接口名称	接口功能	频率限制（次/秒）
DescribeTenantDepartmentList	查询租户组织部门列表	20
DescribeTenantUserList	查询租户人员列表	20
DescribeWorkspaceUserList	查询项目空间人员列表	20

视频服务相关接口

接口名称	接口功能	频率限制（次/秒）
ControlCameraPTZ	云台控制	20
DescribeVideoCloudRecord	云录像接口	20
DescribeVideoLiveStream	实时流接口	20
DescribeVideoRecordStream	历史流接口	20
StopVideoStreaming	断流接口	20
DescribeCameraExtendInfo	获取视频扩展信息	20

物联相关接口

接口名称	接口功能	频率限制（次/秒）
DescribeDeviceTypeList	拉取设备的设备类型列表	20
DescribeProductList	产品列表查询	20
DescribeModelList	模型列表查询	20
DescribeDeviceList	设备列表查询	20
BatchCreateDevice	批量新增设备	20
ModifyDeviceName	批量修改设备名字	20
DescribeDeviceShadowList	获取设备影子数据	20
DescribeDeviceStatusList	设备状态获取	20
DescribeDeviceStatusStat	设备状态统计	20
ControlDevice	设备控制（单个、批量控制）	20
DescribeDeviceTagList	标签列表查询	20
DescribeAlarmTypeList	告警类型获取	20
DescribeAlarmLevelList	告警级别枚举获取	20
DescribeAlarmList	告警列表查询	20
AddAlarmProcessRecord	添加告警处理记录	20
ChangeAlarmStatus	变更告警状态	20
BatchKillAlarm	批量消警	20
DescribeLinkRuleList	联动规则列表查询	20
DescribeRuleDetail	联动规则详情查询	20
DescribeEventList	事件列表查询	20
DescribeActionList	动作列表查询	20
ReportAppMessage	上报应用消息	20

接口名称	接口功能	频率限制（次/秒）
BatchReportAppMessage	批量上报应用消息	20

调用方式

请求结构

最近更新时间：2023-09-12 09:31:05

.....

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 `weilingwith.tencentcloudapi.com`，也支持指定地域域名访问，例如广州地域的域名为 `weilingwith.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 `weilingwith.ap-guangzhou.tencentcloudapi.com` 是一致的。

注意：对时延敏感的业务，建议指定带地域的域名。

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

目前支持的域名列表为：

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

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 01:31:05

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 weilingwith；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输入公共参数 Version 的说明。例如云服务器的版本 2017-03-12。
SignatureMethod	String	否	签名方式, 目前支持 HmacSHA256 和 HmacSHA1。只有指定此参数为 HmacSHA256 时, 才使用 HmacSHA256 算法验证签名, 其他情况均使用 HmacSHA1 验证签名。
Token	String	否	即 安全凭证服务 所颁发的临时安全凭证中的 Token, 使用时需要将 SecretId 和 SecretKey 的值替换为临时安全凭证中的 TmpSecretId 和 TmpSecretKey。使用长期密钥时不能设置此 Token 字段。
Language	String	否	指定接口返回的语言, 仅部分接口支持此参数。取值: zh-CN, en-US。zh-CN 返回中文, en-US 返回英文。

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

HTTP GET 请求结构示例：

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

最近更新時間：2023-12-27 01:46:51

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

根据以上规则，示例中得到的规范请求串如下：

```
POST
/

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

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

2. 拼接待签名字符串

按如下格式拼接待签名字符串：

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

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

注意：

1. Date 必须从时间戳 X-TC-Timestamp 计算得到，且时区为 UTC+0。如果加入系统本地时区信息，例如东八区，将导致白天和晚上调用成功，但是凌晨调用必定失败。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。
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-t
    
```

```
ype: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);

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

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

    return (ss.str());
}

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

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

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

// ***** 步骤 3: 计算签名 *****
char k_key[64] = {0};
sprintf(k_key, "%s%s", "TC3", SECRET_KEY);
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。
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 01:35:30

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

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

2.1. 对参数排序

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

```
{
  'Action': 'DescribeInstances',
  'InstanceIds.0': 'ins-09dx96dg',
  'Limit': 20,
  'Nonce': 11886,
  'Offset': 0,
  'Region': 'ap-guangzhou',
  'SecretId': '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"，因此格式化后就为 Action=DescribeInstances。

注意：“参数值”为原始值而非 url 编码后的值。

然后将格式化后的各个参数用"&"拼接在一起，最终生成的请求字符串为：

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

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

示例的拼接结果为：

```
GETcvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceIds.0=ins-09dx96dg&Limit=20&Nonce=11886&Offset=0&Region=ap-guangzhou&SecretId=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&InstanceIds.0=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 编码前需要先以 UTF-8 进行编码。

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

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

4. 签名失败

根据实际情况，存在以下签名失败的错误码，请根据实际情况处理。

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

5. 签名演示

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

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

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

- [Signature Demo](#)

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

最终输出的 url 可能为：<https://cvm.tencentcloudapi.com/?Action=DescribeInstances&InstanceId.0=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("InstanceIds.0", "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',
        'InstanceIds.0': 'ins-09dx96dg',
        'Limit': 20,
        'Nonce': 11886,
        'Offset': 0,
        'Region': 'ap-guangzhou',
        'SecretId': secret_id,
        'Timestamp': 1465185768, # int(time.time())
        'Version': '2017-03-12'
    }
    s = get_string_to_sign("GET", endpoint, data)
    data["Signature"] = sign_str(secret_key, s, hashlib.sha1)
    print(data["Signature"])
    # 此处会实际调用, 成功后可能产生计费
    
```

```
# resp = requests.get("https://" + endpoint, params=data)
# print(resp.url)
```

Golang

```
package main

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

func main() {
    // 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 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",
        "InstanceIds.0": "ins-09dx96dg",
        "Limit": strconv.Itoa(20),
        "Offset": strconv.Itoa(0),
    }

    var buf bytes.Buffer
    buf.WriteString("GET")
    buf.WriteString("cvm.tencentcloudapi.com")
    buf.WriteString("/")
    buf.WriteString("?")

    // sort keys by ascii asc order
    keys := make([]string, 0, len(params))
    for k, _ := range params {
        keys = append(keys, k)
    }
    sort.Strings(keys)

    for i := range keys {
        k := keys[i]
        buf.WriteString(k)
        buf.WriteString("=")
        buf.WriteString(params[k])
        buf.WriteString("&")
    }
    buf.Truncate(buf.Len() - 1)
```

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

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

PHP

```
<?php
// 需要设置环境变量 TENCENTCLOUD_SECRET_ID, 值为示例的 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 01:31:50

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

最近更新時間：2023-09-12 09:31:08

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。

项目空间管理相关接口

根据标签获取行政区划列表

最近更新时间：2023-11-30 08:49:34

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

根据标签获取行政区划列表

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：DescribeAdministrationByTag。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationToken	是	String	应用token 示例值：Lawbk967Ng41d934u8vveoS
WorkspaceId	是	Integer	工作空间ID 示例值：1016
Tag	是	String	标签值 示例值：城市

3. 输出参数

参数名称	类型	描述
Result	DescribeAdministrationByTagRes	行政区划返回结构 示例值：{"Response": {"RequestId": "c6802352-38e4-4a36-80dd-d80e57ce92a7", "Result": [{"List": [{"AdministrationCode": "130100", "AdministrationName": "石家庄市"}, {"AdministrationCode": "130200", "AdministrationName": "唐山市"}]}}
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 获取行政区划

获取行政区划

输入示例

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

```
{
  "ApplicationToken": "RQ7wnN7IMDutqe0Jj6t4MTqbBAePbryn",
  "WorkspaceId": 1124,
  "Tag": "城市"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "c6802352-38e4-4a36-80dd-d80e57ce92a7",
    "Result": {
      "List": [
        {
          "AdministrationCode": "130100",
          "AdministrationName": "石家庄市"
        },
        {
          "AdministrationCode": "130200",
          "AdministrationName": "唐山市"
        },
        {
          "AdministrationCode": "130300",
          "AdministrationName": "秦皇岛市"
        },
        {
          "AdministrationCode": "130400",
          "AdministrationName": "邯郸市"
        },
        {
          "AdministrationCode": "130500",
          "AdministrationName": "邢台市"
        },
        {
          "AdministrationCode": "130600",
          "AdministrationName": "保定市"
        },
        {
          "AdministrationCode": "130700",
          "AdministrationName": "张家口市"
        },
        {
          "AdministrationCode": "130800",
          "AdministrationName": "承德市"
        },
        {
          "AdministrationCode": "130900",
          "AdministrationName": "沧州市"
        },
        {
          "AdministrationCode": "131000",
          "AdministrationName": "廊坊市"
        },
        {
          "AdministrationCode": "131100",
```

```
"AdministrationName": "衡水市"
},
{
  "AdministrationCode": "140100",
  "AdministrationName": "太原市"
},
{
  "AdministrationCode": "140200",
  "AdministrationName": "大同市"
},
{
  "AdministrationCode": "140300",
  "AdministrationName": "阳泉市"
},
{
  "AdministrationCode": "140400",
  "AdministrationName": "长治市"
},
{
  "AdministrationCode": "140500",
  "AdministrationName": "晋城市"
},
{
  "AdministrationCode": "140600",
  "AdministrationName": "朔州市"
},
{
  "AdministrationCode": "140700",
  "AdministrationName": "晋中市"
},
{
  "AdministrationCode": "140800",
  "AdministrationName": "运城市"
},
{
  "AdministrationCode": "140900",
  "AdministrationName": "忻州市"
},
{
  "AdministrationCode": "141000",
  "AdministrationName": "临汾市"
},
{
  "AdministrationCode": "141100",
  "AdministrationName": "吕梁市"
},
{
  "AdministrationCode": "150100",
  "AdministrationName": "呼和浩特市"
},
{
  "AdministrationCode": "150200",
  "AdministrationName": "包头市"
},
{
  "AdministrationCode": "150300",
  "AdministrationName": "乌海市"
```

```
},
{
  "AdministrationCode": "150400",
  "AdministrationName": "赤峰市"
},
{
  "AdministrationCode": "150500",
  "AdministrationName": "通辽市"
},
{
  "AdministrationCode": "150600",
  "AdministrationName": "鄂尔多斯市"
},
{
  "AdministrationCode": "150700",
  "AdministrationName": "呼伦贝尔市"
},
{
  "AdministrationCode": "150800",
  "AdministrationName": "巴彦淖尔市"
},
{
  "AdministrationCode": "150900",
  "AdministrationName": "乌兰察布市"
},
{
  "AdministrationCode": "152200",
  "AdministrationName": "兴安盟"
},
{
  "AdministrationCode": "152500",
  "AdministrationName": "锡林郭勒盟"
},
{
  "AdministrationCode": "152900",
  "AdministrationName": "阿拉善盟"
},
{
  "AdministrationCode": "210100",
  "AdministrationName": "沈阳市"
},
{
  "AdministrationCode": "210200",
  "AdministrationName": "大连市"
},
{
  "AdministrationCode": "210300",
  "AdministrationName": "鞍山市"
},
{
  "AdministrationCode": "210400",
  "AdministrationName": "抚顺市"
},
{
  "AdministrationCode": "210500",
  "AdministrationName": "本溪市"
},
}
```

```
{
  "AdministrationCode": "210600",
  "AdministrationName": "丹东市"
},
{
  "AdministrationCode": "210700",
  "AdministrationName": "锦州市"
},
{
  "AdministrationCode": "210800",
  "AdministrationName": "营口市"
},
{
  "AdministrationCode": "210900",
  "AdministrationName": "阜新市"
},
{
  "AdministrationCode": "211000",
  "AdministrationName": "辽阳市"
},
{
  "AdministrationCode": "211100",
  "AdministrationName": "盘锦市"
},
{
  "AdministrationCode": "211200",
  "AdministrationName": "铁岭市"
},
{
  "AdministrationCode": "211300",
  "AdministrationName": "朝阳市"
},
{
  "AdministrationCode": "211400",
  "AdministrationName": "葫芦岛市"
},
{
  "AdministrationCode": "220100",
  "AdministrationName": "长春市"
},
{
  "AdministrationCode": "220200",
  "AdministrationName": "吉林市"
},
{
  "AdministrationCode": "220300",
  "AdministrationName": "四平市"
},
{
  "AdministrationCode": "220400",
  "AdministrationName": "辽源市"
},
{
  "AdministrationCode": "220500",
  "AdministrationName": "通化市"
},
{
```

```
"AdministrationCode": "220600",
"AdministrationName": "白山市"
},
{
"AdministrationCode": "220700",
"AdministrationName": "松原市"
},
{
"AdministrationCode": "220800",
"AdministrationName": "白城市"
},
{
"AdministrationCode": "222400",
"AdministrationName": "延边朝鲜族自治州"
},
{
"AdministrationCode": "230100",
"AdministrationName": "哈尔滨市"
},
{
"AdministrationCode": "230200",
"AdministrationName": "齐齐哈尔市"
},
{
"AdministrationCode": "230300",
"AdministrationName": "鸡西市"
},
{
"AdministrationCode": "230400",
"AdministrationName": "鹤岗市"
},
{
"AdministrationCode": "230500",
"AdministrationName": "双鸭山市"
},
{
"AdministrationCode": "230600",
"AdministrationName": "大庆市"
},
{
"AdministrationCode": "230700",
"AdministrationName": "伊春市"
},
{
"AdministrationCode": "230800",
"AdministrationName": "佳木斯市"
},
{
"AdministrationCode": "230900",
"AdministrationName": "七台河市"
},
{
"AdministrationCode": "231000",
"AdministrationName": "牡丹江市"
},
{
"AdministrationCode": "231100",
```

```
"AdministrationName": "黑河市"
},
{
  "AdministrationCode": "231200",
  "AdministrationName": "绥化市"
},
{
  "AdministrationCode": "232700",
  "AdministrationName": "大兴安岭地区"
},
{
  "AdministrationCode": "320100",
  "AdministrationName": "南京市"
},
{
  "AdministrationCode": "320200",
  "AdministrationName": "无锡市"
},
{
  "AdministrationCode": "320300",
  "AdministrationName": "徐州市"
},
{
  "AdministrationCode": "320400",
  "AdministrationName": "常州市"
},
{
  "AdministrationCode": "320500",
  "AdministrationName": "苏州市"
},
{
  "AdministrationCode": "320600",
  "AdministrationName": "南通市"
},
{
  "AdministrationCode": "320700",
  "AdministrationName": "连云港市"
},
{
  "AdministrationCode": "320800",
  "AdministrationName": "淮安市"
},
{
  "AdministrationCode": "320900",
  "AdministrationName": "盐城市"
},
{
  "AdministrationCode": "321000",
  "AdministrationName": "扬州市"
},
{
  "AdministrationCode": "321100",
  "AdministrationName": "镇江市"
},
{
  "AdministrationCode": "321200",
  "AdministrationName": "泰州市"
```



```
},
{
  "AdministrationCode": "321300",
  "AdministrationName": "宿迁市"
},
{
  "AdministrationCode": "330100",
  "AdministrationName": "杭州市"
},
{
  "AdministrationCode": "330200",
  "AdministrationName": "宁波市"
},
{
  "AdministrationCode": "330300",
  "AdministrationName": "温州市"
},
{
  "AdministrationCode": "330400",
  "AdministrationName": "嘉兴市"
},
{
  "AdministrationCode": "330500",
  "AdministrationName": "湖州市"
},
{
  "AdministrationCode": "330600",
  "AdministrationName": "绍兴市"
},
{
  "AdministrationCode": "330700",
  "AdministrationName": "金华市"
},
{
  "AdministrationCode": "330800",
  "AdministrationName": "衢州市"
},
{
  "AdministrationCode": "330900",
  "AdministrationName": "舟山市"
},
{
  "AdministrationCode": "331000",
  "AdministrationName": "台州市"
},
{
  "AdministrationCode": "331100",
  "AdministrationName": "丽水市"
},
{
  "AdministrationCode": "340100",
  "AdministrationName": "合肥市"
},
{
  "AdministrationCode": "340200",
  "AdministrationName": "芜湖市"
},
}
```

```
{
  "AdministrationCode": "340300",
  "AdministrationName": "蚌埠市"
},
{
  "AdministrationCode": "340400",
  "AdministrationName": "淮南市"
},
{
  "AdministrationCode": "340500",
  "AdministrationName": "马鞍山市"
},
{
  "AdministrationCode": "340600",
  "AdministrationName": "淮北市"
},
{
  "AdministrationCode": "340700",
  "AdministrationName": "铜陵市"
},
{
  "AdministrationCode": "340800",
  "AdministrationName": "安庆市"
},
{
  "AdministrationCode": "341000",
  "AdministrationName": "黄山市"
},
{
  "AdministrationCode": "341100",
  "AdministrationName": "滁州市"
},
{
  "AdministrationCode": "341200",
  "AdministrationName": "阜阳市"
},
{
  "AdministrationCode": "341300",
  "AdministrationName": "宿州市"
},
{
  "AdministrationCode": "341500",
  "AdministrationName": "六安市"
},
{
  "AdministrationCode": "341600",
  "AdministrationName": "亳州市"
},
{
  "AdministrationCode": "341700",
  "AdministrationName": "池州市"
},
{
  "AdministrationCode": "341800",
  "AdministrationName": "宣城市"
},
{
```

```
"AdministrationCode": "350100",
"AdministrationName": "福州市"
},
{
"AdministrationCode": "350200",
"AdministrationName": "厦门市"
},
{
"AdministrationCode": "350300",
"AdministrationName": "莆田市"
},
{
"AdministrationCode": "350400",
"AdministrationName": "三明市"
},
{
"AdministrationCode": "350500",
"AdministrationName": "泉州市"
},
{
"AdministrationCode": "350600",
"AdministrationName": "漳州市"
},
{
"AdministrationCode": "350700",
"AdministrationName": "南平市"
},
{
"AdministrationCode": "350800",
"AdministrationName": "龙岩市"
},
{
"AdministrationCode": "350900",
"AdministrationName": "宁德市"
},
{
"AdministrationCode": "360100",
"AdministrationName": "南昌市"
},
{
"AdministrationCode": "360200",
"AdministrationName": "景德镇市"
},
{
"AdministrationCode": "360300",
"AdministrationName": "萍乡市"
},
{
"AdministrationCode": "360400",
"AdministrationName": "九江市"
},
{
"AdministrationCode": "360500",
"AdministrationName": "新余市"
},
{
"AdministrationCode": "360600",
```

```
"AdministrationName": "鹰潭市"
},
{
"AdministrationCode": "360700",
"AdministrationName": "赣州市"
},
{
"AdministrationCode": "360800",
"AdministrationName": "吉安市"
},
{
"AdministrationCode": "360900",
"AdministrationName": "宜春市"
},
{
"AdministrationCode": "361000",
"AdministrationName": "抚州市"
},
{
"AdministrationCode": "361100",
"AdministrationName": "上饶市"
},
{
"AdministrationCode": "370100",
"AdministrationName": "济南市"
},
{
"AdministrationCode": "370200",
"AdministrationName": "青岛市"
},
{
"AdministrationCode": "370300",
"AdministrationName": "淄博市"
},
{
"AdministrationCode": "370400",
"AdministrationName": "枣庄市"
},
{
"AdministrationCode": "370500",
"AdministrationName": "东营市"
},
{
"AdministrationCode": "370600",
"AdministrationName": "烟台市"
},
{
"AdministrationCode": "370700",
"AdministrationName": "潍坊市"
},
{
"AdministrationCode": "370800",
"AdministrationName": "济宁市"
},
{
"AdministrationCode": "370900",
"AdministrationName": "泰安市"
```

```
},
{
  "AdministrationCode": "371000",
  "AdministrationName": "威海市"
},
{
  "AdministrationCode": "371100",
  "AdministrationName": "日照市"
},
{
  "AdministrationCode": "371300",
  "AdministrationName": "临沂市"
},
{
  "AdministrationCode": "371400",
  "AdministrationName": "德州市"
},
{
  "AdministrationCode": "371500",
  "AdministrationName": "聊城市"
},
{
  "AdministrationCode": "371600",
  "AdministrationName": "滨州市"
},
{
  "AdministrationCode": "371700",
  "AdministrationName": "菏泽市"
},
{
  "AdministrationCode": "410100",
  "AdministrationName": "郑州市"
},
{
  "AdministrationCode": "410200",
  "AdministrationName": "开封市"
},
{
  "AdministrationCode": "410300",
  "AdministrationName": "洛阳市"
},
{
  "AdministrationCode": "410400",
  "AdministrationName": "平顶山市"
},
{
  "AdministrationCode": "410500",
  "AdministrationName": "安阳市"
},
{
  "AdministrationCode": "410600",
  "AdministrationName": "鹤壁市"
},
{
  "AdministrationCode": "410700",
  "AdministrationName": "新乡市"
},
}
```

```
{
  "AdministrationCode": "410800",
  "AdministrationName": "焦作市"
},
{
  "AdministrationCode": "410900",
  "AdministrationName": "濮阳市"
},
{
  "AdministrationCode": "411000",
  "AdministrationName": "许昌市"
},
{
  "AdministrationCode": "411100",
  "AdministrationName": "漯河市"
},
{
  "AdministrationCode": "411200",
  "AdministrationName": "三门峡市"
},
{
  "AdministrationCode": "411300",
  "AdministrationName": "南阳市"
},
{
  "AdministrationCode": "411400",
  "AdministrationName": "商丘市"
},
{
  "AdministrationCode": "411500",
  "AdministrationName": "信阳市"
},
{
  "AdministrationCode": "411600",
  "AdministrationName": "周口市"
},
{
  "AdministrationCode": "411700",
  "AdministrationName": "驻马店市"
},
{
  "AdministrationCode": "420100",
  "AdministrationName": "武汉市"
},
{
  "AdministrationCode": "420200",
  "AdministrationName": "黄石市"
},
{
  "AdministrationCode": "420300",
  "AdministrationName": "十堰市"
},
{
  "AdministrationCode": "420500",
  "AdministrationName": "宜昌市"
},
{
```

```
"AdministrationCode": "420600",
"AdministrationName": "襄阳市"
},
{
"AdministrationCode": "420700",
"AdministrationName": "鄂州市"
},
{
"AdministrationCode": "420800",
"AdministrationName": "荆门市"
},
{
"AdministrationCode": "420900",
"AdministrationName": "孝感市"
},
{
"AdministrationCode": "421000",
"AdministrationName": "荆州市"
},
{
"AdministrationCode": "421100",
"AdministrationName": "黄冈市"
},
{
"AdministrationCode": "421200",
"AdministrationName": "咸宁市"
},
{
"AdministrationCode": "421300",
"AdministrationName": "随州市"
},
{
"AdministrationCode": "422800",
"AdministrationName": "恩施土家族苗族自治州"
},
{
"AdministrationCode": "430100",
"AdministrationName": "长沙市"
},
{
"AdministrationCode": "430200",
"AdministrationName": "株洲市"
},
{
"AdministrationCode": "430300",
"AdministrationName": "湘潭市"
},
{
"AdministrationCode": "430400",
"AdministrationName": "衡阳市"
},
{
"AdministrationCode": "430500",
"AdministrationName": "邵阳市"
},
{
"AdministrationCode": "430600",
```

```
"AdministrationName": "岳阳市"
},
{
"AdministrationCode": "430700",
"AdministrationName": "常德市"
},
{
"AdministrationCode": "430800",
"AdministrationName": "张家界市"
},
{
"AdministrationCode": "430900",
"AdministrationName": "益阳市"
},
{
"AdministrationCode": "431000",
"AdministrationName": "郴州市"
},
{
"AdministrationCode": "431100",
"AdministrationName": "永州市"
},
{
"AdministrationCode": "431200",
"AdministrationName": "怀化市"
},
{
"AdministrationCode": "431300",
"AdministrationName": "娄底市"
},
{
"AdministrationCode": "433100",
"AdministrationName": "湘西土家族苗族自治州"
},
{
"AdministrationCode": "440100",
"AdministrationName": "广州市"
},
{
"AdministrationCode": "440200",
"AdministrationName": "韶关市"
},
{
"AdministrationCode": "440300",
"AdministrationName": "深圳市"
},
{
"AdministrationCode": "440400",
"AdministrationName": "珠海市"
},
{
"AdministrationCode": "440500",
"AdministrationName": "汕头市"
},
{
"AdministrationCode": "440600",
"AdministrationName": "佛山市"
```



```
},
{
  "AdministrationCode": "440700",
  "AdministrationName": "江门市"
},
{
  "AdministrationCode": "440800",
  "AdministrationName": "湛江市"
},
{
  "AdministrationCode": "440900",
  "AdministrationName": "茂名市"
},
{
  "AdministrationCode": "441200",
  "AdministrationName": "肇庆市"
},
{
  "AdministrationCode": "441300",
  "AdministrationName": "惠州市"
},
{
  "AdministrationCode": "441400",
  "AdministrationName": "梅州市"
},
{
  "AdministrationCode": "441500",
  "AdministrationName": "汕尾市"
},
{
  "AdministrationCode": "441600",
  "AdministrationName": "河源市"
},
{
  "AdministrationCode": "441700",
  "AdministrationName": "阳江市"
},
{
  "AdministrationCode": "441800",
  "AdministrationName": "清远市"
},
{
  "AdministrationCode": "441900",
  "AdministrationName": "东莞市"
},
{
  "AdministrationCode": "442000",
  "AdministrationName": "中山市"
},
{
  "AdministrationCode": "445100",
  "AdministrationName": "潮州市"
},
{
  "AdministrationCode": "445200",
  "AdministrationName": "揭阳市"
},
}
```

```
{
  "AdministrationCode": "445300",
  "AdministrationName": "云浮市"
},
{
  "AdministrationCode": "450100",
  "AdministrationName": "南宁市"
},
{
  "AdministrationCode": "450200",
  "AdministrationName": "柳州市"
},
{
  "AdministrationCode": "450300",
  "AdministrationName": "桂林市"
},
{
  "AdministrationCode": "450400",
  "AdministrationName": "梧州市"
},
{
  "AdministrationCode": "450500",
  "AdministrationName": "北海市"
},
{
  "AdministrationCode": "450600",
  "AdministrationName": "防城港市"
},
{
  "AdministrationCode": "450700",
  "AdministrationName": "钦州市"
},
{
  "AdministrationCode": "450800",
  "AdministrationName": "贵港市"
},
{
  "AdministrationCode": "450900",
  "AdministrationName": "玉林市"
},
{
  "AdministrationCode": "451000",
  "AdministrationName": "百色市"
},
{
  "AdministrationCode": "451100",
  "AdministrationName": "贺州市"
},
{
  "AdministrationCode": "451200",
  "AdministrationName": "河池市"
},
{
  "AdministrationCode": "451300",
  "AdministrationName": "来宾市"
},
{
```

```
"AdministrationCode": "451400",
"AdministrationName": "崇左市"
},
{
"AdministrationCode": "460100",
"AdministrationName": "海口市"
},
{
"AdministrationCode": "460200",
"AdministrationName": "三亚市"
},
{
"AdministrationCode": "460300",
"AdministrationName": "三沙市"
},
{
"AdministrationCode": "460400",
"AdministrationName": "儋州市"
},
{
"AdministrationCode": "510100",
"AdministrationName": "成都市"
},
{
"AdministrationCode": "510300",
"AdministrationName": "自贡市"
},
{
"AdministrationCode": "510400",
"AdministrationName": "攀枝花市"
},
{
"AdministrationCode": "510500",
"AdministrationName": "泸州市"
},
{
"AdministrationCode": "510600",
"AdministrationName": "德阳市"
},
{
"AdministrationCode": "510700",
"AdministrationName": "绵阳市"
},
{
"AdministrationCode": "510800",
"AdministrationName": "广元市"
},
{
"AdministrationCode": "510900",
"AdministrationName": "遂宁市"
},
{
"AdministrationCode": "511000",
"AdministrationName": "内江市"
},
{
"AdministrationCode": "511100",
```

```
"AdministrationName": "乐山市"  
},  
{  
  "AdministrationCode": "511300",  
  "AdministrationName": "南充市"  
},  
{  
  "AdministrationCode": "511400",  
  "AdministrationName": "眉山市"  
},  
{  
  "AdministrationCode": "511500",  
  "AdministrationName": "宜宾市"  
},  
{  
  "AdministrationCode": "511600",  
  "AdministrationName": "广安市"  
},  
{  
  "AdministrationCode": "511700",  
  "AdministrationName": "达州市"  
},  
{  
  "AdministrationCode": "511800",  
  "AdministrationName": "雅安市"  
},  
{  
  "AdministrationCode": "511900",  
  "AdministrationName": "巴中市"  
},  
{  
  "AdministrationCode": "512000",  
  "AdministrationName": "资阳市"  
},  
{  
  "AdministrationCode": "513200",  
  "AdministrationName": "阿坝藏族羌族自治州"  
},  
{  
  "AdministrationCode": "513300",  
  "AdministrationName": "甘孜藏族自治州"  
},  
{  
  "AdministrationCode": "513400",  
  "AdministrationName": "凉山彝族自治州"  
},  
{  
  "AdministrationCode": "520100",  
  "AdministrationName": "贵阳市"  
},  
{  
  "AdministrationCode": "520200",  
  "AdministrationName": "六盘水市"  
},  
{  
  "AdministrationCode": "520300",  
  "AdministrationName": "遵义市"
```

```
},
{
  "AdministrationCode": "520400",
  "AdministrationName": "安顺市"
},
{
  "AdministrationCode": "520500",
  "AdministrationName": "毕节市"
},
{
  "AdministrationCode": "520600",
  "AdministrationName": "铜仁市"
},
{
  "AdministrationCode": "522300",
  "AdministrationName": "黔东南苗族侗族自治州"
},
{
  "AdministrationCode": "522600",
  "AdministrationName": "黔南布依族苗族自治州"
},
{
  "AdministrationCode": "522700",
  "AdministrationName": "黔西南布依族苗族自治州"
},
{
  "AdministrationCode": "530100",
  "AdministrationName": "昆明市"
},
{
  "AdministrationCode": "530300",
  "AdministrationName": "曲靖市"
},
{
  "AdministrationCode": "530400",
  "AdministrationName": "玉溪市"
},
{
  "AdministrationCode": "530500",
  "AdministrationName": "保山市"
},
{
  "AdministrationCode": "530600",
  "AdministrationName": "昭通市"
},
{
  "AdministrationCode": "530700",
  "AdministrationName": "丽江市"
},
{
  "AdministrationCode": "530800",
  "AdministrationName": "普洱市"
},
{
  "AdministrationCode": "530900",
  "AdministrationName": "临沧市"
},
},
```

```
{
  "AdministrationCode": "532300",
  "AdministrationName": "楚雄彝族自治州"
},
{
  "AdministrationCode": "532500",
  "AdministrationName": "红河哈尼族彝族自治州"
},
{
  "AdministrationCode": "532600",
  "AdministrationName": "文山壮族苗族自治州"
},
{
  "AdministrationCode": "532800",
  "AdministrationName": "西双版纳傣族自治州"
},
{
  "AdministrationCode": "532900",
  "AdministrationName": "大理白族自治州"
},
{
  "AdministrationCode": "533100",
  "AdministrationName": "德宏傣族景颇族自治州"
},
{
  "AdministrationCode": "533300",
  "AdministrationName": "怒江傈僳族自治州"
},
{
  "AdministrationCode": "533400",
  "AdministrationName": "迪庆藏族自治州"
},
{
  "AdministrationCode": "540100",
  "AdministrationName": "拉萨市"
},
{
  "AdministrationCode": "540200",
  "AdministrationName": "日喀则市"
},
{
  "AdministrationCode": "540300",
  "AdministrationName": "昌都市"
},
{
  "AdministrationCode": "540400",
  "AdministrationName": "林芝市"
},
{
  "AdministrationCode": "540500",
  "AdministrationName": "山南市"
},
{
  "AdministrationCode": "540600",
  "AdministrationName": "那曲市"
},
{
```

```
"AdministrationCode": "542500",
"AdministrationName": "阿里地区"
},
{
"AdministrationCode": "610100",
"AdministrationName": "西安市"
},
{
"AdministrationCode": "610200",
"AdministrationName": "铜川市"
},
{
"AdministrationCode": "610300",
"AdministrationName": "宝鸡市"
},
{
"AdministrationCode": "610400",
"AdministrationName": "咸阳市"
},
{
"AdministrationCode": "610500",
"AdministrationName": "渭南市"
},
{
"AdministrationCode": "610600",
"AdministrationName": "延安市"
},
{
"AdministrationCode": "610700",
"AdministrationName": "汉中市"
},
{
"AdministrationCode": "610800",
"AdministrationName": "榆林市"
},
{
"AdministrationCode": "610900",
"AdministrationName": "安康市"
},
{
"AdministrationCode": "611000",
"AdministrationName": "商洛市"
},
{
"AdministrationCode": "620100",
"AdministrationName": "兰州市"
},
{
"AdministrationCode": "620200",
"AdministrationName": "嘉峪关市"
},
{
"AdministrationCode": "620300",
"AdministrationName": "金昌市"
},
{
"AdministrationCode": "620400",
```

```
"AdministrationName": "白银市"
},
{
"AdministrationCode": "620500",
"AdministrationName": "天水市"
},
{
"AdministrationCode": "620600",
"AdministrationName": "武威市"
},
{
"AdministrationCode": "620700",
"AdministrationName": "张掖市"
},
{
"AdministrationCode": "620800",
"AdministrationName": "平凉市"
},
{
"AdministrationCode": "620900",
"AdministrationName": "酒泉市"
},
{
"AdministrationCode": "621000",
"AdministrationName": "庆阳市"
},
{
"AdministrationCode": "621100",
"AdministrationName": "定西市"
},
{
"AdministrationCode": "621200",
"AdministrationName": "陇南市"
},
{
"AdministrationCode": "622900",
"AdministrationName": "临夏回族自治州"
},
{
"AdministrationCode": "623000",
"AdministrationName": "甘南藏族自治州"
},
{
"AdministrationCode": "630100",
"AdministrationName": "西宁市"
},
{
"AdministrationCode": "630200",
"AdministrationName": "海东市"
},
{
"AdministrationCode": "632200",
"AdministrationName": "海北藏族自治州"
},
{
"AdministrationCode": "632300",
"AdministrationName": "黄南藏族自治州"
```



```
},
{
  "AdministrationCode": "632500",
  "AdministrationName": "海南藏族自治州"
},
{
  "AdministrationCode": "632600",
  "AdministrationName": "果洛藏族自治州"
},
{
  "AdministrationCode": "632700",
  "AdministrationName": "玉树藏族自治州"
},
{
  "AdministrationCode": "632800",
  "AdministrationName": "海西蒙古族藏族自治州"
},
{
  "AdministrationCode": "640100",
  "AdministrationName": "银川市"
},
{
  "AdministrationCode": "640200",
  "AdministrationName": "石嘴山市"
},
{
  "AdministrationCode": "640300",
  "AdministrationName": "吴忠市"
},
{
  "AdministrationCode": "640400",
  "AdministrationName": "固原市"
},
{
  "AdministrationCode": "640500",
  "AdministrationName": "中卫市"
},
{
  "AdministrationCode": "650100",
  "AdministrationName": "乌鲁木齐市"
},
{
  "AdministrationCode": "650200",
  "AdministrationName": "克拉玛依市"
},
{
  "AdministrationCode": "650400",
  "AdministrationName": "吐鲁番市"
},
{
  "AdministrationCode": "650500",
  "AdministrationName": "哈密市"
},
{
  "AdministrationCode": "652300",
  "AdministrationName": "昌吉回族自治州"
},
},
```

```
{
  "AdministrationCode": "652700",
  "AdministrationName": "博尔塔拉蒙古自治州"
},
{
  "AdministrationCode": "652800",
  "AdministrationName": "巴音郭楞蒙古自治州"
},
{
  "AdministrationCode": "652900",
  "AdministrationName": "阿克苏地区"
},
{
  "AdministrationCode": "653000",
  "AdministrationName": "克孜勒苏柯尔克孜自治州"
},
{
  "AdministrationCode": "653100",
  "AdministrationName": "喀什地区"
},
{
  "AdministrationCode": "653200",
  "AdministrationName": "和田地区"
},
{
  "AdministrationCode": "654000",
  "AdministrationName": "伊犁哈萨克自治州"
},
{
  "AdministrationCode": "654200",
  "AdministrationName": "塔城地区"
},
{
  "AdministrationCode": "654300",
  "AdministrationName": "阿勒泰地区"
}
]
}
```

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
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound	资源不存在。

告警状态列表查询

最近更新时间：2023-12-01 01:31:41

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

用来查询系统中的告警状态列表

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：DescribeAlarmStatusList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationToken	是	String	应用token 示例值：Lawbk967Ng41d934u8vveoS
WorkspaceId	是	String	工作空间ID 示例值：1016

3. 输出参数

参数名称	类型	描述
Result	DescribeAlarmStatusListRes	告警状态返回结构 示例值：{"List": [{"StatusID": "test", "StatusName": "测试", "StatusType": ""}]}
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 获取告警状态示例

获取告警状态

输入示例

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

{
  "ApplicationToken": "RQ7wnN7IMDutqe0Jj6t4MTqbBAePbryn",
  "WorkspaceId": "1124"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "257e0b36-70c5-4f51-ac81-83fca0ae7522",
    "Result": {
      "List": [
        {
          "StatusID": "test",
          "StatusName": "测试",
          "StatusType": ""
        }
      ]
    }
  }
}
```

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.ApiAuthorizationNotFound	未找到此应用该api授权信息
AuthFailure.TokenNotFound	未找到该token信息
InternalServerError.ApiGatewayInternalError	API网关内部错误
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound	资源不存在。

修改工作空间园区属性

最近更新時間：2023-12-12 02:59:18

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

修改工作空間園區屬性

默認接口請求頻率限制：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

以下請求參數列表僅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：UpdateWorkspaceParkAttributes。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
ApplicationToken	是	String	应用token 示例值：vfy4zPtG74k6OYKr9rNa9
ParkName	否	String	园区简称 示例值：test
ParkNum	否	String	园区编号 示例值：1023

3. 输出参数

参数名称	类型	描述
Result	EmptyRes	修改工作空间园区属性结果 示例值：{"Msg":"ok"}
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 返回成功示例

更改园区行政编码

输入示例

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

```
{
```

```
"WorkspaceId": 1016,
"ApplicationToken": "xxx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "2660b624-b925-485d-a8f4-07d4ed58d9a5",
    "Result": {
      "Msg": "ok"
    }
  }
}
```

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
InternalServerError.UpdateParkInfoFailed	更新工作空间下园区信息错误
InvalidParameter.DuplicateParkCode	园区编码重复
InvalidParameter.ExceedParkLengthLimit	园区名字或园区编码长度超过50字符
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id
ResourceNotFound	资源不存在。

通过城市id查询工作空间列表

最近更新时间：2023-11-30 08:49:32

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

通过城市id查询工作空间列表

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：DescribeCityWorkspaceList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
AdministrativeCodeSet.N	是	Array of String	行政区编码集合 示例值：130500
ApplicationToken	是	String	应用token 示例值：xxxx

3. 输出参数

参数名称	类型	描述
Result	DescribeCityWorkspaceListRes	工作空间信息集合 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"WorkspaceSet":[{"AdministrativeDetailSet":[{"AdministrativeCode":"130000","AdministrativeName":"河北省","AdministrativeTypeCode":"QH01"}, {"AdministrativeCode":"130500","AdministrativeName":"邢台市","AdministrativeTypeCode":"QH02"}, {"AdministrativeCode":"130521","AdministrativeName":"邢台县","AdministrativeTypeCode":"QH03"}],"ChineseName":"华南园区项目","Description":"华南园区项目","ParkName":"","ParkNum":"","Status":0,"WorkspaceId":1016}]}
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 通过行政区id获取项目空间信息

成功返回示例

输入示例

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



```
{
  "AdministrativeCodeSet": [
    "002"
  ],
  "ApplicationToken": "YzenL5LdGoxQM5gqJfCCoMDeGqUSsY78"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "ec9a85c9-b887-4481-a318-1c41f7d46733",
    "Result": {
      "WorkspaceSet": [
        {
          "WorkspaceId": 1016,
          "ChineseName": "华南园区项目",
          "Description": "华南园区项目",
          "Status": 0,
          "ParkName": "园区简称",
          "ParkNum": "001",
          "AdministrativeDetailSet": [
            {
              "AdministrativeTypeCode": "QH01",
              "AdministrativeCode": "440000",
              "AdministrativeName": "广东省"
            }
          ]
        }
      ]
    }
  }
}
```

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
InvalidParameter	参数错误。
InvalidParameterValue.EmptyAdministrativeCode	行政区编码长度为0
ResourceNotFound	资源不存在。

平台通用服务相关接口

查询应用列表

最近更新時間：2023-11-30 08:49:59

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

查詢指定空間關聯的應用列表

默認接口請求頻率限制：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

以下請求參數列表僅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeApplicationList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	项目空间id，本次查询返回的应用均关联至该空间 示例值：1001
ApplicationToken	是	String	应用token 示例值：XaaJvayJyusfUwuoJ2OIT5Z2YAVb6bVo
ApplicationId.N	否	Array of Integer	应用id数组，可选，填了则表示根据id批量查询 示例值：[10000,10001,10002]
PageNumber	否	Integer	请求页号 示例值：1
PageSize	否	Integer	页容量，默认为10 示例值：10

3. 输出参数

参数名称	类型	描述
Result	ApplicationList	应用列表 示例值：{ "ApplicationInfoList": [{ "Address": "http://checkproject-a6f6718d.dtwins.tencent.com/testcodingpush", "ApplicationId": "10041", "ApplicationLogo": { "FileId": "359cc1b5-88c1-4dd6-8ff4-21d0b36ac7e2", "Url": "https://wetwin-manager-1258344699.cos.ap-guangzhou.myqcloud.com/%2F100055/0/20230717/359cc1b5-88c1-4dd6-8ff4-21d0b36ac7e2/%E8%B5%84%E4%BA%A7%E7%AE%A1%E7%90%86-%E9%BB%91.png?q-sign-algorithm=sha1&q-ak=AKIDiFWWFo67rLZXFgrlNinb8NL2SaR2M0OU&q-sign-time=1693452005%3B1693453205&q-key-time=1693452005%3B1693453205&q-header-list=host&q-url-param-list=&q-signature=6bf531d1b08d6f75eb7e153c24746a61d7be37d0" }, "Description": "1234561111", "EnglishName": "zctest1111", "Name": "智能资产管理", "Type": 0 }, { "Address": "http://tws.twins.tencent.com/proxy/29098b25-447d-437e-987a-d493d55e84f5/weiling-pubsub/test", "ApplicationId": "10048", "ApplicationLogo": { "FileId": "", "Url": "" }, "Description": "", "EnglishName": "MsgCenter", "Name": "消息服务", "Type": 0 }, { "Address": "http://t.twins.tencent.com/", "ApplicationId": "10051", "ApplicationLogo": { "FileId": "", "Url": "" }, "Description": "", "EnglishName": "IOT", "Name": "孪生中台(物联感知中心)", "Type": 0 }], "TotalCount": "171" }
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 应用列表

获取应用列表

输入示例

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

{
  "WorkspaceId": 1016,
  "PageNumber": 1,
  "PageSize": 3,
  "ApplicationToken": "XggjvayjyusfUwuoj2OIT4Z2YAVq6bVo"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "2ebee73c-8477-48c0-ad4d-8a748b7842a3",
    "Result": {
      "ApplicationInfoList": [
        {
          "Address": "http://checkproject-a6f6718d.dtwins.tencent.com/testcodingpush",
          "ApplicationId": "10041",
          "ApplicationLogo": {
            "FileId": "359cc1b5-88c1-4dd6-8ff4-21d0b36ac7e2",
            "Url": "https://wetwin-manager-1258344699.cos.ap-guangzhou.myqcloud.com/%2F100055/0/20230717/359cc1b5-88c1-4dd6-8ff4-21d0b36ac7e2/%E8%B5%84%E4%BA%A7%E7%AE%A1%E7%90%86-%E9%BB%91.png?q-sign-algorithm=sha1&q-ak=AKIDiFWWFo67rLZXFgrINinb8NL2SaR2M00U&q-sign-time=1693452005%3B1693453205&q-key-time=1693452005%3B1693453205&q-header-list=host&q-url-param-list=&q-signature=6bf531d1b08d6f75eb7e153c24746a61d7be37d0"
          },
          "Description": "12345611111",
          "EnglishName": "zctest1111",
          "Name": "智能资产管理",
          "Type": 0
        },
        {
          "Address": "http://t-ws.twins.tencent.com/proxy/29098b25-447d-437e-987a-d493d55e84f5/weiling-pubsub/test",
          "ApplicationId": "10048",
          "ApplicationLogo": {
            "FileId": "",
            "Url": ""
          },
          "Description": "",
          "EnglishName": "MsgCenter",
          "Name": "消息服务",
          "Type": 0
        },
        {
          "Address": "http://t.twins.tencent.com/",
          "ApplicationId": "10051",
```

```

"ApplicationLogo": {
  "FileId": "",
  "Url": ""
},
"Description": "",
"EnglishName": "IOT",
"Name": "孪生中台(物联感知中心)",
"Type": 0
}
],
"TotalCount": "171"
}
}
}
    
```

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.ApiAuthorizationNotFound	未找到此应用该api授权信息
AuthFailure.TokenNotFound	未找到该token信息
InternalServerError	内部错误。
InternalServerError.ApiGatewayInternalServerError	API网关内部错误
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidApplicationId	应用id非法

查询边缘应用凭证

最近更新時間：2023-11-30 08:49:58

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

查詢邊緣應用憑證

默認接口請求頻率限制：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

以下請求參數列表僅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

參數名稱	必選	類型	描述
Action	是	String	公共參數 ，本接口取值：DescribeEdgeApplicationToken。
Version	是	String	公共參數 ，本接口取值：2023-04-27。
Region	是	String	公共參數 ，詳見產品支持的 地域列表 。
ApplicationToken	是	String	應用token 示例值：XaaJvayJyusfUwuoJ2OIT5Z2YAVb6bVo
Refresh	否	Boolean	是否刷新token，默認為false 示例值：true

3. 输出参数

參數名稱	類型	描述
Result	ApplicationTokenInfo	邊緣應用令牌信息 示例值：Oq9lShbmhCT3FLnemDCecrXAnAHQe6SJ
RequestId	String	唯一請求 ID，每次請求都會返回。定位問題時需要提供該次請求的 RequestId。

4. 示例

示例1 获取边缘应用凭证

获取边缘应用凭证

输入示例

```
POST / HTTP/1.1
Host: weilingwith.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeEdgeApplicationToken
<公共請求參數>

{
  "Refresh": true,
  "ApplicationToken": "XggJvayJyusfUwuoJ2OIT4Z2YAVq6bVo"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "ff7892ed-9bb5-4bb6-a5d3-23ecb1e50369",
    "Result": {
      "Token": "UH6MurnI2fQpdAkBYT0aaMj2SuTgmclt"
    }
  }
}
```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

查询接口列表

最近更新时间：2023-11-30 08:49:56

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询接口列表

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：DescribeInterfaceList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationToken	是	String	应用token 示例值：XaaJvayJyusfUwuoJ2OIT5Z2YAVb6bVo
PageNumber	否	Integer	请求页号 示例值：1
PageSize	否	Integer	请求页容量，默认全量返回 示例值：10
Keyword	否	String	查询关键字 示例值：视频
Style.N	否	Array of Integer	接口方式 1.http 2消息通知服务 示例值：[1,2]
Type.N	否	Array of Integer	接口分类0. 其他服务 1. IOT服务 2. 空间服务 3.微应用服务 4.场景服务 5.AI算法服务 6.任务算法服务 7.第三方服务 8.3DTiles服务 示例值：[1,2]

3. 输出参数

参数名称	类型	描述
------	----	----


```

{
  "Response": {
    "RequestId": "1cf5725d-d238-40c7-9208-dd073428ccf9",
    "Result": {
      "ApiInfo": [
        {
          "Address": "/TBIM-Server/services/3D-51fab803-8651-4520-902a-53fab2db61fa/v20230814142004569/Tiles/tileset.json",
          "ApiId": "0086a573-570c-4cdb-b945-e22df31eb589",
          "AppId": "10053",
          "ApplyAudit": 0,
          "Body": "",
          "DataAudit": 0,
          "Description": "333",
          "IsCommonSpace": false,
          "Method": "get",
          "Name": "addRegion1导出法线",
          "PathParams": [],
          "PoiCode": "api001011",
          "PreviewUrl": "http://t-apigw.twins.tencent.com/proxy/0086a573-570c-4cdb-b945-e22df31eb589/TBIM-Server/services/3D-51fab803-8651-4520-902a-53fab2db61fa/v20230814142004569/Tiles/tileset.json",
          "QueryParams": [],
          "RequestHeaders": [],
          "ResponseBody": "",
          "ResponseHeaders": [],
          "Status": 1,
          "Style": 1,
          "Type": 8,
          "WorkspaceId": "1016"
        },
        {
          "Address": "/gw/equipment-management-dev/person-mgr-svc-acdb86ce6fdfcdcc/certificateinfo/batch/get",
          "ApiId": "01c5d3ca-fa45-45b9-a4f5-38bf37bbf829",
          "AppId": "23006",
          "ApplyAudit": 0,
          "Body": "W3sidXVpZCI6InpjMTY4NjcyNjlxMjM4MDU2MDE3NDMlLCRrZXkiOiJyb290IiwidHlwZSI6InN0cnVjdCIslmIzRXhwYW5kIjpb0cnVILCJkZXNjcmlwdGlvbiI6IuagueiKgueCuSlSmNoaWxkcmVuljpbeyJ1dWlkljoiemMxNjg2NzI2MjEyMzgwNjM2OTg5NilsmImlkcyIsInR5cGUOiJzdHJpbmciLCJkZXNjcmlwdGlvbiI6ImlkcyIsImxldmVsljowLCJwb3MiOiIwLTAiLCJvcmlcIi6MCwibmV3IjpmYWxzZSwiaG92ZXIiOmZhbHNILCJsYWJlbnI6ImlkcyJ9XSuib3JkZXIiOiJAsImxldmVsljowLCJwb3MiOiIwLTAi",
          "DataAudit": 1,
          "Description": "",
          "IsCommonSpace": false,
          "Method": "post",
          "Name": "CertificateInfo-通过id列表查询",
          "PathParams": [],
          "PoiCode": "api001",
          "PreviewUrl": "http://t-apigw.twins.tencent.com/proxy/01c5d3ca-fa45-45b9-a4f5-38bf37bbf829/gw/equipment-management-dev/person-mgr-svc-acdb86ce6fdfcdcc/certificateinfo/batch/get",
          "QueryParams": [],
          "RequestHeaders": [
            {
              "DefaultValue": "",
              "Dynamic": false,
              "Id": "01c5d3ca-fa45-45b9-a4f5-38bf37bbf829",
              "Name": "app-engine-preview",
              "Required": false,
              "Type": "string",
            }
          ]
        }
      ]
    }
  }
}

```

```
"Value": "1"
},
{
  "DefaultValue": "",
  "Dynamic": false,
  "Id": "01c5d3ca-fa45-45b9-a4f5-38bf37bbf829",
  "Name": "x-sa-account",
  "Required": false,
  "Type": "string",
  "Value": "admin"
},
{
  "DefaultValue": "",
  "Dynamic": false,
  "Id": "01c5d3ca-fa45-45b9-a4f5-38bf37bbf829",
  "Name": "x-sa-device-project-id",
  "Required": false,
  "Type": "string",
  "Value": "1"
},
{
  "DefaultValue": "",
  "Dynamic": false,
  "Id": "01c5d3ca-fa45-45b9-a4f5-38bf37bbf829",
  "Name": "x-sa-project-id",
  "Required": false,
  "Type": "string",
  "Value": "211"
}
],
"ResponseBody": "W3sidXVpZCl6InpjMTY4NjcyNjIxMjM4MDkwMDE1MjgiLCRjZXkiOiJyY290IiwidHlwZSI6InN0cnVjdCIsImZlZXhwYW5klj0cnVlCkZkZXNjcmldGlvbil6luagueiKgucCuSIsImNoaWxkcmVuljpbeyJ1dWlkljoiemMxNjg2Nzl2MjEyMzgwNTIxNjA3MCIsmtleSI6ImNvZGUiLCJ0eXBlljoiaW50IiwizGVzY3JpcHRpb24iOiLov5TIm57ku6PnoIEiLCjsZXZlbcI6MSwicG9zIjoiMCOwliwib3JkZXliOjAsIm5ldyI6ZmFsc2UsImhvdmVyljpmYWxzZSwibGFIZWwiOIlv5TIm57ku6PnoIEifSx7InV1aWQiOij6YzE2ODY3MjYyMTIzODA4NDU3ODM5Iiwia2V5IjoibWVzc2FnZSI6ImN0cmluZSUiOijzdHJpbmciLCkZlXNjcmldGlvbil6luIiOWbnuWkhOeQhua2iOaBryI9LHsidXVpZCl6InpjMTY4NjcyNjIxMjM4MDI0MTIwMDIiLCRjZXkiOiJyZXN1bHQiLCJ0eXBlljoiaW50IiwizGVzY3JpcHRpb24iOiLov5TIm57ku6PnoIEifSx7InV1aWQiOij6YzE2ODY3MjYyMTIzODA4NDI3MTQ2Iiwia2V5IjoiyY2VydGlmaWNhdE5vliwidHlwZSI6InN0cmluZyIsImRlc2NyaXB0aW9uIjoi6K+B5Lmm5ZCN56ewliwibGV2ZWwiOjIsInBvcyl6IjAtMi0liwib3JkZXliOjAsIm5ldyI6ZmFsc2UsImhvdmVyljpmYWxzZSwibGFIZWwiOIlor4Hkuablk3np7AifSx7InV1aWQiOij6YzE2ODY3MjYyMTIzODAxMjM4MDI0Iiwia2V5IjoiyY2VydGlmaWNhdE5vliwidHlwZSI6InN0cmluZyIsImRlc2NyaXB0aW9uIjoi6K+B5Lmm57Yw5Y+3liwibGV2ZWwiOjIsInBvcyl6IjAtMi0liwib3JkZXliOjEsIm5ldyI6ZmFsc2UsImhvdmVyljpmYWxzZSwibGFIZWwiOIlor4Hkuabnvlj7cifSx7InV1aWQiOij6YzE2ODY3MjYyMTIzODA4NDI3MTQ2Iiwia2V5IjoiyY2VydGlmaWNhdE5vliwidHlwZSI6InN0cmluZyIsImRlc2NyaXB0aW9uIjoi6K+B5Lmm57Yw5Y+3liwibGV2ZWwiOjIsInBvcyl6IjAtMi0liwib3JkZXliOjEsIm5ldyI6ZmFsc2UsImhvdmVyljpmYWxzZSwibGFIZWwiOIlvgeS5puWbvueJhy9LHsidXVpZCl6InpjMTY4NjcyNjIxMjM4MDU5NjZlMDciLCRjZXkiOiJyZXJ0aWZpY2F0VHlwZSI6InR5cGUiOijzdHJpbmciLCkZlXNjcmldGlvbil6luivgeS5puexu+WeiylSmxldmVsljoxLCJwb3MiOiilwLTItMyIsIm9yZGVyIjozLCJuzXciOmZhbHNILCJob3ZlciI6ZmFsc2UsImxhYmVsljoi6K+B5Lmm57G75Z6LIn0seyj1dWlkljoiemMxNjg2Nzl2MjEyMzgwMDU2MTAxOCIsImtleSI6ImNlcncpZmljYXRyZWVwZGZl0eVBlcmVlZCIsInR5cGUiOijzdHJpbmciLCkZlXNjcmldGlvbil6luivgeS5puaciaeVioacnylSmxldmVsljoxLCJwb3MiOiilwLTItNCIsIm9yZGVyIjo0LCJuzXciOmZhbHNILCJob3ZlciI6ZmFsc2UsImxhYmVsljoi6K+B5Lmm57pyj5pWl5pyfIn0seyj1dWlkljoiemMxNjg2Nzl2MjEyMzgwNDM5NjZlMDciLCRjZXkiOiJyZXJ0aWZpY2F0VHlwZSI6InN0cmluZyIsImRlc2NyaXB0aW9uIjoi6K+B5Lmm57Yig6Zmk5pe26Ze0liwibGV2ZWwiOjIsInBvcyl6IjAtMi0liwib3JkZXliOjEsIm5ldyI6ZmFsc2UsImhvdmVyljpmYWxzZSwibGFIZWwiOIlLkDpmaTml7bpl7QifSx7InV1aWQiOij6YzE2ODY3MjYyMTIzODA3NjY5MzYxliwia2V5IjoizmlsZUIkTGldZCIsInR5cGUiOijzdHJpbmciLCkZlXNjcmldGlvbil6luimZOS7tmlk5YiX6KgolliwibGV2ZWwiOjIsInBvcyl6IjAtMi03liwib3JkZXliOjEsIm5ldyI6ZmFsc2UsImhvdmVyljpmYWxzZSwibGFIZWwiOIlpYmYkZlZpZOWII+ihqC9LHsidXVpZCl6InpjMTY4NjcyNjIxMjM4MDAwOTY3ODYiLCRjZXkiOiJyZXN1bHQiLCkZlXNjcmldGlvbil6luS4u+mUrizplb/mlbTInoss6Zuq6lqxliwibGV2ZWwiOjIsInBvcyl6IjAtMi04liwib3JkZXliOjgsIm5ldyI6ZmFsc2UsImhvdmVyljpmYWxzZSwibGFIZWwiOIlkuLvpk4s6ZW/5pW05Z6LLOmbquiKsJ9LHsidXVpZCl6InpjMTY4NjcyNjIxMjM4MDcyNTIxODUiLCRjZXkiOiJpcc0RlbgV0ZSI6InR5cGUiOijpbnciLCkZlXNjcmldGlvbil6lumAu+i+keWloOmZpCIsImxldmVsljoxLCJwb3MiOiilwLTItOSIsIm9yZGVyIjo5LCJuzXciOmZhbHNILCJob3ZlciI6ZmFsc2UsImxhYmVsljoi6YC76L6R5Yig6ZmkIn0seyj1dWlkl
```


92ZXliOmZhbHNILCJsYWJlbcI6IuS4u+mUrizplb/mlbTInoss6Zuq6lqxIn0seyj1dWlkljoiemMxNjg2NzEzOTYwMzg5NzU1NzY0MiIsImtleSI6Im1hbmFnZU...


```

"Dynamic": false,
"Id": "02830103-e6af-4109-9197-5cb12b9080bf",
"Name": "a",
"Required": false,
"Type": "value",
"Value": ""
}
],
"RequestHeaders": [],
"ResponseBody": "W3sidXVpZCI6Ijc1Y2ZmMGY2YzVhZDRiNjRiODczYjZkMTAxYjhmNmUxliwia2V5Ijoicm9vdCIsInR5cGUiOiJzdHJ1Y3QiLCJpc0V4cGFuZCI6dHJ1ZSwiZGVzY3JpcHRpb24iOiIiLCJjaGlsZHI6W10slm9yZGVyYjowLCJscXZlbiCl6MCwicG9zljoiMCJ9XQ==",
"ResponseHeaders": [],
"Status": 1,
"Style": 1,
"Type": 6,
"Workspaceld": "1015"
},
{
"Address": "/gw/equipment-management-dev/matter-mgr-svc-be5340126b570de4/mattertype/batch/delete",
"ApiId": "031a8044-844f-4322-8ea1-78a452b942c4",
"AppId": "23006",
"ApplyAudit": 0,
"Body": "W3sidXVpZCI6InpjMTY4NjcxMzkwOTlwMTcwMzU0MTkiLCJrZXkiOiJyb290IiwidHlwZSI6InN0cnVjdCIsImZlZXh5YW5klj0cnVILCJkZXNjcmlwdGlvbil6luagueiKgucCuSIsImNoaWxkcmVuljpbeyJ1dWkljoiemMxNjg2NzEzOTA5MjAxMjg1MzQ3OCIsImtleSI6ImlkcyIsInR5cGUiOiJzdHJpbmciLCJkZXNjcmlwdGlvbil6ImlkcyIsImxldmVsljowLCJwb3MiOiIwLTAiLCJvcmlRiCl6MCwicWibmV3IjpmYWxzZSwiaG92ZXIiOmZhbHNiLCJscXZlbiCl6ImlkcyJ9XSuib3JkZXIiOjAsImxldmVsljowLCJwb3MiOiIwLn1d",
"DataAudit": 1,
"Description": "",
"IsCommonSpace": false,
"Method": "post",
"Name": "MatterType-通过id列表删除",
"PathParams": [],
"PoiCode": "api001",
"PreviewUrl": "http://t-apigw.twins.tencent.com/proxy/031a8044-844f-4322-8ea1-78a452b942c4/gw/equipment-management-dev/matter-mgr-svc-be5340126b570de4/mattertype/batch/delete",
"QueryParams": [],
"RequestHeaders": [
{
"DefaultValue": "",
"Dynamic": false,
"Id": "031a8044-844f-4322-8ea1-78a452b942c4",
"Name": "app-engine-preview",
"Required": false,
"Type": "string",
"Value": "1"
},
{
"DefaultValue": "",
"Dynamic": false,
"Id": "031a8044-844f-4322-8ea1-78a452b942c4",
"Name": "x-sa-account",
"Required": false,
"Type": "string",
"Value": "admin"
},
{
"DefaultValue": "",

```

```
"Dynamic": false,
"Id": "031a8044-844f-4322-8ea1-78a452b942c4",
"Name": "x-sa-device-project-id",
"Required": false,
"Type": "string",
"Value": "1"
},
{
  "DefaultValue": "",
  "Dynamic": false,
  "Id": "031a8044-844f-4322-8ea1-78a452b942c4",
  "Name": "x-sa-project-id",
  "Required": false,
  "Type": "string",
  "Value": "211"
}
],
"ResponseBody": "W3sidXVpZCI6InpjMTY4NjcxMzkwOTlwMTY3OTY3ODciLCJrZXkiOiJyb290liwidHlwZSI6InN0cnVjdClzImZlZXhwY5kjp0cnVILCJkZXNjcmlwdGlvbil6luagueiKgucCuSIsImNoaWxkcmVuljpbeyJ1dWlkIjoiemMxNjg2NzEzOTA5MjAxOTE3MTI1OSIsImtleSI6ImNvZGUiLCJ0eXBlljoiaW50liwiZGVzY3JpcHRpb24iOiLov5Tlm57ku6PnoIEiLCjsZXZlbiCl6MSwicG9zJljoimc0wliwib3JkZXliOiJAsIm5ldyI6ZmFsc2UslmhvdmVyljpmYWxzZSwibGFIZWwiOiL0v5Tlm57ku6PnoIEifSx7InV1aWQioiJ6YzE2ODY3MTM5MDkyMDEwNDUwMzI3Iiwia2V5IjoibWVzc2FnZSIsInR5cGUiOiJzdHJpbmciLCJkZXNjcmlwdGlvbil6lml7bpl7TmiLmILCjsZXZlbiCl6MSwicG9zJljoimc0yIiwib3JkZXliOiJIsIm5ldyI6ZmFsc2UslmhvdmVyljpmYWxzZSwibGFIZWwiOiLml7bpl7TmiLMif0sIm9yZGVyIjowLCJ5ZXZlbiCl6MCwicG9zJljoimcJ9XQ==",
"ResponseHeaders": [],
"Status": 1,
"Style": 1,
"Type": 3,
"Workspaceld": "1016"
}
],
"TotalCount": 416
}
}
}
```

示例2 获取API列表请求示例

获取API列表请求示例

输入示例

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

{
  "PageNumber": 1,
  "PageSize": 10,
  "Type": [
    1
  ],
  "ApplicationToken": "ffCYdM0TZa64H15oczTR2zj5F7S41oc"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "85b33324-1d02-45cc-8250-aa9d82e63544",
    "Result": {
      "ApiInfo": [
        {
          "Address": "/weiling-pubsub/datafusion/",
          "Apild": "013a8423-03f6-4c00-9181-ade7f124ae8f",
          "AppId": "10069",
          "ApplyAudit": 0,
          "Body": "",
          "DataAudit": 0,
          "Description": "",
          "IsCommonSpace": true,
          "Method": "get",
          "Name": "数据融合中心websocket勿删",
          "PathParams": [],
          "PoiCode": "api000",
          "PreviewUrl": "ws://dev-ws.twins.tencent.com/proxy/013a8423-03f6-4c00-9181-ade7f124ae8f/weiling-pubsub/datafusion/",
          "QueryParams": [],
          "RequestHeaders": [],
          "ResponseBody": "",
          "ResponseHeaders": [],
          "Status": 1,
          "Style": 2,
          "Type": 1,
          "Workspaceld": "1015"
        },
        {
          "Address": "/trpc.wetwin.trpc_wetwin_container_svr.AppAPI/DeviceList",
          "Apild": "027d9a7a-1d1d-4974-9992-9a21d2ff843d",
          "AppId": "20134",
          "ApplyAudit": 0,
          "Body": "",
          "DataAudit": 0,
          "Description": "",
          "IsCommonSpace": true,
          "Method": "post",
          "Name": "测试cachess",
          "PathParams": [],
          "PoiCode": "api000",
          "PreviewUrl": "http://dev-apigw.twins.tencent.com/proxy/027d9a7a-1d1d-4974-9992-9a21d2ff843d/trpc.wetwin.trpc_wetwin_container_svr.AppAPI/DeviceList",
          "QueryParams": [],
          "RequestHeaders": [],
          "ResponseBody": "W3sidXVpZCI6IjA3ZjIhZjg3YWVmZm9udCkzLnR5cGUoIjZdHj1Y3QiLCjpc0V4cGFuZCI6dHJlZSwiZGVzY3jpcHRpb24iOiIiLCJjaGlsZHI6IjI0IiwiaS9yZGVyIjowLCJscXZlbi6iMCwicG9zIjoiMCJ9XQ==",
          "ResponseHeaders": [],
          "Status": 1,
          "Style": 1,
          "Type": 1,
          "Workspaceld": "1015"
        },
        {
          "Address": "/trpc.bim.bim_data_bus.TBimOpen",
```

```

"ApiId": "0571bcd5-77ea-458a-ab37-1e9b4ca131f9",
"AppId": "98",
"ApplyAudit": 0,
"Body": "lil=",
"DataAudit": 0,
>Description": "hhh",
"IsCommonSpace": true,
"Method": "post",
>Name": "robbytest",
"PathParams": [],
"PoiCode": "api000002",
"PreviewUrl": "http://dev-apigw.twins.tencent.com/proxy/0571bcd5-77ea-458a-ab37-1e9b4ca131f9/trpc.bim.bim_data_bus.TBimOpen",
"QueryParams": [],
"RequestHeaders": [],
"ResponseBody": "W3sidXVpZCI6IjQ2NzBiZWQ2OGRjMDRmNTg5YmViYjY4NWZhZDY4ZTc4Iiwia2V5Ijoicm9vdCIsInR5cGUiOiJzdHJ1Y3QiLCJpc0V4cGFuZCI6dHJlZSwiZGVzY3JpcHRpb24iOiIiLCJjaGlsZHIibil6W10slm9yZGVyYjowLCJsZXZlbiCl6MCwicG9zIjoicj9XQ==",
"ResponseHeaders": [],
>Status": 1,
>Style": 1,
>Type": 1,
>WorkspaceId": "1015"
},
{
"Address": "/posts/2",
"ApiId": "0fcc883-bd05-4b0d-8252-4f2ce5e7042f",
"AppId": "10052",
"ApplyAudit": 0,
"Body": "",
"DataAudit": 0,
>Description": "123213",
"IsCommonSpace": true,
"Method": "get",
>Name": "集华的接口",
"PathParams": [],
"PoiCode": "api000",
"PreviewUrl": "http://dev-apigw.twins.tencent.com/proxy/0fcc883-bd05-4b0d-8252-4f2ce5e7042f/posts/2",
"QueryParams": [],
"RequestHeaders": [],
"ResponseBody": "W3sidXVpZCI6ImI1NmZmZlwnJlU3ZTQ2N2M4ZDAzMzQ4MTQ2NTgwMWYzIiwia2V5Ijoicm9vdCIsInR5cGUiOiJzdHJ1Y3QiLCJpc0V4cGFuZCI6dHJlZSwiZGVzY3JpcHRpb24iOiIiLCJjaGlsZHIibil6W10slm9yZGVyYjowLCJsZXZlbiCl6MCwicG9zIjoicj9XQ==",
"ResponseHeaders": [],
>Status": 1,
>Style": 1,
>Type": 1,
>WorkspaceId": "1015"
},
{
"Address": "",
"ApiId": "13108ad8-4c73-48f2-a70c-aaded78a9d89",
"AppId": "10070",
"ApplyAudit": 0,
"Body": "",
"DataAudit": 0,
>Description": "",
"IsCommonSpace": false,
"Method": "get",
>Name": "sxy测试",
    
```

```
"PathParams": [],
"PoiCode": "api001",
"PreviewUrl": "http://dev-apigw.twins.tencent.com/proxy/13108ad8-4c73-48f2-a70c-aaded78a9d89",
"QueryParams": [
  {
    "DefaultValue": "",
    "Dynamic": false,
    "Id": "13108ad8-4c73-48f2-a70c-aaded78a9d89",
    "Name": "123123",
    "Required": false,
    "Type": "value",
    "Value": "123123"
  },
  {
    "DefaultValue": "",
    "Dynamic": false,
    "Id": "13108ad8-4c73-48f2-a70c-aaded78a9d89",
    "Name": "123123",
    "Required": false,
    "Type": "value",
    "Value": "123123"
  },
  {
    "DefaultValue": "",
    "Dynamic": false,
    "Id": "13108ad8-4c73-48f2-a70c-aaded78a9d89",
    "Name": "123123",
    "Required": false,
    "Type": "value",
    "Value": "123123"
  },
  {
    "DefaultValue": "",
    "Dynamic": false,
    "Id": "13108ad8-4c73-48f2-a70c-aaded78a9d89",
    "Name": "123123",
    "Required": false,
    "Type": "value",
    "Value": "123123"
  },
  {
    "DefaultValue": "",
    "Dynamic": false,
    "Id": "13108ad8-4c73-48f2-a70c-aaded78a9d89",
    "Name": "123123",
    "Required": false,
    "Type": "value",
    "Value": "123123"
  },
  {
    "DefaultValue": "",
    "Dynamic": false,
    "Id": "13108ad8-4c73-48f2-a70c-aaded78a9d89",
    "Name": "123123",
    "Required": false,
    "Type": "value",
    "Value": "123123"
  },
  {
    "DefaultValue": "",
    "Dynamic": false,
    "Id": "13108ad8-4c73-48f2-a70c-aaded78a9d89",
    "Name": "333333",
    "Required": false,
    "Type": "value",
    "Value": "333333"
  }
]
```

```

    }
  ],
  "RequestHeaders": [],
  "ResponseBody": "W3sidXVpZCl6ljNiZTFiODRmYzk2MDQyYWI5NzQ2ZWE4NzcwOGYyYTk1liwia2V5ljoicm9vdCIsInR5cGUiOiJzdHJ1Y3QiLCJpc0V4cGFuZCl6dHJ1ZSwiZGVzY3JpcHRpb24iOiIlLCJjaGlsZlIibW10slm9yZGVyYlJowLCJsZXZlbnR5bWVudCwG9zljoiMCJ9XQ==",
  "ResponseHeaders": [],
  "Status": 1,
  "Style": 1,
  "Type": 1,
  "WorkspaceId": "1016"
},
{
  "Address": "",
  "ApiId": "136a66f7-a730-40ae-8aad-659f69c449b7",
  "AppId": "20768",
  "ApplyAudit": 0,
  "Body": "",
  "DataAudit": 0,
  "Description": "",
  "IsCommonSpace": true,
  "Method": "get",
  "Name": "测试新增目录接口",
  "PathParams": [],
  "PoiCode": "api000003002",
  "PreviewUrl": "http://dev-apigw.twins.tencent.com/proxy/136a66f7-a730-40ae-8aad-659f69c449b7",
  "QueryParams": [],
  "RequestHeaders": [],
  "ResponseBody": "W3sidXVpZCl6lmYzYTM0MDM2YzUzZDRINDRhNjZyMm5YTFiNDMA3liwia2V5ljoicm9vdCIsInR5cGUiOiJzdHJ1Y3QiLCJpc0V4cGFuZCl6dHJ1ZSwiZGVzY3JpcHRpb24iOiIlLCJjaGlsZlIibW10slm9yZGVyYlJowLCJsZXZlbnR5bWVudCwG9zljoiMCJ9XQ==",
  "ResponseHeaders": [],
  "Status": 1,
  "Style": 1,
  "Type": 1,
  "WorkspaceId": "1015"
},
{
  "Address": "/spaceFusion/appApi/message/batchReport",
  "ApiId": "1388c124-0acc-4cbc-931a-a5bfab400a1c",
  "AppId": "98",
  "ApplyAudit": 0,
  "Body": "",
  "DataAudit": 0,
  "Description": "",
  "IsCommonSpace": true,
  "Method": "post",
  "Name": "应用消息批量上报",
  "PathParams": [],
  "PoiCode": "api000002",
  "PreviewUrl": "http://dev-apigw.twins.tencent.com/proxy/1388c124-0acc-4cbc-931a-a5bfab400a1c/spaceFusion/appApi/message/batchReport",
  "QueryParams": [],
  "RequestHeaders": [],
  "ResponseBody": "",
  "ResponseHeaders": [],
  "Status": 1,
  "Style": 1,
  "Type": 1,
  "WorkspaceId": "1015"
}

```

```

},
{
  "Address": "/trpc.wetwin.trpc_wetwin_container_svr.AppAPI/DescribeAlarmStatusList",
  "ApiId": "1e1005b8-2a79-4f76-97cc-57c819b3751d",
  "AppId": "10051",
  "ApplyAudit": 0,
  "Body": "",
  "DataAudit": 0,
  "Description": "",
  "IsCommonSpace": true,
  "Method": "post",
  "Name": "告警状态列表",
  "PathParams": [],
  "PoiCode": "api000006004",
  "PreviewUrl": "http://dev-apigw.twins.tencent.com/proxy/1e1005b8-2a79-4f76-97cc-57c819b3751d/trpc.wetwin.trpc_wetwin_container_svr.AppAPI/DescribeAlarmStatusList",
  "QueryParams": [],
  "RequestHeaders": [],
  "ResponseBody": "W3sidXVpZCI6JjE5YzQ5YzY2IxoTQ3ZTFhY2I3MTFIMGU4ZmQ4YmRiliwia2V5Jjoicm9vdCIsInR5cGUiOiJzdHJ1Y3QiLCJpc0V4cGFuZCI6dHJ1ZSwiZGVzY3JpcHRpb24iOiIiLCJjaGlsZHIJbil6W10slm9yZGVyYjowLCJscXZlIbCl6MCwicG9zIjoicj9XQ==",
  "ResponseHeaders": [],
  "Status": 1,
  "Style": 1,
  "Type": 1,
  "Workspaceld": "1015"
},
{
  "Address": "",
  "ApiId": "2a0206b6-3094-4c56-964c-568ce2b13a7f",
  "AppId": "10041",
  "ApplyAudit": 0,
  "Body": "",
  "DataAudit": 1,
  "Description": "",
  "IsCommonSpace": true,
  "Method": "",
  "Name": "wewe",
  "PathParams": [],
  "PoiCode": "api0000002",
  "PreviewUrl": "ws://dev-ws.twins.tencent.com/proxy/2a0206b6-3094-4c56-964c-568ce2b13a7f",
  "QueryParams": [],
  "RequestHeaders": [],
  "ResponseBody": "",
  "ResponseHeaders": [],
  "Status": 1,
  "Style": 2,
  "Type": 1,
  "Workspaceld": "1015"
},
{
  "Address": "",
  "ApiId": "32ba90be-a5c9-4e22-8370-df09bb1092e2",
  "AppId": "10001",
  "ApplyAudit": 0,
  "Body": "",
  "DataAudit": 1,
  "Description": "",

```

```

    "IsCommonSpace": true,
    "Method": "post",
    "Name": "abba",
    "PathParams": [],
    "PoiCode": "api000",
    "PreviewUrl": "http://dev-apigw.twins.tencent.com/proxy/32ba90be-a5c9-4e22-8370-df09bb1092e2",
    "QueryParams": [],
    "RequestHeaders": [],
    "ResponseBody": "W3sidXVpZCI6ImQ3ZjI1NDE4NjdhMDQyYWE4MDJlMGUzOGZiNGNmMzlmIiwia2V5ljoicm9vdCl5LnR5cGUlOijzdHJ1Y3QiLCJpc0V4cGFuZCI6dHJlZS5wiZGVzY3JpcHRpb24iOiilLCJjaGlsczhJlbi6W10slm9yZGVyIjowLCJscXZlbnCl6MCwicG9zljoiMCJ9XQ==",
    "ResponseHeaders": [],
    "Status": 1,
    "Style": 1,
    "Type": 1,
    "WorkspaceId": "1015"
  }
],
"TotalCount": 64
}
}
}

```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。

获取租户下的空间列表

最近更新時間：2023-11-30 08:49:55

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

获取租户下的空间列表

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：DescribeWorkspaceList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
ApplicationToken	是	String	应用token 示例值：XaaJvayJyusfUwuoJ2OIT5Z2YAVb6bVo
WorkspaceId	否	Integer	工作空间id，非必填，填了则表示根据id进行批量查询 示例值：1101

3. 输出参数

参数名称	类型	描述
Result	WorkspaceInfoList	项目空间列表 示例值：{ "List": [{ "ChineseName": "lv测试空间", "Description": "lv测试空间", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "1744387200", "ValidityStartTime": "1682524800", "WorkspaceId": 1092 }, { "ChineseName": "RayDataWeb", "Description": "RayDataWeb测试使用测完后删除", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "2000390400", "ValidityStartTime": "1684684800", "WorkspaceId": 1166 }, { "ChineseName": "华东项目空间", "Description": "演练项目空间", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "1693411200", "ValidityStartTime": "1682870400", "WorkspaceId": 1175 }, { "ChineseName": "ces", "Description": "123", "EnglishName": "", "IsCommWorkspace": false, "Status": 1, "TenantId": 100055, "ValidityEndTime": "1688400000", "ValidityStartTime": "1688313600", "WorkspaceId": 1524 }, { "ChineseName": "公共空间", "Description": "租户默认工作空间", "EnglishName": "comm_workspace", "IsCommWorkspace": true, "Status": 0, "TenantId": 100055, "ValidityEndTime": "4834882335", "ValidityStartTime": "1681282335", "WorkspaceId": 1015 }, { "ChineseName": "华南园区项目", "Description": "华南园区项目", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "2153664000", "ValidityStartTime": "1680278400", "WorkspaceId": 1016 }, { "ChineseName": "华北园区项目", "Description": "华北园区项目", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "2153664000", "ValidityStartTime": "1680278400", "WorkspaceId": 1018 }, { "ChineseName": "test", "Description": "ceshi", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "1747584000", "ValidityStartTime": "1685289600", "WorkspaceId": 1263 }, { "ChineseName": "report2test", "Description": "report2test", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "2192976000", "ValidityStartTime": "1687795200", "WorkspaceId": 1439 }] }
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 查询项目空间列表

查询项目空间列表

输入示例

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

{
  "ApplicationToken": "XggjvayjyusfUwuoj2OIT4Z2YAVq6bVo"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d04a9bab-dd91-4999-8584-bbc4d5203d50",
    "Result": {
      "List": [
        {
          "ChineseName": "lv测试空间",
          "Description": "lv测试空间",
          "EnglishName": "",
          "IsCommWorkspace": false,
          "Status": 0,
          "TenantId": 100055,
          "ValidityEndTime": "1744387200",
          "ValidityStartTime": "1682524800",
          "WorkspaceId": 1092
        },
        {
          "ChineseName": "RayDataWeb",
          "Description": "RayDataWeb测试使用完后删除",
          "EnglishName": "",
          "IsCommWorkspace": false,
          "Status": 0,
          "TenantId": 100055,
          "ValidityEndTime": "2000390400",
          "ValidityStartTime": "1684684800",
          "WorkspaceId": 1166
        },
        {
          "ChineseName": "华东项目空间",
          "Description": "演练项目空间",
          "EnglishName": "",
          "IsCommWorkspace": false,
          "Status": 0,
          "TenantId": 100055,
          "ValidityEndTime": "1693411200",
          "ValidityStartTime": "1682870400",
          "WorkspaceId": 1175
        }
      ]
    }
  }
}
```

```
{
  "ChineseName": "ces",
  "Description": "123",
  "EnglishName": "",
  "IsCommWorkspace": false,
  "Status": 1,
  "TenantId": 100055,
  "ValidityEndTime": "1688400000",
  "ValidityStartTime": "1688313600",
  "WorkspaceId": 1524
},
{
  "ChineseName": "公共空间",
  "Description": "租户默认工作空间",
  "EnglishName": "comm_workspace",
  "IsCommWorkspace": true,
  "Status": 0,
  "TenantId": 100055,
  "ValidityEndTime": "4834882335",
  "ValidityStartTime": "1681282335",
  "WorkspaceId": 1015
},
{
  "ChineseName": "华南园区项目",
  "Description": "华南园区项目",
  "EnglishName": "",
  "IsCommWorkspace": false,
  "Status": 0,
  "TenantId": 100055,
  "ValidityEndTime": "2153664000",
  "ValidityStartTime": "1680278400",
  "WorkspaceId": 1016
},
{
  "ChineseName": "华北园区项目",
  "Description": "华北园区项目",
  "EnglishName": "",
  "IsCommWorkspace": false,
  "Status": 0,
  "TenantId": 100055,
  "ValidityEndTime": "2153664000",
  "ValidityStartTime": "1680278400",
  "WorkspaceId": 1018
},
{
  "ChineseName": "test",
  "Description": "ceshi",
  "EnglishName": "",
  "IsCommWorkspace": false,
  "Status": 0,
  "TenantId": 100055,
  "ValidityEndTime": "1747584000",
  "ValidityStartTime": "1685289600",
  "WorkspaceId": 1263
},
{
  "ChineseName": "report2test",
```

```

    "Description": "report2test",
    "EnglishName": "",
    "IsCommWorkspace": false,
    "Status": 0,
    "TenantId": 100055,
    "ValidityEndTime": "2192976000",
    "ValidityStartTime": "1687795200",
    "WorkspaceId": 1439
  }
]
}
}
}
}

```

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
InternalServerError	内部错误。
InvalidParameter	参数错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id

获取文件下载URL

最近更新时间：2023-12-12 02:59:22

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

获取文件下载URL

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：DescribeFileDownloadURL。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间Id 示例值：1016
FileId	是	String	文件Id 示例值：a
ApplicationToken	是	String	应用token 示例值：sFEmY6U

3. 输出参数

参数名称	类型	描述
Result	FileDownloadURL	文件下载URL地址 示例值：{"FileURL":"https://xxx.xxx"}
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 获取文件下载地址

获取文件下载地址返回结果

输入示例

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

{
  "WorkspaceId": 1016,
  "FileId": "3b81cb18-34ce-40f4-99e7-8fb0b9f6eaeab",
```

```
"ApplicationToken": "YzenL5LdGoxQM5gqJfCCoMDeGqUSsY78"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d87de8da-6209-43df-8d87-7e6260850c5f",
    "Result": {
      "FileURL": "https://xxx.xxx"
    }
  }
}
```

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
FailedOperation.DBError	操作DB失败
FailedOperation.StorageInitFailed	对象存储 - 初始化失败
FailedOperation.URLGenerateFailed	对象存储 - url生成失败
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id

文件上传接口

最近更新時間：2023-11-30 08:49:57

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

文件上傳接口

默認接口請求頻率限制：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

以下請求參數列表僅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

參數名稱	必選	類型	描述
Action	是	String	公共參數，本接口取值：DescribeFileUploadURL。
Version	是	String	公共參數，本接口取值：2023-04-27。
Region	是	String	公共參數，詳見產品支持的 地域列表 。
WorkspaceId	是	Integer	工作空間Id 示例值：1016
FileName	是	String	文件名稱 示例值：a
FileSize	是	Integer	文件大小 示例值：10
ApplicationToken	是	String	應用token 示例值：xxx
FileMD5	否	String	文件MD5 示例值：e096797dd8d220861b03f8df7386037b
SaveType	否	String	存儲類型 示例值：TEST2
FileExpireTime	否	Integer	過期時間，過期時間戳，精確到秒的時間戳，noExpireFlag為false時必填 示例值：1680259132
NoExpireFlag	否	Boolean	永不過期標記 示例值：false

3. 输出参数

參數名稱	類型	描述
Result	FileUploadURL	獲取文件上傳地址結果 示例值：{"URL": "", "FileId": "", "DownloadURL": ""}
RequestId	String	唯一請求 ID，每次請求都會返回。定位問題時需要提供該次請求的 RequestId。

4. 示例

示例1 获取上传文件返回地址

正常返回结果

输入示例

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

{
  "WorkspaceId": 1016,
  "FileName": "a",
  "FileSize": 10,
  "FileExpireTime": 1693411200,
  "ApplicationToken": "YzenL5LdGoxQM5gqJfCCoMDeGqUSsY78"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "6d75ec52-741b-4579-bb7e-1c554e3a153c",
    "Result": {
      "DownloadURL": "https://wetwin-managr-xxx",
      "FileId": "dcdc04b4-8417-4b4c-8658-38ea09a67dde",
      "UploadURL": "https://wetwin-manager-xxx"
    }
  }
}
```

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
FailedOperation.DBError	操作DB失败
FailedOperation.StorageInitFailed	对象存储 - 初始化失败
FailedOperation.URLGenerateFailed	对象存储 - url生成失败
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id

创建应用授权令牌

最近更新时间：2023-11-30 08:50:00

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

调用方应用，创建调用租户API的授权令牌。

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：CreateApplicationToken。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ApplicationId	是	Integer	应用id 示例值：10048
Nonce	是	String	一个随机数或者时间戳，用于防止重放攻击，每个请求唯一，建议用uuid 示例值：f9b25481-e060-43e3-ad26-cb4e5f2e5d67
TenantId	是	Integer	租户id 示例值：100055
RequestTime	是	Integer	请求时间，当前时间的unix毫秒时间戳 示例值：1691028252348
Signature	是	String	签名方法见用户使用文档 示例值：cfed53f585017cf8f2559e3ecd84b3b4

3. 输出参数

参数名称	类型	描述
Result	ApplicationTokenInfo	应用令牌信息 示例值：Oq9lShbmhCT3FLnemDCecrXAnAHQe6Sj
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 获取应用凭证示例

输入示例

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

```
{
  "ApplicationId": 10001,
  "Nonce": "e222d195-ab43-4b68-b115-0ad71488f7ed",
  "TenantId": 10000,
  "RequestTime": 1693797989,
  "Signature": "RDcVIT1tOqq5V3K0nnRjuTpFcVL8wlyb"
}
```

输出示例

```
{
  "Response": {
    "Result": {
      "Token": "886c2341d97352aa027916ec0e0339fa"
    },
    "RequestId": "e222d195-ab43-4b68-b115-0ad71488f7ed"
  }
}
```

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
InternalServerError.BusinessLogicError	业务逻辑错误
InternalServerError.Timeout	超时
InvalidParameter.InvalidRequestTime	请求时间非法
InvalidParameterValue.InvalidApplicationId	应用id非法
InvalidParameterValue.InvalidNonce	申请应用token时，传递的nonce参数非法

错误码	描述
InvalidParameterValue.InvalidRequestTime	请求时间非法
InvalidParameterValue.InvalidSignature	签名非法

BIM空间服务相关接口 查询建筑列表

最近更新時間：2023-11-30 08:50:12

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

查詢建築列表

默認接口請求頻率限制：20次/秒。

推薦使用 API Explorer

<> 點擊調試

API Explorer 提供了在線調用、簽名驗證、SDK 代碼生成和快速檢索接口等能力。您可查看每次調用的請求內容和返回結果以及自動生成 SDK 調用示例。

2. 輸入參數

以下請求參數列表僅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

參數名稱	必選	類型	描述
Action	是	String	公共參數 ，本接口取值：DescribeBuildingList。
Version	是	String	公共參數 ，本接口取值：2023-04-27。
Region	是	String	公共參數 ，詳見產品支持的 地域列表 。
WorkspaceId	是	String	項目空間id 示例值："1109"
ApplicationToken	是	String	應用token 示例值："LMb6piRXpioR7nmwFhXpvZd4GC4wf0B4"
HasModel	否	Boolean	是否有模型文件 示例值：true
SpaceCodes.N	否	Array of String	空間編碼 示例值：""

3. 輸出參數

參數名稱	類型	描述
Result	BuildingListRes	查詢建築列表出參 示例值：{"BuildingProfileList": [{"Address": "北京市,丰台区樣例模型", "BuildingId": "2da7d294-483e-4033-a99a-49f7323fd6ef", "BuildingName": "樣例模型1", "Latitude": 39.927326, "Longitude": 116.39177, "SpaceCode": "000009"}, {"Address": "1#樓", "BuildingId": "dfad5b19-79d8-4e40-9a8c-8efa32a1f425", "BuildingName": "1#樓", "Latitude": 39.92442, "Longitude": 116.38269, "SpaceCode": "000007"}]}
RequestId	String	唯一請求 ID，每次請求都會返回。定位問題時需要提供該次請求的 RequestId。

4. 示例

示例1 查詢建築列表

查詢建築列表成功響應示例

輸入示例

```
POST / HTTP/1.1
Host: weilingwith.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeBuildingList
<公共請求參數>
```

```
{
  "WorkspaceId": "1016",
  "HasModel": true,
  "ApplicationToken": "ZRCJHdnhqEUEqO1vyskCgWimPucHhREV"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "6fb5c58b-975c-40a0-ae17-437ddfa0d234",
    "Result": {
      "BuildingProfileList": [
        {
          "Address": "北京市,东城区forest-12",
          "BuildingId": "f207b473-3eb7-4b7c-873b-9e2549354234",
          "BuildingName": "forest-12",
          "Latitude": 39.92836,
          "Longitude": 116.416336,
          "SpaceCode": "000144"
        },
        {
          "Address": "天津市,河东区test-1111",
          "BuildingId": "112acca6-c448-413c-90cf-9da1ea880274",
          "BuildingName": "test-1111",
          "Latitude": 39.128525,
          "Longitude": 117.251396,
          "SpaceCode": "000141"
        },
        {
          "Address": "北京市,东城区forest-11",
          "BuildingId": "c5f12d6b-485f-4c31-82e9-e7d9a6053859",
          "BuildingName": "forest-11",
          "Latitude": 39.92836,
          "Longitude": 116.416336,
          "SpaceCode": "000140"
        },
        {
          "Address": "北京市,东城区forest-10",
          "BuildingId": "44ebb092-8a3e-46c2-a887-803d2e94aee9",
          "BuildingName": "forest-10",
          "Latitude": 39.92836,
          "Longitude": 116.416336,
          "SpaceCode": "000139"
        },
        {
          "Address": "河北省,秦皇岛市,山海关区testdsds",
          "BuildingId": "e89d3f4f-9f0e-4e34-9c94-18589870caa4",
          "BuildingName": "testdsds",
          "Latitude": 39.97888,
          "Longitude": 119.775185,
          "SpaceCode": "000102"
        },
        {
          "Address": "北京市,东城区国际医疗保健中心V1",
          "BuildingId": "5826a9ca-b14c-40de-95da-be635cfe846e",

```

```

"BuildingName": "国际医疗保健中心V1",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000096"
},
{
"Address": "北京市,东城区天健一号楼",
"BuildingId": "d04de59d-3bf1-42a8-a894-74ffe6a2dd25",
"BuildingName": "天健一号楼",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000094"
},
{
"Address": "北京市,东城区测试建筑数据0818",
"BuildingId": "00110b6b-2d67-4cf5-ae08-de0fe5118a4d",
"BuildingName": "测试建筑数据0818",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000092"
},
{
"Address": "广东省,深圳市,南山区前海大厦",
"BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"BuildingName": "前海test",
"Latitude": 22.527027,
"Longitude": 113.899185,
"SpaceCode": "000185"
},
{
"Address": "河北省,邯郸市,峰峰矿区小别墅",
"BuildingId": "044b1384-f965-440a-875b-0a64b61c2388",
"BuildingName": "小别墅",
"Latitude": 36.419296,
"Longitude": 114.21257,
"SpaceCode": "000183"
},
{
"Address": "河北省,秦皇岛市,北戴河区成大中心",
"BuildingId": "41151071-5230-422b-9ff3-dff735172823",
"BuildingName": "成大中心",
"Latitude": 39.83491,
"Longitude": 119.48449,
"SpaceCode": "000176"
},
{
"Address": "天津市,河西区gzip",
"BuildingId": "ec4a49d9-e710-4bd9-a694-f3a99dd4796a",
"BuildingName": "gzip",
"Latitude": 39.10968,
"Longitude": 117.22338,
"SpaceCode": "000175"
},
{
"Address": "广东省,深圳市,南山区腾讯滨海大厦",
"BuildingId": "5fe9a07f-a632-4da9-abb7-9133a18e9505",
"BuildingName": "滨海3层",
    
```

```
"Latitude": 22.522806,
"Longitude": 113.93534,
"SpaceCode": "000174"
},
{
"Address": "北京市,西城区江森",
"BuildingId": "988f3768-933b-4fe7-81ef-174a62e4aa0c",
"BuildingName": "江森",
"Latitude": 39.9126,
"Longitude": 116.36585,
"SpaceCode": "000173"
},
{
"Address": "山西省,长治市,上党区DDDD",
"BuildingId": "1f8315b1-16f1-4c1f-b6e7-e35eb735d802",
"BuildingName": "DDDD",
"Latitude": 36.05315,
"Longitude": 113.051346,
"SpaceCode": "000171"
},
{
"Address": "天津市,河东区小智新设备挂接0628",
"BuildingId": "377a53a6-c5a7-445d-a3b7-c2d721f95408",
"BuildingName": "小智新设备挂接0628",
"Latitude": 39.128525,
"Longitude": 117.251396,
"SpaceCode": "000162"
},
{
"Address": "北京市,东城区测试IFC数据13",
"BuildingId": "d61c36cd-cb38-4909-9348-679ef2693f38",
"BuildingName": "测试IFC数据13",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000078"
},
{
"Address": "北京市,东城区测试IFC数据11",
"BuildingId": "8e3afa64-834f-41b7-8539-1e5a7a38dac4",
"BuildingName": "测试IFC数据11",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000074"
},
{
"Address": "北京市,东城区测试IFC数据10",
"BuildingId": "decd96a8-b312-4b10-a609-95e681f2356d",
"BuildingName": "测试IFC数据10",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000073"
},
{
"Address": "北京市,西城区测试IFC数据07",
"BuildingId": "f25d2dbd-f31c-47bb-8ac7-8208864312f1",
"BuildingName": "测试IFC数据07",
"Latitude": 39.9126,
```



```
"Longitude": 116.36585,
"SpaceCode": "000072"
},
{
"Address": "北京市,东城区测试IFC数据04",
"BuildingId": "07771a01-2430-475a-9d93-456031710d95",
"BuildingName": "测试IFC数据04(1)(1)(1)(1)(1)(1)(1)(1)(1)",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000070"
},
{
"Address": "北京市,东城区测试IFC数据05",
"BuildingId": "d635e346-9660-495e-b6a4-522e97b8a985",
"BuildingName": "测试IFC数据05",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000070"
},
{
"Address": "北京市,东城区测试IFC数据06",
"BuildingId": "c5279f0c-8d2a-4593-b1cf-eb24fc671e38",
"BuildingName": "测试IFC数据06(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)(1)",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000054"
},
{
"Address": "北京市,东城区测试IFC数据03",
"BuildingId": "7ccebedd-8f4c-40cd-8537-3c0ce8adf992",
"BuildingName": "测试IFC数据03",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000033"
},
{
"Address": "北京市,东城区测试IFC数据01",
"BuildingId": "12b929ad-0974-4704-9b4b-90df4b7cf0cd",
"BuildingName": "测试IFC数据01",
"Latitude": 39.92836,
"Longitude": 116.416336,
"SpaceCode": "000031"
},
{
"Address": "北京市,朝阳区验证专用",
"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
"BuildingName": "验证专用",
"Latitude": 39.94446,
"Longitude": 116.50771,
"SpaceCode": "000030"
},
{
"Address": "北京市,朝阳区验证在线发布",
"BuildingId": "3d6ce0c9-7263-4287-88f1-c8d72e87968b",
"BuildingName": "验证在线发布",
"Latitude": 39.94446,
"Longitude": 116.50771,
```

```

"SpaceCode": "000021"
},
{
"Address": "江苏省,常州市,武进区测试楼层模型获取",
"BuildingId": "47f85a70-0317-4706-acfc-96b5a2e3a88e",
"BuildingName": "测试楼层模型获取",
"Latitude": 31.701252,
"Longitude": 119.942444,
"SpaceCode": "000012"
},
{
"Address": "广东省,深圳市,南山区测试建筑",
"BuildingId": "2e374df6-7ba7-45d2-a44d-2a92bb2a6668",
"BuildingName": "测试建筑",
"Latitude": 22.533192,
"Longitude": 113.93048,
"SpaceCode": "000003"
}
]
}
}
}

```

示例2 查询建筑列表示例-prod

查询建筑列表示例-prod

输入示例

```

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

{
"WorkspaceId": "1109",
"ApplicationToken": "LMb6piRXpioR7nmwFhXpvZd4GC4wf0B4"
}

```

输出示例

```

{
"Response": {
"RequestId": "16d8c7ac-449a-45d0-8181-15cf06a0e8e1",
"Result": {
"BuildingProfileList": [
{
"Address": "999",
"BuildingId": "bef95f2a-9474-4635-8c66-3a0f0bf0e9b2",
"BuildingName": "999",
"Latitude": 39.902985,
"Longitude": 116.426636,
"SpaceCode": "000011"
},
{
"Address": "312",

```

```
"BuildingId": "a2284645-2c5f-4018-b20f-5e9298c2daec",
"BuildingName": "312",
"Latitude": 39.935,
"Longitude": 116.43041,
"SpaceCode": "000010"
},
{
"Address": "北京市,丰台区样例模型",
"BuildingId": "2da7d294-483e-4033-a99a-49f7323fd6ef",
"BuildingName": "样例模型1",
"Latitude": 39.927326,
"Longitude": 116.39177,
"SpaceCode": "000009"
},
{
"Address": "2322",
"BuildingId": "f54f2bd9-74ae-4bbf-bbf6-5b41c0d170e1",
"BuildingName": "2322",
"Latitude": 39.926315,
"Longitude": 116.34252,
"SpaceCode": "000008"
},
{
"Address": "1#楼",
"BuildingId": "dfad5b19-79d8-4e40-9a8c-8efa32a1f425",
"BuildingName": "1#楼",
"Latitude": 39.92442,
"Longitude": 116.38269,
"SpaceCode": "000007"
},
{
"Address": "32232",
"BuildingId": "3f25ef75-579b-4041-afcb-a328b106b99e",
"BuildingName": "32232",
"Latitude": 39.94474,
"Longitude": 116.34115,
"SpaceCode": "000006"
},
{
"Address": "3333dgn",
"BuildingId": "daeacc05-67e8-494f-8c7c-ff3678f490ad",
"BuildingName": "3333dgn",
"Latitude": 39.913147,
"Longitude": 116.39677,
"SpaceCode": "000005"
},
{
"Address": "912",
"BuildingId": "c4830dd5-d8b3-4d21-8179-17dd838ef221",
"BuildingName": "912",
"Latitude": 39.91262,
"Longitude": 116.35694,
"SpaceCode": "000381"
},
{
"Address": "",
"BuildingId": "c6a46ced-e0ea-4ab3-9c33-216defb26153",
```

```

"BuildingName": "空间编辑器我的建筑",
"Latitude": 0,
"Longitude": 0,
"SpaceCode": "000019"
},
{
"Address": "",
"BuildingId": "05d663eb-c96f-4c92-9bf8-9792d4df9cee",
"BuildingName": "场地",
"Latitude": 0,
"Longitude": 0,
"SpaceCode": "000004"
},
{
"Address": "",
"BuildingId": "6ac8db4a-af6e-4d5e-a8f4-a58e0c8b98e1",
"BuildingName": "建筑",
"Latitude": 0,
"Longitude": 0,
"SpaceCode": "000003"
}
}
}
}
}
}
    
```

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.TokenNotFound	未找到该token信息
InternalServerError.ApiGatewayInternalError	API网关内部错误

错误码	描述
InternalServerError.ApiRequestPathParameterError	该api访问路径匹配参数异常，请检查请求路径
InternalServerError.BIMAppInternalError	BIM服务内部错误
InternalServerError.BusinessLogicError	业务逻辑错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询建筑三维模型

最近更新时间：2023-11-30 08:50:11

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询建筑三维模型

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeBuildingModel。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
BuildingId	是	String	建筑id 示例值："82d5fb5a-52d0-4636-a225-d46245e911ef"
WorkspaceId	是	String	项目空间id 示例值："1016"
ApplicationToken	是	String	应用token 示例值："ZRCJHdnhqEUEqO1vyskCgWimPucHhREV"

3. 输出参数

参数名称	类型	描述
Result	BuildingModelRes	建模模型信息出参 示例值：{ "Models": [{ "ElementId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ElementName": "验证专用", "ModelType": "MODEL_ORIGIN", "ModelUrl": "https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/cb6bad63-0855-4dc2-8908-20518627479b.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqqVvLagaL8IBZq5896tTxOfjbl2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3Ddbac70b180e5a2d8c343506dae755a01669ffb04" }, { "ElementId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ElementName": "验证专用-外壳", "ModelType": "MODEL_HULL", "ModelUrl": "https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/ff847c36-89f8-4644-9c1f-d265bd5d5167.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqqVvLagaL8IBZq5896tTxOfjbl2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3D7af0ab15cab83314e44026fedc0927467d07b776" }, { "ElementId": "0b00f88f-36f1-4ae3-a7b4-5db12332c005", "ElementName": "-1F", "ModelType": "MODEL_STOREY", "ModelUrl": "https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/2ee86ae4-41a1-4c45-840b-4e9eae1b83a1.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqqVvLagaL8IBZq5896tTxOfjbl2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3De9b58a5293201d24c19e2bc28b49404035b569c1" }] }
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 查询建筑模型

查询建筑模型成功响应示例

输入示例

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

{
  "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
  "WorkspaceId": "1016",
  "ApplicationToken": "ZRCJHdnhqEUEqO1vyskCgWimPucHhREV"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "8f193fa9-506a-421d-afe9-fdde94cc5849",
    "Result": {
      "Models": [
        {
          "ElementId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
          "ElementName": "验证专用",
          "ModelType": "MODEL_ORIGIN",
          "ModelUrl": "https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/cb6bad63-0855-4dc2-8908-20518627479b.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqqVvLagaL8IBZq5896tTxOfjbl2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3Dbac70b180e5a2d8c343506dae755a01669ffb04"
        },
        {
          "ElementId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
          "ElementName": "验证专用-外壳",
          "ModelType": "MODEL_HULL",
          "ModelUrl": "https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/ff847c36-89f8-4644-9c1f-d265bd5d5167.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqqVvLagaL8IBZq5896tTxOfjbl2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3D7af0ab15cab83314e44026fedc0927467d07b776"
        },
        {
          "ElementId": "0b00f88f-36f1-4ae3-a7b4-5db12332c005",
          "ElementName": "-1F",
          "ModelType": "MODEL_STOREY",
          "ModelUrl": "https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/2ee86ae4-41a1-4c45-840b-4e9eae1b83a1.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqqVvLagaL8IBZq5896tTxOfjbl2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3De9b58a5293201d24c19e2bc28b49404035b569c1"
        },
        {
          "ElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
          "ElementName": "RF",
          "ModelType": "MODEL_STOREY",
          "ModelUrl": "https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/c96a40b9-f5b6-4523-b58d-15c79f4a4d3b.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqqVvLagaL8IBZq5896tTxOfjbl2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3D1dfc789fc1770b898e4fccca61a234449f2c8ff3"
        }
      ]
    }
  }
}
```



```
"ElementName": "rvm-外壳",
"ModelType": "MODEL_HULL",
"ModelUrl": "https://bim-1317915939.cos.ap-guangzhou.myqcloud.com/100001/1124/dbdb4717-6629-4a7e-85a7-b720eae82cdd.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDbGuX8hEh5PbnBXvFhy2rSqtYyA7pa6N%26q-sign-time%3D1693827299%3B1694345699%26q-key-time%3D1693827299%3B1694345699%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3D9ba5a3ed7b71a008fb0e11b194f4c1948bc9c9e9"
},
{
"ElementId": "679279f4adaa431daf04bab2ea30ad28",
"ElementName": "/STABILIZER",
"ModelType": "MODEL_STOREY",
"ModelUrl": "https://bim-1317915939.cos.ap-guangzhou.myqcloud.com/100001/1124/d0c77567-597e-48fd-9afb-7afddb5d1786.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDbGuX8hEh5PbnBXvFhy2rSqtYyA7pa6N%26q-sign-time%3D1693827299%3B1694345699%26q-key-time%3D1693827299%3B1694345699%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3Ddf00cf2e40bd39a3d26ce10e19f910d6e425c16f"
},
{
"ElementId": "4cf3865c355048d0be117466b89ff5ae",
"ElementName": "/SAMPLE-ADMIN",
"ModelType": "MODEL_STOREY",
"ModelUrl": "https://bim-1317915939.cos.ap-guangzhou.myqcloud.com/100001/1124/b04791d4-f6ad-43f5-a36d-74eba01b2090.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDbGuX8hEh5PbnBXvFhy2rSqtYyA7pa6N%26q-sign-time%3D1693827299%3B1694345699%26q-key-time%3D1693827299%3B1694345699%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3D8f62e68d47eabbfe029af1d736815aa248691cf2"
},
{
"ElementId": "42ac80b2834442f89c9c73e723986012",
"ElementName": "/PSI-SAMPLE",
"ModelType": "MODEL_STOREY",
"ModelUrl": "https://bim-1317915939.cos.ap-guangzhou.myqcloud.com/100001/1124/12816356-f38e-4252-a8d0-6222df7d9421.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDbGuX8hEh5PbnBXvFhy2rSqtYyA7pa6N%26q-sign-time%3D1693827299%3B1694345699%26q-key-time%3D1693827299%3B1694345699%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3D39f7013ce9d5e7d2d36532e2b54ab4edfbc605f"
}
}
}
}
}
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询建筑信息

最近更新时间：2023-11-30 08:50:10

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询建筑信息

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：DescribeBuildingProfile。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
BuildingId	是	String	建筑id 示例值："2e374df6-7ba7-45d2-a44d-2a92bb2a6668"
WorkspaceId	是	String	项目空间id 示例值："1016"
ApplicationToken	是	String	应用token 示例值："0VGYLwQaH4ejNnwSMIfYtl0zlstprsdZ"

3. 输出参数

参数名称	类型	描述
Result	BuildingProfileRes	查询建筑信息出参 示例值：{ "BuildingProfile": { "Address": "广东省,深圳市,南山区测试建筑", "BuildingId": "2e374df6-7ba7-45d2-a44d-2a92bb2a6668", "BuildingName": "测试建筑", "Latitude": 22.533192, "Longitude": 113.93048, "SpaceCode": "000003" } }
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 查询建筑信息

查询建筑信息成功响应

输入示例

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

{
```

```

"BuildingId": "2e374df6-7ba7-45d2-a44d-2a92bb2a6668",
"WorkspaceId": "1016",
"ApplicationToken": "0VGYLwQaH4ejNnwSMIfYtl0zlstprsdZ"
}
    
```

输出示例

```

{
  "Response": {
    "RequestId": "9d3442d1-0df1-48ef-beba-872e07b2b1dd",
    "Result": {
      "BuildingProfile": {
        "Address": "广东省,深圳市,南山区测试建筑",
        "BuildingId": "2e374df6-7ba7-45d2-a44d-2a92bb2a6668",
        "BuildingName": "测试建筑",
        "Latitude": 22.533192,
        "Longitude": 113.93048,
        "SpaceCode": "000003"
      }
    }
  }
}
    
```

示例2 查询建筑信息示例-prod

查询建筑信息示例-prod

输入示例

```

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

{
  "BuildingId": "9e98dc3920df4230b431404222fefe37",
  "WorkspaceId": "1124",
  "ApplicationToken": "SI5ZDmVuHzhN52o8n1KWagl7sLdrLfN7"
}
    
```

输出示例

```

{
  "Response": {
    "RequestId": "966d76de-83dc-4d6a-bd17-75505fe0af7e",
    "Result": {
      "BuildingProfile": {
        "Address": "陕西省,西安市,碑林区rvm",
        "BuildingId": "9e98dc3920df4230b431404222fefe37",
        "BuildingName": "rvm",
        "Latitude": 34.212498,
        "Longitude": 108.88995,
        "SpaceCode": "000147"
      }
    }
  }
}
    
```

```
}  
}
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询分页构件信息

最近更新时间：2023-11-30 08:50:10

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询分页构件信息

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：DescribeElementProfilePage。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
BuildingId	是	String	建筑id 示例值："956bd069-c802-4bbb-b325-18d30d7bcd3c"
PageNumber	是	Integer	页码 示例值：1
PageSize	是	Integer	页容量 示例值：10
WorkspaceId	是	String	项目空间id 示例值："1016"
ApplicationToken	是	String	应用token 示例值："ZRCJHdnhqEUEqO1vyskCgWimPucHhREV"
ParentElementIds.N	否	Array of String	父级构件id 示例值：""
Level	否	Integer	空间层级 示例值：7
SpaceTypeCode	否	String	空间分类代码 示例值：""
EntityTypes.N	否	Array of String	构件类型 示例值：["TBIM_SPACE", "TBIM_BUILDINGSTOREY"]
IncludeDelete	否	Boolean	是否包含已删除构件 示例值：false
StartTime	否	Integer	时间范围-开始 示例值：1683981442
EndTime	否	Integer	时间范围-结束 示例值：1699879043

3. 输出参数

参数名称	类型	描述
------	----	----

参数名称	类型	描述
Result	ElementProfilePageRes	分页查询构件出参 示例值: { "List": [{ "BottomHeight": -100, "BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c", "ElementDesc": "", "ElementId": "1Kjh8u2qf4uRSkCPoKigTI", "ElementName": "1F (结) -0.100", "EntityType": "TBIM_BUILDINGSTOREY", "IsDelete": 0, "Level": 7, "ParentElementId": "956bd069-c802-4bbb-b325-18d30d7bcd3c", "Sort": 1, "SpaceCode": "000185004", "SpacePoild": "", "SpaceTypeCode": "", "SpaceTypeName": "" },], "TotalCount": 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 查询分页构件信息

查询分页构件信息成功响应示例

输入示例

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

{
  "BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
  "PageNumber": 1,
  "PageSize": 10,
  "EntityTypes": [
    "TBIM_SPACE",
    "TBIM_BUILDINGSTOREY"
  ],
  "Workspaceld": "1016",
  "ApplicationToken": "ZRCJHdnhqEUEqO1vyskCgWimPucHhREV"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "f21ec011-9dde-478f-acd4-6adecb8a1c32",
    "Result": {
      "List": [
        {
          "BottomHeight": -100,
          "BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
          "ElementDesc": "",
          "ElementId": "1Kjh8u2qf4uRSkCPoKigTI",
          "ElementName": "1F (结) -0.100",
          "EntityType": "TBIM_BUILDINGSTOREY",
          "IsDelete": 0,
          "Level": 7,
          "ParentElementId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
          "Sort": 1,
          "SpaceCode": "000185004",
          "SpacePoild": "",
          "SpaceTypeCode": ""
        }
      ]
    }
  }
}
```

```

"SpaceTypeName": ""
},
{
"BottomHeight": 0,
"BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"ElementDesc": "",
"ElementId": "3Mph_S_ef13f4LIAEQPvzw",
"ElementName": "1",
"EntityType": "TBIM_SPACE",
"IsDelete": 0,
"Level": 8,
"ParentElementId": "1Kjh8u2qf4uRSkCPoKj36O",
"Sort": 2,
"SpaceCode": "00018500100000001",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
},
{
"BottomHeight": 0,
"BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"ElementDesc": "",
"ElementId": "1Kjh8u2qf4uRSkCPoKj36O",
"ElementName": "1F 0.000",
"EntityType": "TBIM_BUILDINGSTOREY",
"IsDelete": 0,
"Level": 7,
"ParentElementId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"Sort": 3,
"SpaceCode": "000185001",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
},
{
"BottomHeight": 5900,
"BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"ElementDesc": "",
"ElementId": "1Kjh8u2qf4uRSkCPoKiWWT",
"ElementName": "2F (结) 5.900",
"EntityType": "TBIM_BUILDINGSTOREY",
"IsDelete": 0,
"Level": 7,
"ParentElementId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"Sort": 4,
"SpaceCode": "000185007",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
},
{
"BottomHeight": 6000,
"BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"ElementDesc": "",
"ElementId": "1Kjh8u2qf4uRSkCPoKj38P",
"ElementName": "2F 6.000",
"EntityType": "TBIM_BUILDINGSTOREY",
    
```



```
"IsDelete": 0,
"Level": 7,
"ParentElementId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"Sort": 5,
"SpaceCode": "000185002",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
},
{
"BottomHeight": 11000,
"BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"ElementDesc": "",
"ElementId": "1Kjh8u2qf4uRSkCPoKlbnh",
"ElementName": "3F(结)",
"EntityType": "TBIM_BUILDINGSTOREY",
"IsDelete": 0,
"Level": 7,
"ParentElementId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"Sort": 6,
"SpaceCode": "000185003",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
},
{
"BottomHeight": 11100,
"BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"ElementDesc": "",
"ElementId": "1Kjh8u2qf4uRSkCPoKiVXm",
"ElementName": "F3",
"EntityType": "TBIM_BUILDINGSTOREY",
"IsDelete": 0,
"Level": 7,
"ParentElementId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"Sort": 7,
"SpaceCode": "000185005",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
},
{
"BottomHeight": 16200,
"BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"ElementDesc": "",
"ElementId": "1Kjh8u2qf4uRSkCPoKiVdy",
"ElementName": "F4",
"EntityType": "TBIM_BUILDINGSTOREY",
"IsDelete": 0,
"Level": 7,
"ParentElementId": "956bd069-c802-4bbb-b325-18d30d7bcd3c",
"Sort": 8,
"SpaceCode": "000185006",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
}
```

```

    ],
    "TotalCount": 8
  }
}
}

```

示例2 查询分页构件信息示例-prod

查询分页构件信息示例-prod

输入示例

```

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

{
  "BuildingId": "9e98dc3920df4230b431404222fefe37",
  "PageNumber": 1,
  "PageSize": 5,
  "WorkspaceId": "1124",
  "ApplicationToken": "SI5ZDmVuHzhN52o8n1KWagl7sLdrLfN7"
}

```

输出示例

```

{
  "Response": {
    "RequestId": "aa86cd58-027d-46b4-8d59-a7e15c733f18",
    "Result": {
      "List": [
        {
          "BottomHeight": 0,
          "BuildingId": "9e98dc3920df4230b431404222fefe37",
          "ElementDesc": "",
          "ElementId": "9e98dc3920df4230b431404222fefe37",
          "ElementName": "rvm",
          "EntityType": "TBIM_BUILDING",
          "IsDelete": 0,
          "Level": 6,
          "ParentElementId": "",
          "Sort": 1,
          "SpaceCode": "000147",
          "SpacePoild": "86162",
          "SpaceTypeCode": "",
          "SpaceTypeName": ""
        },
        {
          "BottomHeight": 0,
          "BuildingId": "9e98dc3920df4230b431404222fefe37",
          "ElementDesc": "/SAMPLE-ADMIN built by rvm",
          "ElementId": "4cf3865c355048d0be117466b89ff5ae",
          "ElementName": "/SAMPLE-ADMIN",
          "EntityType": "TBIM_BUILDINGSTOREY",
          "IsDelete": 0,

```

```
"Level": 0,
"ParentElementId": "9e98dc3920df4230b431404222fefe37",
"Sort": 2,
"SpaceCode": "",
"SpacePoild": "861620006",
"SpaceTypeCode": "",
"SpaceTypeName": ""
},
{
"BottomHeight": 0,
"BuildingId": "9e98dc3920df4230b431404222fefe37",
"ElementDesc": "/HS-ADMIN/ADMIN built by rvm",
"ElementId": "f5dfeea83847436a98dd02957b283d0d",
"ElementName": "/HS-ADMIN/ADMIN",
"EntityType": "TBIM_BUILDINGELEMENTPROXY",
"IsDelete": 0,
"Level": 0,
"ParentElementId": "4cf3865c355048d0be117466b89ff5ae",
"Sort": 3,
"SpaceCode": "",
"SpacePoild": "8616200060004",
"SpaceTypeCode": "",
"SpaceTypeName": ""
},
{
"BottomHeight": 0,
"BuildingId": "9e98dc3920df4230b431404222fefe37",
"ElementDesc": "/GRID-STABILIZER/1 built by rvm",
"ElementId": "b3f9777344214077a65d3673b17578bc",
"ElementName": "/GRID-STABILIZER/1",
"EntityType": "TBIM_BUILDINGELEMENTPROXY",
"IsDelete": 0,
"Level": 0,
"ParentElementId": "6ab5d7d6f6264824a1318435a0a09b4c",
"Sort": 4,
"SpaceCode": "",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
},
{
"BottomHeight": 0,
"BuildingId": "9e98dc3920df4230b431404222fefe37",
"ElementDesc": "CYLINDER 2 of EQUIPMENT /P1502B built by rvm",
"ElementId": "fc1a114cf16d4f999f4825294a6b378f",
"ElementName": "CYLINDER 2 of EQUIPMENT /P1502B",
"EntityType": "TBIM_BUILDINGELEMENTPROXY",
"IsDelete": 0,
"Level": 0,
"ParentElementId": "2fa2c48f06ee40a5a8a24f296925e532",
"Sort": 5,
"SpaceCode": "",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
]
```

```
"TotalCount": 2501
```

```
}
}
}
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询构件树

最近更新时间：2023-11-30 08:50:09

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询构件树

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：DescribeElementProfileTree。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
BuildingId	是	String	建筑id 示例值："82d5fb5a-52d0-4636-a225-d46245e911ef"
WorkspaceId	是	String	项目空间id 示例值："1016"
ApplicationToken	是	String	应用token 示例值："OVGYLwQaH4ejNnwSMIfYtl0zlstprsdZ"
ElementId	否	String	父级构件id 示例值："61049c1b-158d-41bd-9f0b-df30a50401bd"
Level	否	Integer	构件级别 示例值：8
SpaceTypeCode	否	String	空间分类代码 示例值：""

3. 输出参数

参数名称	类型	描述
Result	ElementProfileTreeRes	构件树出参 示例值：{"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd", "Root": {"Children": [{"Children": [], "ElementProfile": {"BottomHeight": 3000, "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ElementDesc": "", "ElementId": "6042d426-6973-47ff-8ce1-3a63ef9f3c2f", "ElementName": "", "EntityType": "TBIM_SLASHWALL", "IsDelete": 0, "Level": 8, "ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd", "Sort": 1, "SpaceCode": "00003000300000001", "SpacePoild": "", "SpaceTypeCode": "", "SpaceTypeName": ""}], "ElementProfile": {"BottomHeight": 3000, "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ElementDesc": "", "ElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd", "ElementName": "RF", "EntityType": "TBIM_BUILDINGSTOREY", "IsDelete": 0, "Level": 7, "ParentElementId": "", "Sort": 1, "SpaceCode": "000030003", "SpacePoild": "", "SpaceTypeCode": "", "SpaceTypeName": ""}}}]}
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 查询构件树

查询构件树成功响应示例

输入示例

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

{
  "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
  "ElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
  "WorkspaceId": "1016",
  "ApplicationToken": "0VGYLwQaH4ejNnwSMIfYtI0zlstprsdZ"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "473e2070-39bf-4e12-836b-c821799c9cd9",
    "Result": {
      "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
      "ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
      "Root": {
        "Children": [
          {
            "Children": [],
            "ElementProfile": {
              "BottomHeight": 3000,
              "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
              "ElementDesc": "",
              "ElementId": "6042d426-6973-47ff-8ce1-3a63ef9f3c2f",
              "ElementName": "",
              "EntityType": "TBIM_SLASHWALL",
              "IsDelete": 0,
              "Level": 8,
              "ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
              "Sort": 1,
              "SpaceCode": "00003000300000001",
              "SpacePoild": "",
              "SpaceTypeCode": "",
              "SpaceTypeName": ""
            }
          }
        ],
        "Children": [
          {
            "Children": [],
            "ElementProfile": {
              "BottomHeight": 3000,
              "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
              "ElementDesc": "",
              "ElementId": "f408f4c7-6adc-42fd-9485-d3322d5caf77",
              "ElementName": ""
            }
          }
        ]
      }
    }
  }
}
```

```

"EntityType": "TBIM_SLASHWALL",
"IsDelete": 0,
"Level": 8,
"ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
"Sort": 2,
"SpaceCode": "00003000300000002",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [
{
"Children": [
{
"Children": [],
"ElementProfile": {
"BottomHeight": 3000,
"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
"ElementDesc": "",
"ElementId": "2fb8f038-d0d1-4c11-b290-30f4f7ccf5e0",
"ElementName": "",
"EntityType": "TBIM_DOOR",
"IsDelete": 0,
"Level": 0,
"ParentElementId": "f95c3dcd-dfe4-4828-aab7-6cee05fe2578",
"Sort": 1,
"SpaceCode": "",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
}
],
"ElementProfile": {
"BottomHeight": 3000,
"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
"ElementDesc": "",
"ElementId": "f95c3dcd-dfe4-4828-aab7-6cee05fe2578",
"ElementName": "",
"EntityType": "TBIM_OPENINGELEMENT",
"IsDelete": 0,
"Level": 0,
"ParentElementId": "b46107b5-95f4-4115-9f90-8c72173417d7",
"Sort": 1,
"SpaceCode": "",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
}
],
"ElementProfile": {
"BottomHeight": 3000,
"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
"ElementDesc": "",

```

```

"ElementId": "b46107b5-95f4-4115-9f90-8c72173417d7",
"ElementName": "",
"EntityType": "TBIM_SLASHWALL",
"IsDelete": 0,
"Level": 8,
"ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
"Sort": 3,
"SpaceCode": "00003000300000003",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [],
"ElementProfile": {
"BottomHeight": 3000,
"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
"ElementDesc": "",
"ElementId": "96b96d40-0661-4c5f-a71f-ffd0c5d73791",
"ElementName": "",
"EntityType": "TBIM_SLASHWALL",
"IsDelete": 0,
"Level": 8,
"ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
"Sort": 4,
"SpaceCode": "00003000300000004",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [],
"ElementProfile": {
"BottomHeight": 3000,
"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
"ElementDesc": "",
"ElementId": "b95c73d1-208a-4342-ac54-5627b2021627",
"ElementName": "",
"EntityType": "TBIM_COMMON_DEVICE",
"IsDelete": 0,
"Level": 9,
"ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
"Sort": 5,
"SpaceCode": "00003000300000005",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [],
"ElementProfile": {
"BottomHeight": 3000,
"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
"ElementDesc": "",

```



```
"ElementId": "8ace642a-c45f-401a-890c-6570f08fd97a",
"ElementName": "",
"EntityType": "TBIM_COMMON_DEVICE",
"IsDelete": 0,
"Level": 9,
"ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
"Sort": 6,
"SpaceCode": "00003000300000006",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [],
"ElementProfile": {
"BottomHeight": 3000,
"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
"ElementDesc": "",
"ElementId": "a1162f6d-e8c4-4bd2-af78-dbbdb2918b6d",
"ElementName": "",
"EntityType": "TBIM_COMMON_DEVICE",
"IsDelete": 0,
"Level": 9,
"ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
"Sort": 7,
"SpaceCode": "00003000300000007",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [],
"ElementProfile": {
"BottomHeight": 3000,
"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
"ElementDesc": "",
"ElementId": "8a74aba0-46e3-4097-89e6-7adbcb9ccaef",
"ElementName": "",
"EntityType": "TBIM_CAMERA",
"IsDelete": 0,
"Level": 9,
"ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd",
"Sort": 8,
"SpaceCode": "00003000300000008",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [],
"ElementProfile": {
"BottomHeight": 3000,
"BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
"ElementDesc": "",
```


输入示例

```

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

{
  "BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
  "Level": 7,
  "WorkspaceId": "1124",
  "ApplicationToken": "SI5ZDmVuHzhN52o8n1KWagl7sLdrLfN7"
}
    
```

输出示例

```

{
  "Response": {
    "RequestId": "30078117-d5fa-41a2-b837-bb747aede417",
    "Result": {
      "BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
      "ParentElementId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
      "Root": {
        "Children": [
          {
            "Children": [],
            "ElementProfile": {
              "BottomHeight": 0,
              "BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
              "ElementDesc": "from Rvt",
              "ElementId": "f96fc337-019a-4f91-a73e-169f7e0cd91c",
              "ElementName": "Level : F1",
              "EntityType": "TBIM_BUILDINGSTOREY",
              "IsDelete": 0,
              "Level": 7,
              "ParentElementId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
              "Sort": 1,
              "SpaceCode": "000290001",
              "SpacePoild": "",
              "SpaceTypeCode": "",
              "SpaceTypeName": ""
            }
          }
        ],
        {
          "Children": [],
          "ElementProfile": {
            "BottomHeight": 0,
            "BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
            "ElementDesc": "from Rvt",
            "ElementId": "28ddb220-3d6e-4464-8099-4be964d9cb87",
            "ElementName": "Level : F2",
            "EntityType": "TBIM_BUILDINGSTOREY",
            "IsDelete": 0,
            "Level": 7,
            "ParentElementId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
          }
        }
      }
    }
  }
}
    
```

```
"Sort": 2,
"SpaceCode": "000290002",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [],
"ElementProfile": {
"BottomHeight": 0,
"BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
"ElementDesc": "from Rvt",
"ElementId": "985e2202-2fc9-490c-a80a-587b750c3496",
"ElementName": "Level : 室外地坪",
"EntityType": "TBIM_BUILDINGSTOREY",
"IsDelete": 0,
"Level": 7,
"ParentElementId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
"Sort": 3,
"SpaceCode": "000290003",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [],
"ElementProfile": {
"BottomHeight": 0,
"BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
"ElementDesc": "from Rvt",
"ElementId": "e7cda0c3-5b0c-49e7-b2a6-4a654ca4b07b",
"ElementName": "Level : F3",
"EntityType": "TBIM_BUILDINGSTOREY",
"IsDelete": 0,
"Level": 7,
"ParentElementId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
"Sort": 4,
"SpaceCode": "000290004",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [],
"ElementProfile": {
"BottomHeight": 0,
"BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
"ElementDesc": "from Rvt",
"ElementId": "b66aece1-62e9-49d3-872e-6e66156a853b",
"ElementName": "Level : F4",
"EntityType": "TBIM_BUILDINGSTOREY",
"IsDelete": 0,
"Level": 7,
"ParentElementId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
```

```

"Sort": 5,
"SpaceCode": "000290005",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
},
{
"Children": [],
"ElementProfile": {
"BottomHeight": 0,
"BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
"ElementDesc": "from Rvt",
"ElementId": "073b2c5f-8f62-4dfe-8fe6-bb24679b9803",
"ElementName": "未指定标高",
"EntityType": "TBIM_BUILDINGSTOREY",
"IsDelete": 0,
"Level": 7,
"ParentElementId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
"Sort": 6,
"SpaceCode": "000290001",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
}
],
"ElementProfile": {
"BottomHeight": 0,
"BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
"ElementDesc": "",
"ElementId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
"ElementName": "三层小别墅",
"EntityType": "TBIM_BUILDING",
"IsDelete": 0,
"Level": 6,
"ParentElementId": "",
"Sort": 1,
"SpaceCode": "000290",
"SpacePoild": "",
"SpaceTypeCode": "",
"SpaceTypeName": ""
}
}
}
}
}
}
}
}

```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询构件属性（详情）

最近更新时间：2023-11-30 08:50:08

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询构件属性（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：DescribePropertyList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
BuildingId	是	String	建筑id 示例值："377e4f9d-d20e-4209-b1e8-0ddb32e888c8"
ElementId	是	String	构件id 示例值："b66aece1-62e9-49d3-872e-6e66156a853b"
WorkspaceId	是	String	项目空间id 示例值："1124"
ApplicationToken	是	String	应用token 示例值："SI5ZDmVuHzhN52o8n1KWagl7sLdrLfN7"

3. 输出参数

参数名称	类型	描述
Result	ElementPropertyRes	构件属性信息出参 示例值：{"BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8", "Coordinates": {"Altitude": 0, "Latitude": 39.12852507853361, "Longitude": 117.25139600000003}, "ElementId": "b66aece1-62e9-49d3-872e-6e66156a853b", "ElementName": "Level : F4", "EntityTypeCode": "TBIM_BUILDINGSTOREY", "EntityType": "楼层", "PropertySet": [{"Content": "{Elevation=9500, Story Above=None}", "Description": "", "Name": "Constraints"}, {"Content": "{Computation Height=0}", "Description": "", "Name": "Dimensions"}, {"Content": "{Is ground plane=0}", "Description": "", "Name": "Display"}, {"Content": "{storeyType=TOP_FLOOR, floorHeight=3000, bottomHeight=9500}", "Description": "", "Name": "EDITOR_BUILDING_STOREY_EXTEND"}, {"Content": "{Scope Box=None}", "Description": "", "Name": "Extents"}, {"Content": "{IfcGUID=1LvAmy3\$j18hHP\$GECp3rk, Export to IFC=0}", "Description": "", "Name": "IFC Parameters"}, {"Content": "{Structural=No, Design Option=None, Building Story=Yes}", "Description": "", "Name": "Identity Data"}, {"Content": "{Name=Level : F4, Type=C_上标高+层标, Childs=[Roofs], Family=Level, Type Id=87266, Category=OST_Levels, ClassName=OdBmCDANode, ODAUniqueID=214592, LocalizedName=, Family and Type=Level : C_上标高+层标}", "Description": "", "Name": "Invalid group"}, {"Content": "{Material=None}", "Description": "", "Name": "Materials and Finishes"}], "Translate": {"X": 0, "Y": 0, "Z": 0}}
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 查询构件属性示例-prod

查询构件属性示例-prod

输入示例

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

{
  "BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
  "ElementId": "b66aece1-62e9-49d3-872e-6e66156a853b",
  "WorkspaceId": "1124",
  "ApplicationToken": "S15ZDmVuHzhN52o8n1KWagl7sLdrLfN7"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "da778098-4425-4519-bd0b-589e9e83ef6f",
    "Result": {
      "BuildingId": "377e4f9d-d20e-4209-b1e8-0ddb32e888c8",
      "Coordinates": {
        "Altitude": 0,
        "Latitude": 39.12852507853361,
        "Longitude": 117.25139600000003
      },
      "ElementId": "b66aece1-62e9-49d3-872e-6e66156a853b",
      "ElementName": "Level : F4",
      "EntityTypeCode": "TBIM_BUILDINGSTOREY",
      "EntityTypeName": "楼层",
      "PropertySet": [
        {
          "Content": "{Elevation=9500, Story Above=None}",
          "Description": "",
          "Name": "Constraints"
        },
        {
          "Content": "{Computation Height=0}",
          "Description": "",
          "Name": "Dimensions"
        },
        {
          "Content": "{Is ground plane=0}",
          "Description": "",
          "Name": "Display"
        },
        {
          "Content": "{storeyType=TOP_FLOOR, floorHeight=3000, bottomHeight=9500}",
          "Description": "",
          "Name": "EDITOR_BUILDING_STOREY_EXTEND"
        }
      ]
    }
  }
}
```



```

"Content": "{Scope Box=None}",
"Description": "",
"Name": "Extents"
},
{
"Content": "{IfcGUID=1LvAmy3$j18hHP$GEcP3rk, Export to IFC=0}",
"Description": "",
"Name": "IFC Parameters"
},
{
"Content": "{Structural=No, Design Option=None, Building Story=Yes}",
"Description": "",
"Name": "Identity Data"
},
{
"Content": "{Name=Level : F4, Type=C_上标高+层标, Childs=[Roofs], Family=Level, Type Id=87266, Category=OST_Levels, ClassName=OdBmCD
ANode, ODAUniqueID=214592, LocalizedName=, Family and Type=Level : C_上标高+层标}",
"Description": "",
"Name": "Invalid group"
},
{
"Content": "{Material=None}",
"Description": "",
"Name": "Materials and Finishes"
}
],
"Translate": {
"X": 0,
"Y": 0,
"Z": 0
}
}
}
}
    
```

示例2 查询构件属性（详情）

查询构件属性（详情）成功响应示例

输入示例

```

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

{
"BuildingId": "2e374df6-7ba7-45d2-a44d-2a92bb2a6668",
"ElementId": "2e374df6-7ba7-45d2-a44d-2a92bb2a6668",
"WorkspaceId": "1016",
"ApplicationToken": "fquCgDi2QfNDMINyQ5YPoMR2Zw4tQs2B"
}
    
```

输出示例

```

{
  "Response": {
    "RequestId": "37ff2cfc-5c07-4c14-9f4f-82e52b155754",
    "Result": {
      "BuildingId": "2e374df6-7ba7-45d2-a44d-2a92bb2a6668",
      "Coordinates": {
        "Altitude": 0,
        "Latitude": 22.533192029618775,
        "Longitude": 113.93048000000002
      },
      "ElementId": "2e374df6-7ba7-45d2-a44d-2a92bb2a6668",
      "ElementName": "测试建筑",
      "EntityTypeCode": "TBIM_BUILDING",
      "EntityTypeName": "建筑",
      "PropertySet": [
        {
          "Content": "{overground_floor=0, underground_floor=0, overground_floor_height=0, underground_floor_height=0}",
          "Description": "editor building extend",
          "Name": "EDITOR_BUILDING_EXTEND"
        },
        {
          "Content": "{name=bh1.ifc, size=7384171, type=IFC, pinyin=bh1.ifc, fileKey=/bim/6ced4b4a-e34c-49fb-b8fc-15b63d6fd682.ifc, uploadTS=1685446303474, relTaskId=4058f87f-2bc1-4f28-9c05-25426987dafb}",
          "Description": "building file",
          "Name": "TBIM_BUILDING_FILE"
        },
        {
          "Content": "{NumberOfStoreys=5}",
          "Description": "",
          "Name": "Pset_BuildingCommon"
        },
        {
          "Content": "",
          "Description": "element extend attributes",
          "Name": "TBIM_PROPERTY_EXTEND_ATTR"
        },
        {
          "Content": "{MODEL_HULL=00e79829-d8d1-468b-a497-1a6172be051c.glb, MODEL_ORIGIN=b92563fe-c61f-45df-a9e5-c01ecddb4c77.glb, MODEL_STOREY={30eZ3ov6r4ygu6QtqZ6IN$=5149f7ea-d3ab-470e-9d50-06f190bf2e5a.glb, 30eZ3ov6r4ygu6QtqZ6INI=23268a8e-c01d-4f92-b6ef-a3f5b3740765.glb, 30eZ3ov6r4ygu6QtqZ6INW=955e5c87-dbd5-4197-9eb5-f3216265f36f.glb, 30eZ3ov6r4ygu6QtqZ6INf=8b8fdeda-22d0-44b0-bd09-2dd0592d4713.glb, 30eZ3ov6r4ygu6QtqZ6INs=e72910c7-9d8b-464f-912d-f21fe2b87f2b.glb}}",
          "Description": "",
          "Name": "TBIM_BUILDING_GLB_MODEL"
        }
      ],
      "Translate": {
        "X": 0,
        "Y": 0,
        "Z": 0
      }
    }
  }
}
    
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询场景列表

最近更新时间：2023-11-30 08:50:07

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询场景列表

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：DescribeSceneList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	String	项目空间id 示例值："1016"
ApplicationToken	是	String	应用token 示例值："x1DSknvmd61KzG20nGIBAx2fD4GZB8yh"

3. 输出参数

参数名称	类型	描述
Result	SceneListRes	场景列表出参 示例值：{"SceneList":[{"SceneId":"0abe73dd-d06f-44f5-af82-43dea23af230","SceneName":"三层商务办公楼单位修正","Version":"20230705160410780"},{"SceneId":"e9f2b87f-9078-4546-8d4f-99356ec3f82c","SceneName":"商务办公楼模型","Version":"20230630151240729"}]}
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 查询场景列表

查询场景列表成功响应示例

输入示例

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

{
  "WorkspaceId": "1016",
  "ApplicationToken": "ZRCJHdnhqEUEqO1vyskCgWimPucHhREV"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "b7ab87c7-f4ae-420b-8e0a-ebfe0eb6e99f",
    "Result": {
      "SceneList": [
        {
          "SceneId": "fa08fec8-8bf9-4902-8240-2c5c2777dfe4",
          "SceneName": "测试数据发布",
          "Version": "20230727161859024"
        },
        {
          "SceneId": "d0d34f13-5aaa-4e3f-b9df-0c47a1f8b6e6",
          "SceneName": "测试服务",
          "Version": "20230630144242437"
        },
        {
          "SceneId": "9fe1cb82-78fc-4758-ab03-a31dc0de8a89",
          "SceneName": "幼儿园1F",
          "Version": "20230705155913563"
        },
        {
          "SceneId": "0fd68fbc-79b9-4909-8148-b43f431487f6",
          "SceneName": "测试数据发布1",
          "Version": "20230727163120647"
        },
        {
          "SceneId": "2e730413-7d99-4c1a-8dac-b12d348e9bb6",
          "SceneName": "小别墅单位修正",
          "Version": "20230705155451510"
        },
        {
          "SceneId": "12733040-c69a-4dd1-aa1e-2325bb331602",
          "SceneName": "三层小别野",
          "Version": "20230630151635871"
        },
        {
          "SceneId": "0abe73dd-d06f-44f5-af82-43dea23af230",
          "SceneName": "三层商务办公楼单位修正",
          "Version": "20230705160410780"
        },
        {
          "SceneId": "4738b98c-68d0-4f83-9487-9af9d020d310",
          "SceneName": "测试数据发布2",
          "Version": "20230727163522015"
        },
        {
          "SceneId": "a9d100b1-1273-4d47-b573-6d286381809f",
          "SceneName": "江森服务",
          "Version": "20230630151236394"
        },
        {
          "SceneId": "e9f2b87f-9078-4546-8d4f-99356ec3f82c",
          "SceneName": "商务办公楼模型",

```

```
"Version": "20230630151240729"
},
{
  "ScenelId": "cc079cea-b5d0-4f9d-9b83-3f7b542e6614",
  "SceneName": "BH-test",
  "Version": "20230712150531776"
},
{
  "ScenelId": "bad4abea-5458-4580-a1ff-e0c979cc280d",
  "SceneName": "54-building",
  "Version": "20230727173527939"
},
{
  "ScenelId": "366bb60a-4fa7-4fa1-ad1b-1087f610860b",
  "SceneName": "江森",
  "Version": "20230630184023922"
},
{
  "ScenelId": "ac9ec5a4-3b0f-44f7-a71a-ddff713e0998",
  "SceneName": "测试数据发布3",
  "Version": "20230727163704761"
},
{
  "ScenelId": "8d0b8de6-78cc-450a-b027-b122a812e890",
  "SceneName": "测试",
  "Version": "20230727180522999"
},
{
  "ScenelId": "40379866-aff9-43d9-9e9d-c7c066ade9c7",
  "SceneName": "测试数据发布4",
  "Version": "20230727163735539"
},
{
  "ScenelId": "7ef255bd-0d0d-4bc0-b0d6-9737e063fcfa",
  "SceneName": "54test",
  "Version": "20230727181029004"
},
{
  "ScenelId": "5e1e1ad0-4085-4457-94ca-3e196065776e",
  "SceneName": "我的测试aaa",
  "Version": "20230731171838709"
},
{
  "ScenelId": "61a64cb0-2a63-4dc1-be27-f699b199e137",
  "SceneName": "测试新增服务",
  "Version": "20230801194415857"
},
{
  "ScenelId": "1698626f-a02e-4036-9584-40f93bd8c8fe",
  "SceneName": "三层小别墅验证标高",
  "Version": "20230711195949520"
},
{
  "ScenelId": "805395e7-31d6-4826-98b5-ec376027eee5",
  "SceneName": "54new",
  "Version": "20230731161731660"
},
}
```

```
{
  "ScenelId": "ec3fe40e-0673-403a-8b72-eb7245ff39f5",
  "SceneName": "test322",
  "Version": "20230731170522736"
},
{
  "ScenelId": "dfdbb59a-141c-40d8-a05c-636111f8145e",
  "SceneName": "ga1122",
  "Version": "20230816103702972"
},
{
  "ScenelId": "93effc84-8c4e-491a-8feb-1da4b278069d",
  "SceneName": "测试我的服务啦啦啦",
  "Version": "20230801195016874"
},
{
  "ScenelId": "3ea5e8ec-797b-4dd8-887c-9e85eb272b59",
  "SceneName": "test1000",
  "Version": "20230804153330669"
},
{
  "ScenelId": "e972baef-20c1-479d-aaef-c2b84b50e69c",
  "SceneName": "发布参数测试",
  "Version": "20230804172909247"
},
{
  "ScenelId": "3cc80e6c-0d69-419f-8e80-7e089f4ce286",
  "SceneName": "路桥",
  "Version": "20230808180259076"
},
{
  "ScenelId": "3e305d44-997f-4708-b1fa-264245b68ee4",
  "SceneName": "test0002",
  "Version": "20230804174001351"
},
{
  "ScenelId": "41a5d281-fc2a-46f6-83e4-c108159c8bf0",
  "SceneName": "test003",
  "Version": "20230804180150171"
},
{
  "ScenelId": "d1972cce-21b9-42de-a6ef-eebf6fc1a12c",
  "SceneName": "测2",
  "Version": "20230811145020507"
},
{
  "ScenelId": "a4f17670-3096-4632-a173-763d8eed4b24",
  "SceneName": "test004",
  "Version": "20230804182830425"
},
{
  "ScenelId": "0d24ea55-44e2-4fa1-ad28-92b064d7baac",
  "SceneName": "testdgn",
  "Version": "20230808180741817"
},
{
  "ScenelId": "7ff17051-713e-4c7e-9f61-053554ee8706",
```

```
"SceneName": "vfaaa",
"Version": "20230808105202390"
},
{
"ScenelId": "23e8851c-3da6-40bb-a145-5fb228649a4d",
"SceneName": "testtest",
"Version": "20230807142948556"
},
{
"ScenelId": "f17915fb-e95b-429f-9195-410f2a3a349e",
"SceneName": "原地质",
"Version": "20230808190956761"
},
{
"ScenelId": "f3ceb0d9-9b9e-41fc-a829-48701718743b",
"SceneName": "修正dgn",
"Version": "20230808200842033"
},
{
"ScenelId": "31694f6d-c29b-470a-9fe4-e4db3b882480",
"SceneName": "ceshi",
"Version": "20230809142831983"
},
{
"ScenelId": "8500f89d-5cf6-4170-b6ef-6252b0b96fe5",
"SceneName": "Testtest",
"Version": "20230815104643390"
},
{
"ScenelId": "bd9adb96-8f71-4f97-84fa-19eb49c88948",
"SceneName": "test00w",
"Version": "20230815163012361"
}
]
}
}
}
```

示例2 查询场景列表示例-prod

查询场景列表示例-prod

输入示例

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

{
"WorkspaceId": "1124",
"ApplicationToken": "SI5ZDmVuHzhN52o8n1KWagl7sLdrLfN7"
}
```

输出示例


```
{
  "Response": {
    "RequestId": "e8ac69f5-ee77-41c7-98f0-4c8719ce151e",
    "Result": {
      "SceneList": [
        {
          "SceneId": "b26aeb0c-7715-478f-8270-115a88f4958d",
          "SceneName": "回归2验证",
          "Version": "20230615202042025"
        },
        {
          "SceneId": "84d5fe8a-ea6b-4144-9af5-ad4534589e0b",
          "SceneName": "回归01",
          "Version": "20230615202654784"
        },
        {
          "SceneId": "e41e558d-fc1a-40a8-ba60-867ff02e5092",
          "SceneName": "回归011",
          "Version": "20230615203039251"
        },
        {
          "SceneId": "62add4e8-54a9-439e-8b6a-9c1f13d2f8c7",
          "SceneName": "test(1)(1)(1)(1)",
          "Version": "20230616005825213"
        },
        {
          "SceneId": "baf8418c-1b17-4c32-99e1-002f375a05d0",
          "SceneName": "回归12",
          "Version": "20230615203303801"
        },
        {
          "SceneId": "4335d58f-602b-4f38-afb2-0e44573c0e27",
          "SceneName": "回归01-底标高修改",
          "Version": "20230615213006859"
        },
        {
          "SceneId": "f3f655b5-27a0-401e-a8e1-cfbfbf9738e92",
          "SceneName": "test",
          "Version": "20230616002041546"
        },
        {
          "SceneId": "50accaba-dff7-4ee1-a676-4edc1ea59897",
          "SceneName": "test(1)",
          "Version": "20230616003658547"
        },
        {
          "SceneId": "372ab2d4-b163-4377-9f15-faa63ded0e65",
          "SceneName": "test(1)(1)(1)",
          "Version": "20230616004614656"
        },
        {
          "SceneId": "4fae00c4-def6-41ec-ab88-9497ed87779a",
          "SceneName": "test(1)(1)(1)(1)",
          "Version": "20230616004740399"
        },
        {
```

```
"ScenelId": "6cb48ccb-ac15-4401-a41b-bf2f24ee4e48",
"SceneName": "阜外医院",
"Version": "20230616140720093"
},
{
"ScenelId": "fe2fa6dd-4985-4394-914f-bafc6db4d038",
"SceneName": "好久好久",
"Version": "20230704102712493"
},
{
"ScenelId": "ebd98dcb-e024-4b2a-9c8a-7edd255b45f9",
"SceneName": "测试法线",
"Version": "20230705220800744"
},
{
"ScenelId": "6dd01099-472e-4741-b933-5a50331e9de4",
"SceneName": "换行1",
"Version": "20230704104225382"
},
{
"ScenelId": "1a6f61e4-1616-49f1-bf30-19fdc14f358e",
"SceneName": "导入编辑测试",
"Version": "20230705221915959"
},
{
"ScenelId": "571ac33b-1258-40c3-9bc7-55e508ee384f",
"SceneName": "验证修改",
"Version": "20230705225914902"
},
{
"ScenelId": "54a5b4c9-bc12-489f-84f7-d7d787b250c3",
"SceneName": "验证配准",
"Version": "20230705230128637"
},
{
"ScenelId": "31f257da-62a9-475f-ad55-dcd05334ef5d",
"SceneName": "test(1)",
"Version": "20230616004333025"
},
{
"ScenelId": "b798bdb6-e70c-4f58-b6fd-ecc992845a16",
"SceneName": "验证建筑-bim模型楼层复制逻辑",
"Version": "20230707102558358"
},
{
"ScenelId": "15fdc22c-3e69-4919-8c9f-a434332f7c7b",
"SceneName": "121",
"Version": "20230718145215925"
},
{
"ScenelId": "e7ec508e-4516-4aab-9528-a5644f2490ec",
"SceneName": "滨海外壳集成编辑器数据",
"Version": "20230712193129991"
},
{
"ScenelId": "0e357904-9b42-4f56-9e2b-462ff5f8e3a2",
"SceneName": "郑州CIM地质",
```

```
"Version": "20230713191738653"
},
{
  "ScenelId": "0d6819ce-12a8-4f43-85b4-ea7e2f500824",
  "SceneName": "magTest",
  "Version": "20230713194529513"
},
{
  "ScenelId": "5a968e10-d673-43ad-9813-b7aa2911bd63",
  "SceneName": "郑州CIM-B",
  "Version": "20230714093320141"
},
{
  "ScenelId": "cd2038b5-354d-4154-9e85-df523a3e1de9",
  "SceneName": "郑州CIM-C",
  "Version": "20230714101241138"
},
{
  "ScenelId": "dea2ac05-edea-4845-b280-42d0c2717142",
  "SceneName": "滨海12",
  "Version": "20230717210708851"
},
{
  "ScenelId": "dcb17b4e-0023-47f3-a495-b21e90d78c3d",
  "SceneName": "三层小别墅rvt",
  "Version": "20230717215119464"
},
{
  "ScenelId": "c51819fe-48e3-46e8-9b59-b9a7b5bd6f98",
  "SceneName": "dizhi-test",
  "Version": "20230724213033660"
},
{
  "ScenelId": "c79df355-697e-416f-9bf8-02fdd0daeb38",
  "SceneName": "成大未调整位置",
  "Version": "20230725200938887"
},
{
  "ScenelId": "5ef13ba7-51d2-48bc-996a-af39fccd34b7",
  "SceneName": "地质-zz",
  "Version": "20230726123425489"
},
{
  "ScenelId": "fc98f5c2-2e86-45a8-958e-5b5f8ed74098",
  "SceneName": "rvt模型全量发布",
  "Version": "20230731161418347"
},
{
  "ScenelId": "37275988-564f-4d6a-9a83-f8e66139d85a",
  "SceneName": "地质DGN",
  "Version": "20230815200226260"
},
{
  "ScenelId": "e136dc3d-e31f-4796-960c-3d72d20977f8",
  "SceneName": "731revit",
  "Version": "20230810141952165"
},
}
```

```
{
  "ScenelId": "7ae60a1a-baf8-4a0e-9a38-2ae7fce6b0fc",
  "SceneName": "单个发布rvt",
  "Version": "20230801214349885"
},
{
  "ScenelId": "0afd1428-a2cc-4b76-9ed8-0b0f86b48a8a",
  "SceneName": "测试取样大大",
  "Version": "20230815200657047"
},
{
  "ScenelId": "3642d2dd-2c76-4338-855f-1dbfed759fcc",
  "SceneName": "郑州测试",
  "Version": "20230817160720633"
},
{
  "ScenelId": "1570465b-9b7c-4861-8f16-5fd6ab7fcfb5",
  "SceneName": "test",
  "Version": "20230824192151326"
},
{
  "ScenelId": "922cd922-bf56-4131-aff0-84787aa41e80",
  "SceneName": "成大3",
  "Version": "20230725200606168"
},
{
  "ScenelId": "a04326ea-b7fc-4539-8de0-a2217a5f24f1",
  "SceneName": "阜外医院(1)",
  "Version": "20230828143707186"
},
{
  "ScenelId": "db6a1260-e3aa-4452-900a-d3051b2ddcf3",
  "SceneName": "地质2-618",
  "Version": "20230724221123184"
},
{
  "ScenelId": "d7c983d6-646d-4e33-8f05-b54eff28f00a",
  "SceneName": "阜外",
  "Version": "20230828150633511"
},
{
  "ScenelId": "96beff92-b88d-4cd1-a8f4-4bfcde1d4db7",
  "SceneName": "测试建筑012",
  "Version": "20230828161027282"
},
{
  "ScenelId": "76a3cea8-e4df-4403-8ec3-e680d610270e",
  "SceneName": "测试建筑012-2",
  "Version": "20230828161727992"
},
{
  "ScenelId": "4999cab6-2e60-4786-bc76-c81fd31c7eba",
  "SceneName": "测试大模型发布",
  "Version": "20230828211131266"
},
{
  "ScenelId": "a1b54411-38c2-43ea-88d4-7b38d293539b",
```

```

"SceneName": "未来城市鸟瞰-上传",
"Version": "20230828102844507"
},
{
"ScenelId": "fa046369-b08a-4c4e-a7b2-45fecac1406b",
"SceneName": "工厂类模型",
"Version": "20230901100227099"
}
]
}
}
}
}
    
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询指定空间设备编号列表

最近更新时间：2023-11-30 08:50:07

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询指定空间设备编号列表

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeSpaceDeviceIdList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
ElementIds.N	是	Array of String	构件id列表 示例值：["377a53a6-c5a7-445d-a3b7-c2d721f95408"]
IsCascade	是	Boolean	是否级联 示例值：true
WorkspaceId	是	String	项目空间id 示例值："1016"
PageNumber	是	Integer	页码 示例值：1
PageSize	是	Integer	页容量 示例值：10
ApplicationToken	是	String	应用token 示例值："GUsHDOIP614sDxEIliCx7GIKTqyONoX"

3. 输出参数

参数名称	类型	描述
Result	SpaceDeviceIdListRes	设备ID列表 示例值：{"DeviceIds": ["2b538c57-933f-4820-b1d0-4df70d64aee3", "ad655ba1-975a-4238-94d7-5d2015d1cf1f", "5a1a0b73-7aa3-4699-a893-21846e7853ec", "0a198d6e-40f0-499f-bf99-7a959b7f4ef2"]}
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 查询指定空间设备编号列表

查询指定空间设备编号列表成功响应示例

输入示例

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

```
{
  "ElementIds": [
    "377a53a6-c5a7-445d-a3b7-c2d721f95408"
  ],
  "IsCascade": true,
  "WorkspaceId": "1016",
  "PageNumber": 1,
  "PageSize": 10,
  "ApplicationToken": "GUsHDOIP614sDxEIliCx7GIKTqyONoX"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1c3e7242-f4fb-4000-b8ae-0724b481693d",
    "Result": {
      "DeviceIds": [
        "2b538c57-933f-4820-b1d0-4df70d64aee3",
        "ad655ba1-975a-4238-94d7-5d2015d1cf1f",
        "5a1a0b73-7aa3-4699-a893-21846e7853ec",
        "0a198d6e-40f0-499f-bf99-7a959b7f4ef2"
      ]
    }
  }
}
```

示例2 查询指定空间设备编号列表示例-prod

查询指定空间设备编号列表示例-prod

输入示例

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

```
{
  "ElementIds": [
    "306fc545-fc98-4513-ae6e-751f990ae61a"
  ],
  "IsCascade": true,
  "WorkspaceId": "1133",
  "PageNumber": 1,
  "PageSize": 10,
  "ApplicationToken": "SI5ZDmVuHzhN52o8n1KWagl7sLdrLfn7"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "51ddc6c7-0ee6-4d0d-b95a-dda432859da7",
    "Result": {
      "DeviceIds": [
        "675a9490-dd66-4ac7-b39b-c74977899a1a",
        "14dfa61a-762b-4785-bd39-cd709c6cf3d9",
        "840e3fe7-4eb4-4a64-9036-02ba94956e06",
        "3247728f-15d2-48a2-a0fc-d07a6a6023f0",
        "90f965da-5ed9-469c-bace-7ecd09e2693c",
        "1635422d-5de3-4e38-9f68-94b43fd78185",
        "35bc7027-567a-48d2-a3ee-63c57be9ae9f",
        "749287c4-c48f-42f4-b083-838c1b2fef09",
        "50a294f4-9289-43d0-aa27-efc31c7405ac",
        "3049938c-3048-4fef-a2e8-2843ede35edb"
      ]
    }
  }
}
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询指定空间下设备与构件绑定关系列表

最近更新时间：2023-11-30 08:50:06

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询指定空间下设备与构件绑定关系列表

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：DescribeSpaceDeviceRelationList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
ElementIds.N	是	Array of String	构件id列表 示例值：["82d5fb5a-52d0-4636-a225-d46245e911ef"]
IsCascade	是	Boolean	是否级联 示例值：true
WorkspaceId	是	String	项目空间id 示例值："1016"
PageNumber	是	Integer	页码 示例值：1
PageSize	是	Integer	页容量 示例值：10
ApplicationToken	是	String	应用token 示例值："0VGYLwQaH4ejNnwSMIfYtl0zlstprsdZ"

3. 输出参数

参数名称	类型	描述
Result	SpaceDeviceRelationRes	查询指定空间下设备与构件绑定关系列表输出参 示例值：{"SpaceDeviceRelationList": [{"DeviceId": "1c902558-d360-4a93-9b6f-1b49970534f7", "ElementId": "8a74aba0-46e3-4097-89e6-7adbeb9ccaef"}, {"DeviceId": "b66603af-7f8c-46c0-8712-1353a08d5806", "ElementId": "af7f7d83-17f8-4dff-80e2-a5129a39eab0"}]}
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 查询指定空间下设备与构件绑定关系列表

查询指定空间下设备与构件绑定关系列表成功响应示例

输入示例

```

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

{
  "ElementIds": [
    "82d5fb5a-52d0-4636-a225-d46245e911ef"
  ],
  "IsCascade": true,
  "WorkspaceId": "1016",
  "PageNumber": 1,
  "PageSize": 10,
  "ApplicationToken": "0VGYLwQaH4ejNnwSMIfYtI0zIstprsdZ"
}
    
```

输出示例

```

{
  "Response": {
    "RequestId": "6923a12d-747c-4cf2-99d5-6c74af01cb39",
    "Result": {
      "SpaceDeviceRelationList": [
        {
          "DeviceId": "1c902558-d360-4a93-9b6f-1b49970534f7",
          "ElementId": "8a74aba0-46e3-4097-89e6-7adbeb9ccaef"
        },
        {
          "DeviceId": "b66603af-7f8c-46c0-8712-1353a08d5806",
          "ElementId": "af7f7d83-17f8-4dff-80e2-a5129a39eab0"
        },
        {
          "DeviceId": "282e3b5f-5a08-45cf-ab86-fb5d3ca6ceea",
          "ElementId": "f596e5e3-68fb-48e6-bd74-845d623e600f"
        },
        {
          "DeviceId": "0826a816-e153-4448-873d-b284f9cf6f22",
          "ElementId": "b95c73d1-208a-4342-ac54-5627b2021627"
        },
        {
          "DeviceId": "5f089c7e-0530-44c9-9435-c6aaf92f6a29",
          "ElementId": "8ace642a-c45f-401a-890c-6570f08fd97a"
        },
        {
          "DeviceId": "c0e3f641-6ec6-444e-8ac9-3cfec6755cb0",
          "ElementId": "a1162f6d-e8c4-4bd2-af78-dbbdb2918b6d"
        },
        {
          "DeviceId": "e9e0fe2e-b188-4f42-9a4c-acd63c1d3ac5",
          "ElementId": "14d0e599-396d-4d64-8843-bf4ea10e7f75"
        },
        {
          "DeviceId": "90ad0dc7-2f59-4af3-98db-6d43d5602ff2",
          "ElementId": "36d2a1dc-5aaf-49d3-8151-3327d83a062b"
        }
      ]
    }
  }
}
    
```



```
"DeviceId": "3247728f-15d2-48a2-a0fc-d07a6a6023f0",
"ElementId": "b3ca676b-389e-4bb9-a549-22e6a34b06e9"
},
{
"DeviceId": "90f965da-5ed9-469c-bace-7ecd09e2693c",
"ElementId": "8c06cb9e-ed92-4b26-9491-eb7a615ef3d0"
},
{
"DeviceId": "1635422d-5de3-4e38-9f68-94b43fd78185",
"ElementId": "308adda6-5121-437f-b9d7-a60484cd4f78"
},
{
"DeviceId": "35bc7027-567a-48d2-a3ee-63c57be9ae9f",
"ElementId": "83068b31-ecf0-4e1f-8e21-08588830bb37"
},
{
"DeviceId": "749287c4-c48f-42f4-b083-838c1b2fef09",
"ElementId": "572d813d-64fa-4b50-b2c9-cc1a729f7b90"
},
{
"DeviceId": "50a294f4-9289-43d0-aa27-efc31c7405ac",
"ElementId": "238c7691-0472-468c-9c66-4cc3dbdff880"
},
{
"DeviceId": "3049938c-3048-4fef-a2e8-2843ede35edb",
"ElementId": "3425fbe2-3d6c-471f-9527-b878f35d1f03"
}
]
}
}
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询设备绑定的空间信息

最近更新时间：2023-11-30 08:50:05

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询设备绑定的空间信息

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：DescribeSpaceInfoByDeviceId。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
DeviceId	是	String	设备id 示例值："1c902558-d360-4a93-9b6f-1b49970534f7"
WorkspaceId	是	String	项目空间id 示例值："1016"
ApplicationToken	是	String	应用token 示例值："0VGYLwQaH4ejNnwSMIfYtl0zlstprsdZ"

3. 输出参数

参数名称	类型	描述
Result	DeviceSpaceInfoRes	设备绑定的空间信息出参 示例值：{ "BottomHeight": 3000, "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ElementId": "8a74aba0-46e3-4097-89e6-7adbeb9ccaef", "ElementName": "", "EntityType": "TBIM_CAMERA", "Level": 9, "SpaceCode": "00003000300000008" }
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 查询设备绑定的空间信息

查询设备绑定的空间信息成功响应示例

输入示例

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

{
```

```

"DeviceId": "1c902558-d360-4a93-9b6f-1b49970534f7",
"WorkspaceId": "1016",
"ApplicationToken": "0VGYLwQaH4ejNnwSMIfYtl0zlstprsdZ"
}
    
```

输出示例

```

{
  "Response": {
    "RequestId": "901e3a31-d49c-4570-b96c-dcddee09c22d",
    "Result": {
      "BottomHeight": 3000,
      "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef",
      "ElementId": "8a74aba0-46e3-4097-89e6-7adbeb9ccaef",
      "ElementName": "",
      "EntityType": "TBIM_CAMERA",
      "Level": 9,
      "SpaceCode": "00003000300000008"
    }
  }
}
    
```

示例2 查询设备绑定的空间信息示例-prod

查询设备绑定的空间信息示例-prod

输入示例

```

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

{
  "DeviceId": "3049938c-3048-4fef-a2e8-2843ede35edb",
  "WorkspaceId": "1133",
  "ApplicationToken": "SI5ZDmVuHzhN52o8n1KWagl7sLdrLfN7"
}
    
```

输出示例

```

{
  "Response": {
    "RequestId": "750095c6-29f0-4598-b1d7-16e15d43e774",
    "Result": {
      "BottomHeight": 24750,
      "BuildingId": "306fc545-fc98-4513-ae6e-751f990ae61a",
      "ElementId": "3425fbe2-3d6c-471f-9527-b878f35d1f03",
      "ElementName": "",
      "EntityType": "TBIM_CUSTOMCOLUMN",
      "Level": 8,
      "SpaceCode": "00021804800000470"
    }
  }
}
    
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询设备绑定的空间层级关系

最近更新时间：2023-11-30 08:50:05

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询设备绑定的空间层级关系

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeSpaceRelationByDeviceId。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
DeviceId	是	String	设备id 示例值："ad655ba1-975a-4238-94d7-5d2015d1cf1f"
WorkspaceId	是	String	工作空间id 示例值："1016"
ApplicationToken	是	String	应用token 示例值："GUshDOIP614sDxEIiliCx7GIKTqyONoX"

3. 输出参数

参数名称	类型	描述
Result	SpaceRelationRes	空间层级关系出参 示例值：{"SpaceRelation": {"Children": [{"Children": [{"Children": [], "ElementId": "bdbbf537-d6c4-4b1e-80ae-e0bcaab8b06f", "ElementName": "test_no_100", "Level": 9, "ParentSpaceCode": "000162002", "SpaceCode": "00016200200000003"}], "ElementId": "f3c5ffa5-05e3-4c26-a447-0edf71aa9783", "ElementName": "1F", "Level": 7, "ParentSpaceCode": "000162", "SpaceCode": "000162002"}], "ElementId": "377a53a6-c5a7-445d-a3b7-c2d721f95408", "ElementName": "小智新设备挂接0628", "Level": 6, "ParentSpaceCode": "", "SpaceCode": "000162"}}
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 查询指定空间下设备与构件绑定关系列表

查询指定空间下设备与构件绑定关系列表成功响应示例

输入示例

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

```
{
  "DeviceId": "ad655ba1-975a-4238-94d7-5d2015d1cf1f",
  "WorkspaceId": "1016",
  "ApplicationToken": "GUshDOIP614sDxEIliCx7GIKTqyONoX"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "30b9a52a-1ed8-4d97-b17d-2459b938ca3a",
    "Result": {
      "SpaceRelation": {
        "Children": [
          {
            "Children": [
              {
                "Children": [],
                "ElementId": "bdbbf537-d6c4-4b1e-80ae-e0bcaab8b06f",
                "ElementName": "test_no_100",
                "Level": 9,
                "ParentSpaceCode": "000162002",
                "SpaceCode": "00016200200000003"
              }
            ],
            "ElementId": "f3c5ffa5-05e3-4c26-a447-0edf71aa9783",
            "ElementName": "1F",
            "Level": 7,
            "ParentSpaceCode": "000162",
            "SpaceCode": "000162002"
          }
        ],
        "ElementId": "377a53a6-c5a7-445d-a3b7-c2d721f95408",
        "ElementName": "小智新设备挂接0628",
        "Level": 6,
        "ParentSpaceCode": "",
        "SpaceCode": "000162"
      }
    }
  }
}
```

示例2 查询设备绑定的空间层级关系示例-prod

查询设备绑定的空间层级关系示例-prod

输入示例

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

```
{
  "DeviceId": "3049938c-3048-4fef-a2e8-2843ede35edb",
```

```
"WorkspaceId": "1133",
"ApplicationToken": "SI5ZDmVuHzhN52o8n1KWagl7sLdrLfN7"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "9565acba-fb0e-4ec7-be20-ab274b105fd9",
    "Result": {
      "SpaceRelation": {
        "Children": [
          {
            "Children": [
              {
                "Children": [],
                "ElementId": "3425fbe2-3d6c-471f-9527-b878f35d1f03",
                "ElementName": "",
                "Level": 8,
                "ParentSpaceCode": "000218048",
                "SpaceCode": "00021804800000470"
              }
            ],
            "ElementId": "57e7c90c-e3e2-46e0-b9ee-8084c1889620",
            "ElementName": "7F",
            "Level": 7,
            "ParentSpaceCode": "000218",
            "SpaceCode": "000218048"
          }
        ],
        "ElementId": "306fc545-fc98-4513-ae6e-751f990ae61a",
        "ElementName": "BHDA-1",
        "Level": 6,
        "ParentSpaceCode": "",
        "SpaceCode": "000218"
      }
    }
  }
}
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询空间分类

最近更新时间：2023-11-30 08:50:04

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询空间分类

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：DescribeSpaceTypeList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	String	项目空间ID 示例值："1124"
PageNumber	是	Integer	页码 示例值：1
PageSize	是	Integer	页容量 示例值：10
ApplicationToken	是	String	应用token 示例值："SI5ZDmVuHzhN52o8n1KWagl7sLdrLfN7"

3. 输出参数

参数名称	类型	描述
Result	SpaceTypeListRes	空间分类列表出参 示例值：{"SpaceTypeList": [{"SpaceTypeCode": "DIANTIJIAN", "SpaceTypeName": "电梯间"}, {"SpaceTypeCode": "LOUTIJIAN", "SpaceTypeName": "楼梯间"}, {"SpaceTypeCode": "WODEFANGJIAN", "SpaceTypeName": "我的房间"}, {"SpaceTypeCode": "XIUXISHI", "SpaceTypeName": "休息室"}]}
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 查询空间分类

查询空间分类成功响应示例

输入示例

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

<公共请求参数>

```
{
  "WorkspaceId": "1016",
  "PageNumber": 1,
  "PageSize": 10,
  "ApplicationToken": "ZRCJHdnhqEUEqO1vyskCgWimPucHhREV"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "dbb6362d-4e7c-459f-a6e9-ecef04842722",
    "Result": {
      "SpaceTypeList": [
        {
          "SpaceTypeCode": "CESHILEIXING",
          "SpaceTypeName": "测试类型"
        },
        {
          "SpaceTypeCode": "HAHAHA",
          "SpaceTypeName": "哈哈哈哈哈"
        }
      ]
    }
  }
}
```

示例2 查询空间分类示例-prod
查询空间分类示例-prod
输入示例

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

{
  "WorkspaceId": "1124",
  "PageNumber": 1,
  "PageSize": 10,
  "ApplicationToken": "SI5ZDmVuHzhN52o8n1KWagl7sLdrLfN7"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "21770e5e-a17e-4023-8cc9-51129f6cd37a",
    "Result": {
      "SpaceTypeList": [
        {
```

```

"SpaceTypeCode": "DIANTIJIAN",
"SpaceTypeName": "电梯间"
},
{
"SpaceTypeCode": "LOUTIJIAN",
"SpaceTypeName": "楼梯间"
},
{
"SpaceTypeCode": "WODEFANGJIAN",
"SpaceTypeName": "我的房间"
},
{
"SpaceTypeCode": "XIUXISHI",
"SpaceTypeName": "休息室"
}
]
}
}
}
    
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询租户楼栋数量和楼栋建筑面积

最近更新时间：2023-11-30 08:50:03

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询租户楼栋数量和楼栋建筑面积

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：DescribeTenantBuildingCountAndArea。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceIdList.N	是	Array of String	租户所有工作空间ID列表 示例值：["1015", "1016", "1018"]
ApplicationToken	是	String	应用token 示例值："ZRCJHdnhqEUEqO1vyskCgWimPucHhREV"

3. 输出参数

参数名称	类型	描述
Result	SpaceDataTotalStatsRes	租户所有项目空间楼栋数量与建筑面积统计结果 示例值：{ "BuildingArea": 202928.77732501808, "BuildingCount": 137 }
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 查询租户楼栋数量和楼栋建筑面积

查询租户楼栋数量和楼栋建筑面积成功响应示例

输入示例

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

{
  "WorkspaceIdList": [
    "1015",
    "1016",
    "1018"
  ]
}
```



```
],
"ApplicationToken": "ZRCJHdnhqEUEqO1vyskCgWimPucHhREV"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "44e3188a-67a3-4a92-8e7b-517459c29816",
    "Result": {
      "BuildingArea": 202928.77732501808,
      "BuildingCount": 137
    }
  }
}
```

示例2 查询租户楼栋数量和楼栋建筑面积示例-prod

查询租户楼栋数量和楼栋建筑面积示例-prod

输入示例

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

{
  "WorkspaceIdList": [
    "1124",
    "1133"
  ],
  "ApplicationToken": "S15ZDmVuHzhN52o8n1KWagl7sLdrLfn7"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "fa570973-a1d2-4fbc-807e-a0546d7819cc",
    "Result": {
      "BuildingArea": 682550.335429272,
      "BuildingCount": 239
    }
  }
}
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

查询项目空间楼栋数量与建筑面积

最近更新时间：2023-11-30 08:50:02

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询项目空间楼栋数量与建筑面积

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：DescribeWorkspaceBuildingCountAndArea。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
WorkspaceIdList.N	是	Array of String	工作空间ID列表 示例值：["1016", "1015", "1018"]
ApplicationToken	是	String	应用token 示例值："fquCgDi2QfNDMINyQ5YPoMR2Zw4tQs2B"

3. 输出参数

参数名称	类型	描述
Result	SpaceDataListStatsRes	查询项目空间楼栋数量与建筑面积出参 示例值：{"List": [{"BuildingArea": 0, "BuildingCount": 2, "WorkspaceId": "1018", "WorkspaceName": "华北园区项目"}, {"BuildingArea": 202928.77732501808, "BuildingCount": 127, "WorkspaceId": "1016", "WorkspaceName": "华南园区项目"}, {"BuildingArea": 0, "BuildingCount": 8, "WorkspaceId": "1015", "WorkspaceName": "公共空间"}]}
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 查询项目空间楼栋数量与建筑面积

查询项目空间楼栋数量与建筑面积成功响应示例

输入示例

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

```
{
  "WorkspaceIdList": [
    "1016",
```

```
"1015",
"1018"
],
"ApplicationToken": "fquCgDi2QfNDMINyQ5YPoMR2Zw4tQs2B"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "42569a1b-01e1-4064-9288-b89ad76d55b7",
    "Result": {
      "List": [
        {
          "BuildingArea": 0,
          "BuildingCount": 2,
          "WorkspaceId": "1018",
          "WorkspaceName": "华北园区项目"
        },
        {
          "BuildingArea": 202928.77732501808,
          "BuildingCount": 127,
          "WorkspaceId": "1016",
          "WorkspaceName": "华南园区项目"
        },
        {
          "BuildingArea": 0,
          "BuildingCount": 8,
          "WorkspaceId": "1015",
          "WorkspaceName": "公共空间"
        }
      ]
    }
  }
}
```

示例2 查询项目空间楼栋数量与建筑面积示例-prod

查询项目空间楼栋数量与建筑面积-prod

输入示例

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

{
  "WorkspaceIdList": [
    "1124",
    "1133"
  ],
  "ApplicationToken": "SI5ZDmVuHzhN52o8n1KWagI7sLdrLfN7"
}
```

输出示例

```

{
  "Response": {
    "RequestId": "94a7a0d6-b2d4-4da7-a39d-ed79489e09c2",
    "Result": {
      "List": [
        {
          "BuildingArea": 35990.08433069048,
          "BuildingCount": 141,
          "WorkspaceId": "1124",
          "WorkspaceName": "测试空间"
        },
        {
          "BuildingArea": 646560.2510985816,
          "BuildingCount": 98,
          "WorkspaceId": "1133",
          "WorkspaceName": "体验项目空间"
        }
      ]
    }
  }
}
    
```

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
InternalServerError.BIMAppInternalError	BIM服务内部错误
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
ResourceNotFound.ElementNotFound	未查询到构件信息

孪生门户相关接口

查询租户组织部门列表

最近更新时间：2023-11-30 08:50:02

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询租户组织部门列表

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：DescribeTenantDepartmentList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Offset	是	Integer	翻页页码 示例值：1
Limit	是	Integer	翻页大小 示例值：10
ApplicationToken	是	String	token 示例值：xxx
TenantId	否	String	租户ID 示例值：10001
UpdateAt	否	Integer	更新时间戳，单位秒 示例值：1680836659
DepartmentId	否	String	部门ID 示例值：12
Cursor	否	String	用户id 示例值：财务部

3. 输出参数

参数名称	类型	描述
Result	SsoDepartmentsResult	返回数据
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 查询租户组织部门列表

查询租户组织部门列表

输入示例

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

```
{
  "Offset": 1,
  "Limit": 1,
  "DepartmentId": "6",
  "ApplicationToken": "zD3G1qtO9Fy4mu5HAMydrXNHBputBQVS"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "fbb81f22-548b-43e9-afad-08c595058205",
    "Result": {
      "Departments": [
        {
          "DepartmentId": "32",
          "Name": "外部测试",
          "ParentDepartmentId": "6"
        }
      ],
      "Total": 2
    }
  }
}
```

示例2 查询组织部门列表

查询组织部门列表

输入示例

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

```
{
  "Offset": 1,
  "Limit": 1,
  "DepartmentId": "6",
  "ApplicationToken": "zD3G1qtO9Fy4mu5HAMydrXNHBputBQVS"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3e2b7483-9dc0-48af-8d5a-95bebecf8c3",

```

```
"Result": {
  "Departments": [
    {
      "DepartmentId": "32",
      "Name": "外部测试",
      "ParentDepartmentId": "6"
    }
  ],
  "Total": 2
}
```

示例3 组织部门列表

组织部门列表

输入示例

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

{
  "Offset": 1,
  "Limit": 1,
  "DepartmentId": "6",
  "ApplicationToken": "zD3G1qtO9Fy4mu5HAMydrXNH8putBQVS"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "86a86966-f838-44ec-b1e4-469429cf0b61",
    "Result": {
      "Departments": [
        {
          "DepartmentId": "32",
          "Name": "外部测试",
          "ParentDepartmentId": "6"
        }
      ],
      "Total": 2
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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
InternalServerError.UnknownError	未知错误
InvalidParameter	参数错误。

查询租户人员列表

最近更新时间：2023-11-30 08:50:01

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询租户人员列表

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：DescribeTenantUserList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Offset	是	Integer	翻页页码 示例值：1
Limit	是	Integer	翻页大小 示例值：10
ApplicationToken	是	String	token 示例值：xxx
TenantId	否	String	租户ID 示例值：10001
UpdateAt	否	Integer	更新时间戳，单位秒 示例值：1681314106
DepartmentId	否	String	部门ID 示例值：12
Cursor	否	String	用户id 示例值：0d0a98561b3c4f72963eba5c5fcf56e1
Status	否	Integer	状态，0，获取所有数据，1正常启用，2禁用 示例值：1
WorkspaceId	否	String	项目空间id 示例值：1016
Keyword	否	String	关键词 示例值：test
NoRecursive	否	String	是否递归获取子级数据，0需要，1不需要，默认为0 示例值：1

3. 输出参数

参数名称	类型	描述
Result	SsoUserResult	返回数据

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 查询租户人员列表

查询租户人员列表

输入示例

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

{
  "Offset": 1,
  "Limit": 1,
  "TenantId": "1",
  "ApplicationToken": "Z2GWSzYAcnKt1zIK65FQz5XsFk5nTKqx"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "2a9b97e6-3f19-43ea-9ec6-37985e4eac99",
    "Result": {
      "Total": 28,
      "Users": [
        {
          "BelongTeam": 0,
          "CreateAt": 1691565476,
          "DepartmentId": "6",
          "DepartmentName": "产品部",
          "DepartmentUserId": 431,
          "Email": "test@tencent.com",
          "Password": "",
          "Phone": "159***6666",
          "RealName": "test",
          "Status": 1,
          "TenantId": "100055",
          "UpdateAt": 1691745205,
          "UserGroup": "",
          "UserId": "0d0a98561b3c4f72963eba5c5fcf56e1",
          "UserName": "",
          "UserType": "99"
        }
      ]
    }
  }
}
```

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.TokenNotFound	未找到该token信息
InternalServerError.ApiGatewayInternalError	API网关内部错误
InternalServerError.UnknownError	未知错误
InvalidParameter	参数错误。

查询项目空间人员列表

最近更新时间：2023-11-30 08:50:00

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

查询项目空间人员列表

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：DescribeWorkspaceUserList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Offset	是	Integer	翻页页码 示例值：1
Limit	是	Integer	翻页大小 示例值：10
WorkspaceId	是	String	工作空间ID 示例值：1016
ApplicationToken	是	String	token 示例值：xxx
TenantId	否	String	租户ID 示例值：10001
UpdateAt	否	Integer	更新时间戳，单位秒 示例值：1689755305

3. 输出参数

参数名称	类型	描述
Result	SsoTeamUserResult	返回数据
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 查询项目空间人员列表

查询项目空间人员列表

输入示例

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

X-TC-Action: DescribeWorkspaceUserList

<公共请求参数>

```
{
  "Offset": 1,
  "Limit": 10,
  "WorkspaceId": "1016",
  "ApplicationToken": "o9ftuUNOFkOVf2o5etj0nHou396ufAes"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "68f43238-01a6-46c6-9204-f23cc3983756",
    "Result": {
      "Total": 3,
      "Users": [
        {
          "CreateAt": 1692081934,
          "DepartmentId": "6",
          "DepartmentName": "产品部",
          "Email": "test1@tencent.com",
          "LinkFilter": 1,
          "Phone": "159****6666",
          "RealName": "test1",
          "Status": 1,
          "TenantId": "100055",
          "UserId": "0d0a98561b3c4f72963eba5c5fcf56e1",
          "UserType": "99"
        },
        {
          "CreateAt": 1691139338,
          "DepartmentId": "1",
          "DepartmentName": "研发部",
          "Email": "test2@tencent.com",
          "LinkFilter": 1,
          "Phone": "139****9999",
          "RealName": "test2",
          "Status": 1,
          "TenantId": "100055",
          "UserId": "baea0677027445f49ebcf0ed29480cc2",
          "UserType": "3"
        },
        {
          "CreateAt": 1686233544,
          "DepartmentId": "21",
          "DepartmentName": "交付组",
          "Email": "test3@tencent.com",
          "LinkFilter": 1,
          "Phone": "132****0000",
          "RealName": "test3",
          "Status": 1,
          "TenantId": "100055",
          "UserId": "159ac75273614918be42b3b0a208d1ab",
          "UserType": "3"
        }
      ]
    }
  }
}
```

```

}
]
}
}
}
}
    
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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
InternalServerError.AppApiNoSpacePermission	该应用未关联该项目空间数据，无法获取该项目空间数据
InternalServerError.UnknownError	未知错误
InvalidParameter	参数错误。

视频服务相关接口

云台控制

最近更新時間：2023-11-30 08:49:38

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

云台控制

默认接口請求頻率限制：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

以下請求參數列表仅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

參數名稱	必選	類型	描述
Action	是	String	公共參數 ，本接口取值：ControlCameraPTZ。
Version	是	String	公共參數 ，本接口取值：2023-04-27。
Region	是	String	公共參數 ，詳見產品支持的 地域列表 。
WID	是	String	設備的唯一標識 示例值：4de8f447-b4b6-4f6c-a6a2-55eeb7b0da03
CMD	是	String	ptz指令 left - 向左移動 right - 向右移動 up - 向上移動 down - 向下 zoomOut - 鏡頭縮小 zoomIn - 鏡頭放大 irisIn - 光圈縮小 irisOut - 光圈放大 focusIn - 焦距變近 focusOut - 焦距變遠 示例值：left
WorkspaceId	是	Integer	工作空間Id 示例值：1016
ApplicationToken	是	String	應用token 示例值：xxx

3. 输出参数

參數名稱	類型	描述
Result	EmptyRes	控制攝像頭結果返回 示例值：{"Msg":"ok"}
RequestId	String	唯一請求 ID，每次請求都會返回。定位問題時需要提供該次請求的 RequestId。

4. 示例

示例1 云台控制

成功响应

输入示例

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

{
  "WID": "c6cb1e55-7bd7-4f08-a97b-0b530f8a66f3",
  "CMD": "left",
  "Workspaceld": 1016,
  "ApplicationToken": "ct5E8QNPQEZzqNW7ShPAJQVzrTV9xIbL"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "745845ec-f5be-4e2f-87b6-8e46b9269d0a",
    "Result": {
      "Msg": "ok"
    }
  }
}
```

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
-----	----

错误码	描述
InternalError	内部错误。
MissingParameter.EmptyWID	WID参数缺失
UnsupportedOperation.UnsupportedCMD	不支持的控制指令

云录像接口

最近更新时间：2023-11-30 08:49:37

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

云录像接口

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：DescribeVideoCloudRecord。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WID	是	String	设备的唯一标识 示例值：c6cb1e55-7bd7-4f08-a97b-0b530f8a66f3
StartTime	是	Integer	录像开始时间（s） 示例值：1693466144
EndTime	是	Integer	录像结束时间（s） 示例值：1693470107
WorkspaceId	是	Integer	工作空间Id 示例值：1016
ApplicationToken	是	String	应用token 示例值：xxx

3. 输出参数

参数名称	类型	描述
Result	VideoCloudRecordRes	获取云录像结果 示例值：{"RecordSet":[],"TotalCount":0}
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 云录像

成功响应

输入示例

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

```
{
  "WID": "c0e3f641-6ec6-444e-8ac9-3cfec6755cb0",
  "StartTime": 1693466144,
  "EndTime": 1693470107,
  "WorkspaceId": 1016,
  "ApplicationToken": "RewYT6DXVWDg51Sx8tdcj89cX1pEns2h"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "59bb6656-68d4-4684-aa53-2039f5bd3069",
    "Result": {
      "RecordSet": [],
      "TotalCount": 0
    }
  }
}
```

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
InternalServerError	内部错误。
InvalidParameterValue.InvalidTimeRange	时间范围参数错误
MissingParameter.EmptyWID	WID参数缺失
ResourceNotFound.CameraBaseInfoNotExist	摄像头基础信息不存在

实时流接口

最近更新時間：2023-11-30 08:49:36

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

实时流接口

默认接口請求頻率限制：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

以下請求參數列表仅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

參數名稱	必選	類型	描述
Action	是	String	公共參數，本接口取值：DescribeVideoLiveStream。
Version	是	String	公共參數，本接口取值：2023-04-27。
Region	是	String	公共參數，詳見產品支持的 地域列表 。
WID	是	String	設備的唯一標識 示例值：c0e3f641-6ec6-444e-8ac9-3cfecc755cb0
Protocol	是	String	枚舉如下： flv rtmp hls webrtc raw (視頻原始幀) 示例值：hls
WorkspaceId	是	Integer	工作空間Id 示例值：1016
ApplicationToken	是	String	應用token 示例值：xxx
StreamId	否	Integer	主碼流傳0，子碼流傳1，默認主碼流 示例值：0
Env	否	String	設備所在環境，公有雲私有化項目傳0或者不傳，混合雲項目一般傳空間id 示例值：0

3. 输出参数

參數名稱	類型	描述
Result	VideoRecordStreamRes	視頻實時流獲取結果 示例值：{"FLV":"","HLS":"https://t-video.twins.tencent.com:443/hls1935/76d8109e-64cd-4a85-b01ticket=7a2368131f084363a4a6cb146cbfc353","RAW":{"IP":"","InnerIP":"","NATIP":"","Port":0,"SM4Vector":"","StreamEnKey":"","StreamToken":"","RTMF64cd-4a85-b011-4a0c29a654e8","WebRTC":""}}
RequestId	String	唯一請求 ID，每次請求都會返回。定位問題時需要提供該次請求的 RequestId。

4. 示例

示例1 获取实时流地址

成功响应

输入示例

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

{
  "WID": "76d8109e-64cd-4a85-b011-4a0c29a654e8",
  "Protocol": "hls",
  "Workspaceld": 1016,
  "ApplicationToken": "ct5E8QNPQEZzqNW7ShPAJQVzrTV9xIbL"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "96233e8f-3a98-470b-8bf2-45fbd4988bd8",
    "Result": {
      "FLV": "",
      "HLS": "https://t-video.twins.tencent.com:443/hls1935/xxx",
      "RAW": {
        "IP": "",
        "InnerIP": "",
        "NATIP": "",
        "Port": 0,
        "SM4Vector": "",
        "StreamEnKey": "",
        "StreamToken": ""
      },
      "RTMP": "",
      "Stream": "76d8109e-64cd-4a85-b011-4a0c29a654e8",
      "WebRTC": ""
    }
  }
}
```

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
FailedOperation.LockRedisCacheFailed	锁定Redis缓存错误
FailedOperation.SendMsgToIOTFailed	向IOT服务发送消息失败
InternalServerError	内部错误。
InvalidParameter.InvalidMetaData	错误的Meta数据，无法构建握手参数
InvalidParameterValue.InvalidProtocol	非法protocol协议参数
InvalidParameterValue.InvalidStreamId	非法StreamId，StreamId仅支持0或1
LimitExceeded.VideoStreamThresholdExceeded	视频流超过阈值
LimitExceeded.VideoTranscode	视频转码超出阈值
MissingParameter.EmptyWID	WID参数缺失
ResourceNotFound.CameraBaseInfoNotExist	摄像头基础信息不存在
ResourceNotFound.CameraStatusNotExist	摄像头状态不存在
ResourceNotFound.SRSHookServiceNode	srs_hook服务节点缺失
ResourceNotFound.SRSNotExist	该deviceId的srs信息缺失
ResourceNotFound.VideoPushServiceNode	VideoPush服务节点缺失
ResourceUnavailable.DeviceOffline	设备已离线

历史流接口

最近更新時間：2023-11-30 08:49:35

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

历史流接口

默认接口請求頻率限制：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

以下請求參數列表仅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

參數名稱	必選	類型	描述
Action	是	String	公共參數 ，本接口取值：DescribeVideoRecordStream。
Version	是	String	公共參數 ，本接口取值：2023-04-27。
Region	是	String	公共參數 ，詳見產品支持的 地域列表 。
WID	是	String	設備唯一標識 示例值：693bc0a4-ca2b-4f37-af04-4c8c0a88e000
Protocol	是	String	枚舉如下： flvsh rtmp hls webrtc raw (視頻原始幀) 示例值：hls
StartTime	是	Integer	開始時間 (精確到毫秒) 示例值：1699876800
EndTime	是	Integer	結束時間 (精確到毫秒) 示例值：1699876817
PlayBackRate	是	Float	倍速 0.5、1、2、4 示例值：1
WorkspaceId	是	Integer	工作空間id 示例值：1016
ApplicationToken	是	String	應用token 示例值：Lawbk967Ng41d934u8vveoSJgsj
Stream	否	String	流的唯一標識，播放鏈接尾綴 示例值：c6cb1e55-7bd7-4f08-a97b-0b530f8a60f1
Env	否	String	公有云私有化項目傳0或者不傳；混合云項目一般傳空間id 示例值：0

3. 輸出參數

參數名稱	類型	描述
RequestId	String	唯一請求 ID，每次請求都會返回。定位問題時需要提供該次請求的 RequestId。

4. 示例

示例1 历史流

成功响应

输入示例

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

{
  "WID": "c0e3f641-6ec6-444e-8ac9-3cfec6755cb0",
  "Protocol": "hls",
  "StartTime": 1693466137,
  "EndTime": 1693466144,
  "PlayBackRate": 1,
  "Workspaceld": 1016,
  "ApplicationToken": "wA9bELu14oulQCBK876UDBxe00dNcxQW"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "2beb0875-c859-4fde-a46e-5dbdf1421f14"
  }
}
```

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
FailedOperation.SendMsgToIOTFailed	向IOT服务发送消息失败
InternalServerError	内部错误。
InvalidParameter.InvalidMediaData	错误的媒体数据
InvalidParameterValue.InvalidGB28181Config	网络视频录像机GB28181协议错误
InvalidParameterValue.InvalidProtocol	非法protocol协议参数
InvalidParameterValue.InvalidSaveType	保存类型参数错误
InvalidParameterValue.InvalidTimeRange	时间范围参数错误
InvalidParameterValue.InvalidVideoPlayRate	播放速率错误，需要等于0.5、1、1.5、2、4、8或16
LimitExceeded.VideoStreamThresholdExceeded	视频流超过阈值
LimitExceeded.VideoTranscode	视频转码超出阈值
MissingParameter.EmptyWID	WID参数缺失
ResourceNotFound.CameraBaseInfoNotExist	摄像头基础信息不存在
ResourceNotFound.CameraInfoNotExist	摄像头信息不存在
ResourceNotFound.NVRorCVRConfigNotExist	网络视频录像机或中心级视频网络存储设备相关配置信息缺失
ResourceNotFound.SRSHookServiceNode	srs_hook服务节点缺失
ResourceNotFound.SRSNotExist	该deviceId的srs信息缺失
ResourceNotFound.StreamNotExist	视频流缺失
ResourceNotFound.VideoPushServiceNode	VideoPush服务节点缺失

断流接口

最近更新時間：2023-11-30 08:49:35

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

斷流接口

默認接口請求頻率限制：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

以下請求參數列表僅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

參數名稱	必選	類型	描述
Action	是	String	公共參數 ，本接口取值：StopVideoStreaming。
Version	是	String	公共參數 ，本接口取值：2023-04-27。
Region	是	String	公共參數 ，詳見產品支持的 地域列表 。
Stream	是	String	流的唯一標識，播放鏈接尾綴 示例值：c0e3f641-6ec6-444e-8ac9-3cfec755cb0
WID	是	String	設備的唯一標識 示例值：c0e3f641-6ec6-444e-8ac9-3cfec755cb0
WorkspaceId	是	Integer	工作空間Id 示例值：1016
ApplicationToken	是	String	應用token 示例值：xxxx

3. 输出参数

參數名稱	類型	描述
Result	EmptyRes	停流接口返回結果 示例值：{"Msg":"ok"}
RequestId	String	唯一請求 ID，每次請求都會返回。定位問題時需要提供該次請求的 RequestId。

4. 示例

示例1 斷流接口

成功響應

輸入示例

```
POST / HTTP/1.1
Host: weilingwith.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: StopVideoStreaming
<公共請求參數>
```

```
{
  "Stream": "63ca1c08-81b7-499c-a95e-70e6abed4118",
  "WID": "63ca1c08-81b7-499c-a95e-70e6abed4118",
  "WorkspaceId": 1016,
  "ApplicationToken": "RewYT6DXVWDg51Sx8tdcj89cX1pEns2h"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "abb1962d-cc04-4113-9302-95a21f6ba18d",
    "Result": {
      "Msg": "ok"
    }
  }
}
```

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
InternalServerError	内部错误。
MissingParameter.EmptyStream	stream参数缺失
MissingParameter.EmptyWID	WID参数缺失
ResourceNotFound.CameraBaseInfoNotExist	摄像头基础信息不存在
ResourceNotFound.CameraStatusNotExist	摄像头状态不存在
ResourceUnavailable.DeviceOffline	设备已离线

错误码	描述
UnsupportedOperation.NotStreaming	未推流

获取视频扩展信息

最近更新时间：2023-11-30 08:49:37

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

获取视频扩展信息

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：DescribeCameraExtendInfo。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WID	是	String	设备的唯一标识 示例值：4de8f447-b4b6-4f6c-a6a2-55eeb7b0da03
WorkspaceId	是	Integer	工作空间Id 示例值：1016
ApplicationToken	是	String	应用token 示例值：xxx

3. 输出参数

参数名称	类型	描述
Result	CameraExtendInfoRes	获取视频扩展信息结果 示例值：{"SaveType":"cosmtav","SaveDay":60,"LiveResolutioin":0,"HistoryResolution":0}
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 获取视频扩展信息

成功响应

输入示例

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

{
  "WID": "c0e3f641-6ec6-444e-8ac9-3cfec6755cb0",
  "WorkspaceId": 1016,
  "ApplicationToken": "ct5E8QNPQEZzqNW7ShPAJQVzrTV9xIbL"
}
```

输出示例

```

{
  "Response": {
    "RequestId": "825de5ec-46bd-4682-b4fc-84e807ce2eed",
    "Result": {
      "HistoryResolution": 0,
      "LiveResolution": 480,
      "SaveDay": 0,
      "SaveType": "nvr"
    }
  }
}
    
```

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
InternalServerError	内部错误。
MissingParameter.EmptyWID	WID参数缺失
ResourceNotFound.CameraInfoNotExist	摄像头信息不存在

物联相关接口

拉取设备的设备类型列表

最近更新時間：2023-12-05 02:49:31

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

拉取设备的设备类型列表

默认接口請求频率限制：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

以下請求参数列表仅列出了接口請求参数和部分公共参数，完整公共参数列表见 [公共請求参数](#)。

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeDeviceTypeList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
ApplicationToken	是	String	应用token 示例值：xxx
Flag	否	Integer	默认0只拉取设备列表关联的设备类型列表；1拉取空间下所有的设备类型列表 示例值：0

3. 输出参数

参数名称	类型	描述
Result	DeviceTypeSet	设备的设备类型列表 示例值：{"Set": [{"Code": "w0301006", "IsSubsystem": 0, "Name": "烟感报警器", "ParentCode": "0301", "ParentName": "火灾自动报警及联动系统"}]}
RequestId	String	唯一請求 ID，每次請求都会返回。定位问题时需要提供该次請求的 RequestId。

4. 示例

示例1 设备列表类型查询结果

成功响应

输入示例

```
POST / HTTP/1.1
Host: weilingwith.tencentcloudapi.com
Content-Type: application/json
X-TC-Action: DescribeDeviceTypeList
<公共請求参数>
```



```
{
  "WorkspaceId": 1016,
  "ApplicationToken": "YzenL5LdGoxQM5gqJfCCoMDeGqUSsY78"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "7be5626c-d9f9-4883-b4c4-3d77ba2fe281",
    "Result": {
      "Set": [
        {
          "Code": "w0301006",
          "IsSubsystem": 0,
          "Name": "烟感报警器",
          "ParentCode": "0301",
          "ParentName": "火灾自动报警及联动系统"
        }
      ]
    }
  }
}
```

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.TokenNotFound	未找到该token信息
InvalidParameter.TokenFieldNotFound	未找到token字段

错误码	描述
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id

产品列表查询

最近更新时间：2023-11-30 08:49:41

1. 接口描述

接口请求域名: weilingwith.tencentcloudapi.com。

产品列表查询

默认接口请求频率限制: 20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 , 本接口取值: DescribeProductList。
Version	是	String	公共参数 , 本接口取值: 2023-04-27。
Region	是	String	公共参数 , 详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值: 1016
PageNumber	是	Integer	分页查询, 第几页 示例值: 1
PageSize	是	Integer	每页条数, 大于0 示例值: 1
ApplicationToken	是	String	应用token 示例值: xxx
DeviceTypeSet.N	否	Array of String	设备类型 示例值: w020202
ProductIdSet.N	否	Array of Integer	产品 pid 示例值: 12
ModelIdSet.N	否	Array of String	模型id 示例值: 29

3. 输出参数

参数名称	类型	描述
Result	ProductSet	产品列表查询结果 示例值: {"PageNumber": 1,"PageSize": 1,"Product": [{"Attribute": 1,"DeviceTypeId": "w777","DeviceTypeName": "", "MaintenanceMfr": "未知厂商","Manufacturer": "未知厂商","ModelId": "", "ModelName": "", "ModelType": 0,"ProductAbility": 1,"ProductId": 2000051,"ProductName": "视频网关","ProductType": "33","WorkspaceId": 1016}], "TotalPage": 132,"TotalRow": 132}
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 产品列表查询

正常返回结果

输入示例

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

```
X-TC-Action: DescribeProductList
```

```
<公共请求参数>
```

```
{
  "WorkspaceId": 1016,
  "PageNumber": 1,
  "PageSize": 1,
  "ApplicationToken": "YzenL5LdGoxQM5gJfCCoMDeGqUSsY78"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "8372d7e4-f04f-43bd-8f7d-d8b4586fdc8b",
    "Result": {
      "PageNumber": 1,
      "PageSize": 1,
      "Product": [
        {
          "Attribute": 1,
          "DeviceTypeId": "w777",
          "DeviceTypeName": "",
          "MaintenanceMfr": "未知厂商",
          "Manufacturer": "未知厂商",
          "ModelId": "",
          "ModelName": "",
          "ModelType": 0,
          "ProductAbility": 1,
          "ProductId": 2000051,
          "ProductName": "视频网关",
          "ProductType": "33",
          "WorkspaceId": 1016
        }
      ],
      "TotalPage": 132,
      "TotalRow": 132
    }
  }
}
```

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.ApiAuthorizationNotFound	未找到此应用该api授权信息
InternalServerError.AppApiNoSpacePermission	该应用未关联该项目空间数据，无法获取该项目空间数据
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound	资源不存在。

模型列表查询

最近更新时间：2023-11-30 08:49:42

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

模型列表查询/单个查询（产品模型/标准模型）

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeModelList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
PageNumber	是	Integer	分页查询，第几页，大于0 示例值：1
PageSize	是	Integer	每页条数，大于0 示例值：20
ApplicationToken	是	String	应用token 示例值：xxx
DeviceTypeSet.N	否	Array of String	设备类型 示例值：w02020
ProductIdSet.N	否	Array of Integer	产品 pid 示例值：29
ModelIdSet.N	否	Array of String	模型id 示例值：0123

3. 输出参数

参数名称	类型	描述
Result	ModelSet	模型列表查询结果 示例值：{"PageNumber": 1, "PageSize": 1, "Set": [{"DeviceType": "w0404024", "DeviceTypeName": "空调温控器", "ModelId": "2000055", "ModelName": "边缘温控器", "ModelParams": "x-json:{", "ModelType": 1, "RelatedProduct": [{"Id": 2000055, "Name": "边缘温控器"}], "WorkspaceId": 1016}, {"TotalPage": 156, "TotalRow": 156}
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 物模型列表返回示例

物模型列表

输入示例

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

{
  "WorkspaceId": 1016,
  "PageNumber": 1,
  "PageSize": 1,
  "ApplicationToken": "YzenL5LdGoxQM5gqJfCCoMDeGqUSsY78"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "934fc559-1b2c-42b4-9bb7-b21dad005d90",
    "Result": {
      "PageNumber": 1,
      "PageSize": 1,
      "Set": [
        {
          "DeviceType": "w0404024",
          "DeviceTypeName": "空调温控器",
          "ModelId": "2000055",
          "ModelName": "边缘温控器",
          "ModelParams": "x-json:{"properties":[{"required":false,"name":"当前温度","id":"temperature","mode":"rw","desc":"","define":{"type":"double","min":-30,"max":55,"unit":"°C"},"order":1681354602280},{"required":false,"name":"设定温度","id":"setTemperature","mode":"rw","desc":"","define":{"type":"double","min":16,"max":35,"unit":"°C"},"order":1681354625728},{"required":false,"name":"运行模式","id":"mode","mode":"rw","desc":"","define":{"type":"enum","mapping":{"0":"关闭","1":"制冷","2":"制热","3":"送风"}},"order":168135464730},{"required":false,"name":"风速","id":"setWind","mode":"rw","desc":"","define":{"type":"enum","mapping":{"0":"自动","1":"1档","2":"2档","3":"3档","4":"4档","5":"5档"}},"order":1684390032099,"level":1,"key":"properties","parentId":"setWind","value":[],"type":"枚举型"},"services":[{"required":false,"name":"控制","id":"control","mode":"rw","desc":"","define":{"callType":"async","inputData":[{"required":false,"name":"设定温度","id":"setTemperature","desc":"","mode":"rw","define":{"type":"double"},"order":1684292612782},{"required":false,"name":"设定运行模式","id":"mode","desc":"","mode":"rw","define":{"type":"enum","mapping":{"0":"关闭","1":"制冷","2":"制热","3":"送风"}},"order":1684292898776},{"required":false,"name":"设定风速","id":"setWind","desc":"","mode":"rw","define":{"type":"enum","mapping":{"1":"1档","2":"2档","3":"3档","4":"4档","5":"5档"}},"order":1684292933365}],outputData":[{"required":false,"name":"是否成功","id":"success","desc":"","mode":"rw","define":{"type":"bool","mapping":{"0":"失败","1":"成功","false":"失败","true":"成功"}},"order":1684292612782},{"required":false,"name":"响应码","id":"code","desc":"","mode":"rw","define":{"type":"int"},"order":1684292665854},{"required":false,"name":"错误消息","id":"errmsg","desc":"","mode":"rw","define":{"type":"string","min":0,"max":999999},"order":1684292770509],"level":1,"key":"services","parentId":"control","value1":[{"required":false,"name":"设定温度","id":"setTemperature","desc":"","mode":"rw","define":{"type":"double"},"order":1682596793666,"type":"双精度浮点数","key":"services","parentId":"control","value":[],"level":2},"value2":[{"required":false,"name":"是否成功","id":"success","desc":"","mode":"rw","define":{"type":"bool","mapping":{"0":"失败","1":"成功","false":"失败","true":"成功"}},"order":1682596861463,"type":"布尔型","key":"services","parentId":"control","value":[],"level":2},"value":[{"required":false,"name":"设定温度","id":"setTemperature","desc":"","mode":"rw","define":{"type":"double"},"order":1682596793666,"type":"双精度浮点数","key":"services","parentId":"control","value":[],"level":2},{"required":false,"name":"是否成功","id":"success","desc":"","mode":"rw","define":{"type":"bool","mapping":{"0":"失败","1":"成功","false":"失败","true":"成功"}},"order":1682596861463,"type":"布尔型","key":"services","parentId":"control","value":[],"level":2},"events":[{"required":false,"name":"高温告警","id":"highTemperature","mode":"rw","desc":"","define":{"properties":[{"required":false,"name":"告警类型","id":"eventType","desc":"","mode":"rw","define":{"type":"int"},"order":1684378396844},{"required":false,"name":"告警时间","id":"eventTs","desc":"","mode":"rw","define":{"type":"long"},"order":1684378416811}],"required":false,"name":"描述","id":"describe","desc":"","mode":"rw","define":{"type"
```

```

        {\ "string",\ "min":0,\ "max":100},\ "order":1684378433361},{\ "required":false,\ "name":\ "告警等级",\ "id":\ "alarmLevel",\ "desc":\ "",\ "mode
        \":\ "rw",\ "define":{\ "type":\ "int"},\ "order":1684378459333}],\ "type":\ "notify"},{\ "required":false,\ "name":\ "低温告警",\ "id":\ "lowTemperatur
        e",\ "mode":\ "rw",\ "desc":\ "",\ "define":{\ "type":\ "int"},\ "order":1684381625535},{\ "required":false,\ "name":\ "告警类型",\ "id":\ "eventType",\ "desc":\ "",\ "mode":\ "rw
        \",\ "define":{\ "type":\ "int"},\ "order":1684381625535},{\ "required":false,\ "name":\ "告警时间",\ "id":\ "eventTs",\ "desc":\ "",\ "mode":\ "rw",\ "de
        fine":{\ "type":\ "long"},\ "order":1684381625535},{\ "required":false,\ "name":\ "描述",\ "id":\ "describe",\ "desc":\ "",\ "mode":\ "rw",\ "define":
        {\ "type":\ "string",\ "min":0,\ "max":100},\ "order":1684381625535},{\ "required":false,\ "name":\ "告警等级",\ "id":\ "alarmLevel",\ "desc":\ "",\ "m
        ode":\ "rw",\ "define":{\ "type":\ "int"},\ "order":1684381625535}],\ "type":\ "notify",\ "level":1,\ "key":\ "events",\ "parentId":\ "lowTemperature
        \",\ "value":{\ "required":false,\ "name":\ "告警类型",\ "id":\ "eventType",\ "desc":\ "",\ "mode":\ "rw",\ "define":{\ "type":\ "int"},\ "order":1684378
        396844,\ "type":\ "整数型",\ "key":\ "events",\ "parentId":\ "lowTemperature",\ "value":[\ "level":2],\ "required":false,\ "name":\ "告警时间",\ "id
        \":\ "eventTs",\ "desc":\ "",\ "mode":\ "rw",\ "define":{\ "type":\ "long"},\ "order":1684378416811,\ "type":\ "64位长整数",\ "key":\ "events",\ "parent
        Id":\ "lowTemperature",\ "value":[\ "level":2],\ "required":false,\ "name":\ "描述",\ "id":\ "describe",\ "desc":\ "",\ "mode":\ "rw",\ "define":{\ "typ
        e":\ "string",\ "min":0,\ "max":100},\ "order":1684378433361,\ "type":\ "字符串",\ "key":\ "events",\ "parentId":\ "lowTemperature",\ "value":[\ "le
        vel":2],\ "required":false,\ "name":\ "告警等级",\ "id":\ "alarmLevel",\ "desc":\ "",\ "mode":\ "rw",\ "define":{\ "type":\ "int",\ "min":\ "",\ "max
        \":\ ""},\ "order":1684378459333,\ "type":\ "整数型",\ "key":\ "events",\ "parentId":\ "lowTemperature",\ "value":[\ "level":2]}}",
        "ModelType": 1,
        "RelatedProduct": [
        {
        "Id": 2000055,
        "Name": "边缘温控器"
        }
        ],
        "WorkspaceId": 1016
        }
        ],
        "TotalPage": 156,
        "TotalRow": 156
        }
        }
        }
    
```

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.TokenExpired	token已过期,请重新申请
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id
ResourceNotFound	资源不存在。

设备列表查询

最近更新時間：2024-01-31 01:29:00

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

设备列表查询/单个查询（支持通过筛选条件查询，设备类型、标签、PID、空间）

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：DescribeDeviceList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
PageNumber	是	Integer	分页查询，第几页，必传，大于0 示例值：1
PageSize	是	Integer	每页条数，必传大于0 示例值：1
ApplicationToken	是	String	应用token 示例值：xxx
DeviceTypeSet.N	否	Array of String	设备类型，非必填 示例值：w0202
ProductIdSet.N	否	Array of Integer	产品 pid，非必填 示例值：29
TagIdSet.N	否	Array of Integer	设备标签，非必填 示例值：0
SpaceCodeSet.N	否	Array of String	空间层级 示例值：100100
DeviceTagSet.N	否	Array of String	设备标签名，非必填 示例值：中文标签测试
WIDSet.N	否	Array of String	设备wid,非必填 示例值：1c902558-d360-4a93-9b6f-1b49970534f7
Field	否	CustomField	自定义字段 示例值：{"Id": 67, "Val": "1"}
GroupIdSet.N	否	Array of Integer	分组id列表，非必填 示例值：[1,2,3]
IsActive	否	String	是否激活，默认全部，"1"激活，"0"未激活 示例值："1"

参数名称	必选	类型	描述
IsCamera	否	String	是否为摄像头，默认全部，"true"摄像头，"false"非摄像头 示例值: "false"

3. 输出参数

参数名称	类型	描述
Result	DescribeDeviceListRes	查询设备列表结果 示例值: {"DeviceDataSet": [{"ActiveTime": "2023-04-13T09:41:38+08:00", "DeviceName": "1c902558-d360-4a93-9b6f-1b49970534f7", "DeviceTagSet": ["中文标签测试"], "DeviceTypeCode": "1c902558-d360-4a93-9b6f-1b49970534f7", "DeviceTypeName": "1c902558-d360-4a93-9b6f-1b49970534f7", "FieldList": [{"Id": 1, "Key": "color", "Name": "颜色", "Val": ""}], "IsActive": 1, "IsLive": false, "Location": {"X": -197697.515625, "Y": 22156.90234375, "Z": 0}, "ModelId": "", "ModelName": "", "OtaVersion": "v1", "ParentWID": "", "ParentWIDName": "", "ProductAbility": 1, "ProductId": 2000053, "ProductName": "其它网关", "SN": "tony001", "SpaceInfoSet": [{"Code": "001234", "Id": "7e0eba7e-aa9b-43bd-8a68-2ea7153828de", "Level": 6, "Name": "验证专用"}], "WID": "1c902558-d360-4a93-9b6f-1b49970534f7"}, {"PageNumber": 1, "PageSize": 1, "TotalPage": 8790, "TotalRow": 8790}]
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 设备列表

成功响应

输入示例

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

{
  "WorkspaceId": 1016,
  "DeviceTypeSet": [
    "w0701005"
  ],
  "PageNumber": 1,
  "PageSize": 1,
  "ApplicationToken": "wA9bELuI4oulQCBK876UDBxe00dNcxQW"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "dd0b4025-2db7-4da3-998a-dda6cfc19441",
    "Result": {
      "DeviceDataSet": [
        {
          "ActiveTime": "2023-04-13T14:11:24+08:00",
          "DeviceName": "5f089c7e-0530-44c9-9435-c6aaf92f6a29",
          "DeviceTagSet": [
            "西安",
            "真实"
          ]
        }
      ]
    }
  }
}
```

```

],
"DeviceTypeCode": "5f089c7e-0530-44c9-9435-c6aaf92f6a29",
"DeviceTypeName": "5f089c7e-0530-44c9-9435-c6aaf92f6a29",
"FieldList": [
{
"Id": 1,
"Key": "color",
"Name": "颜色",
"Val": ""
},
{
"Id": 3,
"Key": "height",
"Name": "高度",
"Val": ""
},
{
"Id": 64,
"Key": "size",
"Name": "尺寸",
"Val": ""
},
{
"Id": 65,
"Key": "weight",
"Name": "重量",
"Val": ""
},
{
"Id": 66,
"Key": "width",
"Name": "宽度",
"Val": ""
},
{
"Id": 67,
"Key": "long",
"Name": "长度",
"Val": ""
},
{
"Id": 68,
"Key": "volume",
"Name": "体积",
"Val": ""
}
],
"IsActive": 1,
"IsLive": false,
"Location": {
"X": 30769.744140625,
"Y": 10080.6162109375,
"Z": 24750
},
"ModelId": "2000056",
"ModelName": "边缘摄像头",
"ParentWID": ""

```

```
"ParentWIDName": "",
"ProductAbility": 3,
"ProductId": 2000056,
"ProductName": "边缘摄像头",
"SN": "DS-2DE7530IW-A20190315AACHD00715442W",
"SpaceInfoSet": [
  {
    "Code": "000137",
    "Id": "85dc534a-4f69-4553-aad0-368cda0f1e92",
    "Level": 6,
    "Name": "滨海大厦"
  },
  {
    "Code": "000137048",
    "Id": "bf0fb1b4-493c-458f-bf8c-7279963b4928",
    "Level": 7,
    "Name": "7F"
  },
  {
    "Code": "00013704800003110",
    "Id": "472ee223-5caf-46c1-beca-d7238a937733",
    "Level": 9,
    "Name": ""
  }
],
"WID": "5f089c7e-0530-44c9-9435-c6aaf92f6a29"
},
"PageNumber": 1,
"PageSize": 1,
"TotalPage": 16,
"TotalRow": 16
}
}
}
```

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.ApiAuthorizationNotFound	未找到此应用该api授权信息
AuthFailure.TokenNotFound	未找到该token信息
FailedOperation.ApiRuleConfigError	Api规则配置信息错误
InternalServerError.ApiGatewayInternalError	API网关内部错误
InternalServerError.BusinessLogicError	业务逻辑错误
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id
ResourceNotFound.EmptyDeviceList	空设备列表
UnauthorizedOperation.UnauthorizedApi	应用api的数据授权未配置，请先配置权限

批量新增设备

最近更新时间：2023-11-30 08:49:54

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

单个/批量新增设备

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：BatchCreateDevice。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间Id 示例值：1016
AddDeviceSet.N	是	Array of AddDeviceInfo	设备信息项 示例值：{"ProductId": 200022,"ParentWID": "xxx-aaa","KeySource": 0,"SN": "test"}
ApplicationToken	是	String	应用token 示例值：xxx

3. 输出参数

参数名称	类型	描述
Result	BatchCreateDeviceRes	批量新增设备返回结果 示例值：{"FailSet": [], "SuccessSet": [{"ParentWID": "", "ProductId": 2000054, "SN": "atest", "WID": "75c83fbd-4862-4c4e-8091-18ccba3e2419"}]}
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 批量添加设备

成功响应

输入示例

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

{
```

```
"WorkspaceId": 0,
"AddDeviceSet": [
{
"ProductId": 0,
"ParentWID": "abc",
"KeySource": 0,
"SN": "abc"
}
],
"ApplicationToken": "abc"
}
```

输出示例

```
{
"Response": {
"RequestId": "27e1d3a2-5f8a-415b-9af4-41eb1751a68d",
"Result": {
"FailSet": [],
"SuccessSet": [
{
"ParentWID": "",
"ProductId": 2000054,
"SN": "atest2",
"WID": "279ac171-ae22-4d75-a421-25b6012049e4"
}
]
}
}
}
```

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
FailedOperation.DBError	操作DB失败
InternalServerError.ManagerSrvFailed	manager服务操作失败
InternalServerError.UnknownError	未知错误
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
LimitExceeded.DeviceLimitExceeded	设备数量超过限制
MissingParameter	缺少参数错误。

批量修改设备名字

最近更新时间：2023-11-30 08:49:40

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

批量修改设备名字

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：ModifyDeviceName。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
Set.N	是	Array of DeviceModifyInfo	设备修改信息集合 示例值：{"WID":"xx","DeviceName":"a"}
ApplicationToken	是	String	应用token 示例值：xxx

3. 输出参数

参数名称	类型	描述
Result	EmptyRes	返回请求结果 示例值：{"Msg":"ok"}
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 修改设备名字

成功响应

输入示例

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

{
  "WorkspaceId": 1016,
  "Set": [
```

```
{
  "WID": "be2ea997-b7f4-4167-9545-9aa275158a92",
  "DeviceName": "a"
}
],
"ApplicationToken": "YVySZSL1Lp5UtSJ5ujVLJYOKDEGfCCH2"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "7ea50911-8e3d-40aa-82ee-c00d23cc044d",
    "Result": {
      "Msg": "ok"
    }
  }
}
```

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
FailedOperation.ModifyDeviceError	修改设备信息失败
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound.EmptyDeviceList	空设备列表
ResourceNotFound.WIDNotExist	设备Wid不存在

获取设备影子数据

最近更新时间：2023-11-30 08:49:47

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

获取设备影子数据

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：DescribeDeviceShadowList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
WIDSet.N	是	Array of String	WID 示例值：8eaebd6f-d780-424e-98d6-01de5c78e25f
PageNumber	是	Integer	分页查询，第几页 示例值：1
PageSize	是	Integer	每页条数 示例值：20
ApplicationToken	是	String	应用token 示例值：Lawbk967Ng41d934u8vveoS
DeviceTypeSet.N	否	Array of String	设备类型code 示例值：w20011
ProductIdSet.N	否	Array of Integer	产品 pid 示例值：2000252
TagIdSet.N	否	Array of Integer	设备标签id 示例值：45
SpaceCodeSet.N	否	Array of String	空间层级，（支持空间多层，比如具体建筑、具体楼层） 示例值：000381
DeviceTagSet.N	否	Array of String	设备标签名 示例值：test

3. 输出参数

参数名称	类型	描述
Result	DeviceShadowRes	获取设备影子结果 示例值：{"PageNumber": 1, "PageSize": 20, "Set": [{"DeviceShadow": "x-json":{"desired": {}, "reported": {"events": {}, "profile": {"modelId": "w0745001"}, "properties": {"Ept": 96, "IA": 32, "IB": 42}, "reportTs": 1693370542126}}, {"DeviceShadowUpdateTime": "2023-08-30 12:42:22", "WID": "026d1dd8-a6d7-4fb7-9a94-0df07edaaff8"}], "TotalPage": 1, "TotalRow": 1}

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 设备影子查询

正常响应

输入示例

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

{
  "Workspaceld": 1016,
  "WIDSet": [
    "026d1dd8-a6d7-4fb7-9a94-0df07edaaff8"
  ],
  "PageNumber": 1,
  "PageSize": 20,
  "ApplicationToken": "YzenL5LdGoxQM5gqJfCCoMDeGqUSsY78"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "16528537-3858-42bd-973a-ae8c80bdeee3",
    "Result": {
      "PageNumber": 1,
      "PageSize": 20,
      "Set": [
        {
          "DeviceShadow": "x-json:{"desired":{"},"reported":{"events":{"},"profile":{"modelld":"w0745001"},"properties":{"Ept":"96","IA":"32","IB":"42"},"reportTs":"1693370542126}}",
          "DeviceShadowUpdateTime": "2023-08-30 12:42:22",
          "WID": "026d1dd8-a6d7-4fb7-9a94-0df07edaaff8"
        }
      ],
      "TotalPage": 1,
      "TotalRow": 1
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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
FailedOperation.ApiRuleConfigError	Api规则配置信息错误
FailedOperation.RedisOperationFailed	redis操作错误
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound.EmptyDeviceList	空设备列表
UnauthorizedOperation.UnauthorizedApi	应用api的数据授权未配置，请先配置权限

设备状态获取

最近更新時間：2023-11-30 08:49:46

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

设备状态获取

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：DescribeDeviceStatusList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
ApplicationToken	是	String	应用token 示例值：xxx
PageNumber	是	Integer	分页查询，第几页，必传，大于0 示例值：1
PageSize	是	Integer	每页条数，必传大于0 示例值：1
DeviceTypeSet.N	否	Array of String	设备类型 示例值：w010101
ProductIdSet.N	否	Array of Integer	产品 pid 示例值：29
TagIdSet.N	否	Array of Integer	设备标签id 示例值：0
SpaceCodeSet.N	否	Array of String	空间位置（支持空间多层，比如具体建筑、具体楼层） 示例值：100110
WIDSet.N	否	Array of String	设备编号列表 示例值：be2ea997-b7f4-4167-9545-9aa275158a92
DeviceTagSet.N	否	Array of String	设备标签名，非必填 示例值：test
DeviceStatusSet.N	否	Array of String	通信在/离线状态（online=normal+fault、offline） 示例值：online
StatusSet.N	否	Array of String	设备业务状态 （正常-normal、故障-fault、离线-offline） 示例值：normal

参数名称	必选	类型	描述
IsAlive.N	否	Array of Boolean	推流状态，推流中=true，未推流=false 仅摄像头有的状态 示例值：false

3. 输出参数

参数名称	类型	描述
Result	DeviceStatusRes	查询设备状态结果 示例值：{"DeviceStatusSet": [{"DeviceStatus": "online", "DeviceStatusUpdateTime": "2023-08-31 00:57:28", "IsAlive": true, "Status": "normal", "WID": "be2ea997-b7f4-4167-9545-9aa275158a92"}], "PageNumber": 1, "PageSize": 1, "TotalPage": 1, "TotalRow": 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 设备状态查询

正常响应

输入示例

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

{
  "WorkspaceId": 1016,
  "ApplicationToken": "YVySZSL1Lp5UtSJ5uJVLJYOKDEGFCH2",
  "WIDSet": [
    "be2ea997-b7f4-4167-9545-9aa275158a92"
  ],
  "PageNumber": 1,
  "PageSize": 1
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1286216c-1022-4bd6-83a2-3dfb0bee6fe2",
    "Result": {
      "DeviceStatusSet": [
        {
          "DeviceStatus": "online",
          "DeviceStatusUpdateTime": "2023-08-31 00:57:28",
          "IsAlive": true,
          "Status": "normal",
          "WID": "be2ea997-b7f4-4167-9545-9aa275158a92"
        }
      ],
      "PageNumber": 1,
      "PageSize": 1,
      "TotalPage": 1,

```



```
"TotalRow": 1
}
}
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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
FailedOperation.ApiRuleConfigError	Api规则配置信息错误
FailedOperation.RedisOperationFailed	redis操作错误
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound.EmptyDeviceList	空设备列表
UnauthorizedOperation.UnauthorizedApi	应用api的数据授权未配置，请先配置权限

设备状态统计

最近更新时间：2023-11-30 08:49:45

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

设备状态统计

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：DescribeDeviceStatusStat。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
Level	是	Integer	所属空间地理层级，必填。0表示查询所有层级（1、2）的设备状态，1表示楼栋，2表示楼层 示例值：0
WorkspaceId	是	Integer	工作空间id 示例值：1016
ApplicationToken	是	String	应用token 示例值：xxx
SpaceCodeSet.N	否	Array of String	空间位置，非必填。为空表示查询所有（1，2）层级 示例值：100110
DeviceTypeSet.N	否	Array of String	设备类型，非必填。为空表示查询所有设备类型 示例值：w02023

3. 输出参数

参数名称	类型	描述
Result	DeviceStatusStatRes	设备状态统计结果 示例值：{"DeviceTypeOverviewSet":[{"DeviceType":"w0715010","Fault":0,"Name":"多功能网关","Normal":6,"Offline":8212,"Total":8218}],"FaultSum":15,"NormalSum":87,"OfflineSum":8689,"StatLevelSet":[],"Total":8791,"WorkspaceId":1016}
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 设备状态统计列表

正常响应

输入示例

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

```
X-TC-Action: DescribeDeviceStatusStat
```

```
<公共请求参数>
```

```
{
  "Level": 1,
  "WorkspaceId": 1016,
  "ApplicationToken": "YVySZSL1Lp5UtSJ5uJVLJYOKDEGfCCH2"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "503f6887-d9af-48b9-91cb-999644f45e63",
    "Result": {
      "DeviceTypeOverviewSet": [
        {
          "DeviceType": "w0715010",
          "Fault": 0,
          "Name": "多功能网关",
          "Normal": 6,
          "Offline": 8212,
          "Total": 8218
        }
      ],
      "FaultSum": 15,
      "NormalSum": 87,
      "OfflineSum": 8689,
      "StatLevelSet": [],
      "Total": 8791,
      "WorkspaceId": 1016
    }
  }
}
```

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
FailedOperation.RedisOperationFailed	redis操作错误
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id

设备控制（单个、批量控制）

最近更新时间：2024-01-17 01:27:57

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

设备控制（单个、批量控制）

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：ControlDevice。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
WIDSet.N	是	Array of String	设备wid，最大100个 示例值：70e673c7-11b8-4ca0-b11c-dfa360a44638
ControlData	是	String	控制内容 示例值：left
ApplicationToken	是	String	应用token 示例值：xxx
IsSynchronized	否	Boolean	是否同步返回设备下控ack结果 示例值：false

3. 输出参数

参数名称	类型	描述
Result	ControlDeviceSet	设备控制后结果集 示例值：{"Set": [{"Code": 0, "Result": "success", "Seq": "21909094-b304-4942-af6a-eb781ddd0a2a", "WID": "70e673c7-11b8-4ca0-b11c-dfa360a44638"}]}
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 设备控制

成功响应

输入示例

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

```
X-TC-Action: ControlDevice
```

```
<公共请求参数>
```

```
{
  "WorkspaceId": 1016,
  "WIDSet": [
    "70e673c7-11b8-4ca0-b11c-dfa360a44638"
  ],
  "ControlData": "left",
  "ApplicationToken": "YzenL5LdGoxQM5gqJfCCoMDeGqUSsY78"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "21909094-b304-4942-af6a-eb781ddd0a2a",
    "Result": {
      "Set": [
        {
          "Code": 0,
          "Result": "success",
          "Seq": "21909094-b304-4942-af6a-eb781ddd0a2a",
          "WID": "70e673c7-11b8-4ca0-b11c-dfa360a44638"
        }
      ]
    }
  }
}
```

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
FailedOperation.ApiRuleConfigError	Api规则配置信息错误
FailedOperation.SendMsgError	消息发送错误
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id
ResourceNotFound.EmptyDeviceList	空设备列表
UnauthorizedOperation.APIAuthFailed	wId 未找到或被应用api未被授权
UnauthorizedOperation.UnauthorizedApi	应用api的数据授权未配置，请先配置权限

标签列表查询

最近更新时间：2023-11-30 08:49:45

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

标签列表查询

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

</> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeDeviceTagList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
PageNumber	是	Integer	分页查询，第几页，必传，大于0 示例值：1
PageSize	是	Integer	每页条数，必传大于0 示例值：1
ApplicationToken	是	String	应用token 示例值：xxx

3. 输出参数

参数名称	类型	描述
Result	DeviceTagRes	设备标签查询结果 示例值：{"PageNumber": 1, "PageSize": 1, "Set": [{"TagId": 35, "TagName": "西安真实摄像头"}], "TotalPage": 43, "TotalRow": 43}
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 设备标签列表查询

正常响应

输入示例

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



```
{
  "WorkspaceId": 1016,
  "PageNumber": 1,
  "PageSize": 1,
  "ApplicationToken": "YVySZSL1Lp5UtSJ5uJVLJYOKDEGFCH2"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "e812b515-ad1d-496a-9042-4494c4c8128e",
    "Result": {
      "PageNumber": 1,
      "PageSize": 1,
      "Set": [
        {
          "TagId": 35,
          "TagName": "西安真实摄像头"
        }
      ],
      "TotalPage": 43,
      "TotalRow": 43
    }
  }
}
```

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
-----	----

错误码	描述
FailedOperation.GetTagsFailed	获取标签错误
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id

告警类型获取

最近更新时间：2023-11-30 08:49:48

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

告警类型获取

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：DescribeAlarmTypeList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
ApplicationToken	是	String	应用token 示例值：xxx
ParentType	否	String	一级类型 示例值：test

3. 输出参数

参数名称	类型	描述
Result	DescribeAlarmTypeListRes	告警类型列表查询 示例值：{"AlarmTypeSet": [{"EnglishName": "excessive", "Id": 26, "Name": "人流数量异常", "ParentId": 25, "Type": 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 告警状态列表

成功响应

输入示例

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

{
  "WorkspaceId": 1016,
  "ApplicationToken": "YVySZSL1Lp5UtSJ5uJVLJYOKDEGFcCH2"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "3840c20f-f364-4569-8120-1c29f072dd24",
    "Result": {
      "AlarmTypeSet": [
        {
          "EnglishName": "excessive",
          "Id": 26,
          "Name": "人流数量异常",
          "ParentId": 25,
          "Type": 1
        }
      ]
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound	资源不存在。

告警级别枚举获取

最近更新时间：2023-11-30 08:49:50

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

告警级别枚举获取

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：DescribeAlarmLevelList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
ApplicationToken	是	String	应用token 示例值：xxx

3. 输出参数

参数名称	类型	描述
Result	DescribeAlarmLevelListRes	告警级别列表查询结果 示例值：{"AlarmLevelSet": [{"LevelId": 3, "LevelName": "严重"}, {"LevelId": 4, "LevelName": "紧急"}, {"LevelId": 5, "LevelName": "致命"}, {"LevelId": 1, "LevelName": "提示"}, {"LevelId": 2, "LevelName": "一般"}]}
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 告警级别列表

成功响应

输入示例

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

{
  "WorkspaceId": 1016,
  "ApplicationToken": "baSTzPx0vZ6LPuv2EifNa5CqRBJ9hoY0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "54b9f3e5-9e78-43aa-92b6-d895d31372d6",
    "Result": {
      "AlarmLevelSet": [
        {
          "LevelId": 3,
          "LevelName": "严重"
        },
        {
          "LevelId": 4,
          "LevelName": "紧急"
        },
        {
          "LevelId": 5,
          "LevelName": "致命"
        },
        {
          "LevelId": 1,
          "LevelName": "提示"
        },
        {
          "LevelId": 2,
          "LevelName": "一般"
        }
      ]
    }
  }
}
```

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
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id
ResourceNotFound	资源不存在。

告警列表查询

最近更新时间：2023-11-30 08:49:49

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

告警列表查询

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeAlarmList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
BeginTime	是	Integer	告警开始时间，必填，时间戳秒 示例值：1684464518
EndTime	是	Integer	告警结束时间，必填，时间戳秒 示例值：1684723718
PageNumber	是	Integer	分页查询，第几页 示例值：1
PageSize	是	Integer	每页条数 示例值：1
WorkspaceId	是	Integer	工作空间id 示例值：1016
ApplicationToken	是	String	应用token 示例值：xxx
Statuses.N	否	Array of String	告警状态 示例值：processed
AlarmTypeSet.N	否	Array of AlarmTypeInfo	告警类型 示例值：{"Type": "test1", "SubType": "test2" }
LevelSet.N	否	Array of Integer	告警级别id 示例值：1
IdSet.N	否	Array of String	告警id 示例值：atest
AppIdSet.N	否	Array of Integer	应用id 示例值：0
WIDSet.N	否	Array of String	设备id 示例值：xxx-aaa
SpaceCodeSet.N	否	Array of String	空间层级 示例值：10010

参数名称	必选	类型	描述
ExtendOne.N	否	Array of String	应用扩展字段1 示例值: {"a":"t"}
ExtendTwo.N	否	Array of String	应用扩展字段2 示例值: x-json:{"a":"t"}
ProcessorSet.N	否	Array of String	当前告警处理人, 填写生中台的用户id 示例值: 13
GroupIdSet.N	否	Array of Integer	分组id 示例值: [1,2,3]

3. 输出参数

参数名称	类型	描述
Result	DescribeAlarmListRes	告警列表查询结果 示例值: {"AlarmInfoSet": [{"Appld": 0, "Desc": "水量使用超限", "DeviceName": "20楼水表", "Echo": "", "Extend": "x-json:{"alarmLevel":3,"alarmValue":13730945.6}", "ExtendOne": "x-json:", "ExtendTwo": "x-json:", "HandlePersonSet": [], "HandleRecordSet": [], "Id": "0c6ec1cc-25d6-43bc-96b1-6b4131f1771a", "Level": 3, "LevelName": "严重", "Position": "滨海大厦-7F", "ReportImg": {"Data": "", "Type": 0}, "Status": "unprocessed", "SubType": "water_limit", "SubTypeName": "水量使用超限", "Time": 1693416976, "Type": "energy", "TypeName": "能源", "WID": "b1fd7aa3-4407-4b82-9dc3-aff9b3034a9f", "WorkspaceId": 1016}], "PageNumber": 1, "PageSize": 1, "TotalPage": 125, "TotalRow": 125}
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 告警列表

成功响应

输入示例

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

{
  "BeginTime": 1693416387,
  "EndTime": 1693416987,
  "PageNumber": 1,
  "PageSize": 1,
  "WorkspaceId": 1016,
  "ApplicationToken": "YVySZSL1Lp5UtSJ5uJVlJYOKDEGfCCH2"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "bb4a6274-71b1-4107-98a2-06819376fb4e",
    "Result": {
      "AlarmInfoSet": [
        {
```

```
"AppId": 0,
"Desc": "水量使用超限",
"DeviceName": "20楼水表",
"Echo": "",
"Extend": "x-json:{\"alarmLevel\":3,\"alarmValue\":13730945.6}",
"ExtendOne": "x-json:",
"ExtendTwo": "x-json:",
"HandlePersonSet": [],
"HandleRecordSet": [],
"Id": "0c6ec1cc-25d6-43bc-96b1-6b4131f1771a",
"Level": 3,
"LevelName": "严重",
"Position": "滨海大厦-7F",
"ReportImg": {
  "Data": "",
  "Type": 0
},
"Status": "unprocessed",
"SubType": "water_limit",
"SubTypeName": "水量使用超限",
"Time": 1693416976,
"Type": "energy",
"TypeName": "能源",
"WID": "b1fd7aa3-4407-4b82-9dc3-aff9b3034a9f",
"WorkspaceId": 1016
}
],
"PageNumber": 1,
"PageSize": 1,
"TotalPage": 125,
"TotalRow": 125
}
}
}
```

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
FailedOperation.ApiRuleConfigError	Api规则配置信息错误
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id
ResourceNotFound	资源不存在。
UnauthorizedOperation.UnauthorizedApi	应用api的数据授权未配置，请先配置权限

添加告警处理记录

最近更新时间：2023-11-30 08:49:55

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

添加告警处理记录

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：AddAlarmProcessRecord。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
RecordSet.N	是	Array of ProcessRecordInfo	处理记录项 示例值：{"Id": "cdf85c7c-cf04-41b5-9631-fa4a7290e615", "ProcessTime": 1688114263000, "ProcessType": "add_record-other", "ProcessDescription": "test", "AppendixFileId": "[{"fileId": "b87ae7ec-6874-4103-bfa0-2489374476eb", "fileName": "ecs.png"}]"}
WorkspaceId	是	Integer	工作空间id 示例值：1016
ApplicationToken	是	String	应用token 示例值：xxx
ApplicationId	否	Integer	非孪生平台外部应用appid 示例值：10051
ExtendOne	否	String	此字段填写的是非孪生中台的用户id（多个用逗号分隔），如果是非孪生中台用户必填该字段 示例值：cda323aa-da67-451f-8d27-71cb6748b0bf

3. 输出参数

参数名称	类型	描述
Result	EmptyRes	添加告警处理记录结果 示例值：{"Msg": "ok"}
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 添加告警处理记录

成功响应

输入示例

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

{
  "RecordSet": [
    {
      "Id": "7fd854c2-e9c4-4347-b0a6-3bf5a80ac3f6",
      "Processor": "12",
      "ProcessTime": 1693417958,
      "ProcessType": "add_record"
    }
  ],
  "WorkspaceId": 1016,
  "ApplicationToken": "baSTzPx0vZ6LPuv2EifNa5CqRBJ9hoY0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1f4ae18a-6a8e-477f-9335-d59546768e93",
    "Result": {
      "Msg": "ok"
    }
  }
}
```

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
FailedOperation.ApiRuleConfigError	Api规则配置信息错误
InvalidParameter	参数错误。
InvalidParameterValue.InvalidTenantId	metadata租户信息无效
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id
LimitExceeded.ApiLimitExceeded	api接口批量最大个数超限
OperationDenied.ApiPermissionDenied	Api没有操作权限
ResourceNotFound	资源不存在。
ResourceNotFound.AlarmIdNotExist	告警id不存在
UnauthorizedOperation.UnauthorizedApi	应用api的数据授权未配置，请先配置权限

变更告警状态

最近更新時間：2023-11-30 08:49:52

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

變更告警狀態

默認接口請求頻率限制：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

以下請求參數列表僅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

參數名稱	必選	類型	描述
Action	是	String	公共參數，本接口取值：ChangeAlarmStatus。
Version	是	String	公共參數，本接口取值：2023-04-27。
Region	是	String	公共參數，詳見產品支持的 地域列表 。
Id	是	String	告警的id，返回的列表中的id 示例值：40b52dda-c596-4510-84c4-a5e1aa6ae359
Status	是	String	告警狀態 processed unprocessed processing misreport shield 示例值：processed
ProcessTime	是	Integer	告警處理時間 示例值：1683343450674
ProcessType	是	String	處理類型 示例值：kill_warning-other
WorkspaceId	是	Integer	工作空間id 示例值：1016
UserId	是	String	當前操作用戶id 示例值：1
UserName	是	String	當前操作用戶名稱 示例值：atest
ApplicationToken	是	String	應用token 示例值：xxx
Processor	否	String	平台告警處理人 示例值：a
ProcessDescription	否	String	告警處理的描述信息 示例值：描述
ProcessExtend	否	String	告警處理的擴展信息，可以為JSON字符串 示例值：aaa
ExtendOne	否	String	擴展字段1，目前存的应用告警處理人 示例值：extend
ApplicationId	否	Integer	應用id 示例值：123

3. 输出参数

参数名称	类型	描述
Result	EmptyRes	返回空结果 示例值: {"Msg":"ok"}
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 变更告警状态

成功响应

输入示例

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

{
  "Id": "40b52dda-c596-4510-84c4-a5e1aa6ae359",
  "Status": "processed",
  "ProcessTime": 1693416387,
  "ProcessType": "kill_warning",
  "WorkspaceId": 1016,
  "UserId": "1",
  "UserName": "a",
  "ApplicationToken": "YVySZSL1Lp5UtSJ5ujVLJYOKDEGfCCH2"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "9544b470-8b37-4aeb-b7ea-d6cb0175ada1",
    "Result": {
      "Msg": "ok"
    }
  }
}
```

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
FailedOperation.ApiRuleConfigError	Api规则配置信息错误
InvalidParameter	参数错误。
InvalidParameter.InvalidAlarmContent	告警content无效
InvalidParameter.StatusNotMatchProcessType	状态与处理类型不符
InvalidParameterValue.InvalidAlarmStatus	告警状态无效
InvalidParameterValue.InvalidTenantId	metadata租户信息无效
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
MissingParameter.EmptyAlarmId	告警id为空
OperationDenied.ApiPermissionDenied	Api没有操作权限
ResourceNotFound	资源不存在。
ResourceNotFound.AlarmIDNotFound	告警id找不到
UnauthorizedOperation.UnauthorizedApi	应用api的数据授权未配置，请先配置权限

批量消警

最近更新時間：2023-11-30 08:49:53

1. 接口描述

接口請求域名：weilingwith.tencentcloudapi.com。

批量消警

默認接口請求頻率限制：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

以下請求參數列表僅列出了接口請求參數和部分公共參數，完整公共參數列表見 [公共請求參數](#)。

參數名稱	必選	類型	描述
Action	是	String	公共參數 ，本接口取值：BatchKillAlarm。
Version	是	String	公共參數 ，本接口取值：2023-04-27。
Region	是	String	公共參數 ，詳見產品支持的 地域列表 。
BeginTime	是	Integer	告警開始時間，必填，時間戳秒 示例值：1693416987
EndTime	是	Integer	告警結束時間，必填，時間戳秒 示例值：1693417958
StatusSet.N	是	Array of String	告警狀態：unprocessed processing 示例值：unprocessed
WorkspaceId	是	Integer	工作空間id 示例值：1016
UserId	是	String	當前操作用戶id 示例值：12
UserName	是	String	當前操作用戶名稱 示例值：a
ApplicationToken	是	String	應用token 示例值：xxx
ProcessorId	否	String	當前告警處理人，填孛生中台用戶id，多個用逗號分隔 示例值：12
AlarmTypeSet.N	否	Array of AlarmTypeInfo	告警子類型(如果傳告警子類型，則必傳父類型) 示例值：{"Type":"","SubType":""}
LevelSet.N	否	Array of Integer	告警級別，包括1-5 示例值：3
WIDSet.N	否	Array of String	設備id 示例值：wueh-xxx
IdSet.N	否	Array of String	告警id 示例值：dfac0463-b6a7-4df6-b825-c5414e17a419

参数名称	必选	类型	描述
Desc	否	String	告警处理的说明 示例值：消警

3. 输出参数

参数名称	类型	描述
Result	EmptyRes	批量消警结果 示例值：{"Msg":"ok"}
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 批量消警

成功响应

输入示例

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

{
  "BeginTime": 1693416987,
  "EndTime": 1693417958,
  "StatusSet": [
    "processed"
  ],
  "WorkspaceId": 1016,
  "UserId": "1",
  "UserName": "a",
  "ApplicationToken": "baSTzPx0vZ6LPuv2EifNa5CqRBj9hoY0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "b834d00c-0a1a-4e86-8765-f7114e905466",
    "Result": {
      "Msg": "ok"
    }
  }
}
```

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
FailedOperation.ApiRuleConfigError	Api规则配置信息错误
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
OperationDenied.ApiPermissionDenied	Api没有操作权限
ResourceNotFound	资源不存在。
UnauthorizedOperation.UnauthorizedApi	应用api的数据授权未配置，请先配置权限

联动规则列表查询

最近更新时间：2023-11-30 08:49:42

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

联动规则列表查询

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeLinkRuleList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
PageNumber	是	Integer	分页查询，第几页，必传，大于0 示例值：1
PageSize	是	Integer	每页条数，必传大于0 示例值：1
ApplicationToken	是	String	应用token 示例值：xxx
TriggerType	否	String	事件触发类型 示例值：timer
IdSet.N	否	Array of Integer	联动id 示例值：112

3. 输出参数

参数名称	类型	描述
Result	DescribeLinkRuleListRes	联动规则列表查询结果 示例值：{"LinkRuleSet": [{"ActionSet": [{"Id": 28, "Name": "瞿杨动作"}], "BeginDate": "2023-05-03", "EndDate": "2023-05-22", "EventSet": [{"Id": 33, "Name": "体验"}, {"Id": 24, "Name": "体验", "Status": 24, "ValidPeriod": ""}], "PageNumber": 1, "PageSize": 1, "TotalPage": 7, "TotalRow": 7}
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 联动规则列表

成功响应

输入示例

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

```
{
  "WorkspaceId": 1016,
  "TriggerType": "timer",
  "PageNumber": 1,
  "PageSize": 1,
  "ApplicationToken": "baSTzPx0vZ6LPuv2EifNa5CqRBj9hoY0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "aa1ac6c0-5721-4a84-bf87-dbd285bac6b7",
    "Result": {
      "LinkRuleSet": [
        {
          "ActionSet": [
            {
              "Id": 59,
              "Name": "体验123"
            },
            {
              "Id": 28,
              "Name": "瞿扬动作"
            }
          ],
          "BeginDate": "2023-05-03",
          "EndDate": "2023-05-22",
          "EventSet": [
            {
              "Id": 34,
              "Name": "联动新增事件"
            },
            {
              "Id": 33,
              "Name": "体验"
            },
            {
              "Id": 24,
              "Name": "体验",
              "Status": 24,
              "ValidPeriod": ""
            }
          ],
          "PageNumber": 1,
          "PageSize": 1,
          "TotalPage": 7,
          "TotalRow": 7
        }
      ]
    }
  }
}
```

```
}
}
```

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
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound	资源不存在。

联动规则详情查询

最近更新时间：2023-11-30 08:49:40

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

联动规则详情查询

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeRuleDetail。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
WorkspaceId	是	String	工作空间id 示例值：1016
Id	是	String	联动id 示例值：29
ApplicationToken	是	String	应用token 示例值：xxx

3. 输出参数

参数名称	类型	描述
Result	RuleDetailRes	规则详情查询结果 示例值：{"ActionInfoSet": [{"ApplyDevice": "", "CreateTime": "2023-05-23 14:38:13", "Desc": "事件范围内同一建筑", "json": "", "MsgType": "origin", "Name": "事件范围内同一建筑", "SinkConfig": "x-json:{"nearbySearch": {"deviceType": "w0201001", "pid": 0, "limit": 2, "objectLevel": 9, "scopeLevel": 6, "scopeRadius": 100, "deviceAssi": "nearbyDevices"}}, "BeginDate": "", "EndDate": "", "EventInfoSet": [{"Condition": "x-json: [{"type": "objmodel", "attr": "properties.idtest", "oper": "=", "val": "348", "obj": {"id": "idtest", "name": "idtest", "type": "ir直连类型", "modelId": "2000085", "define": {"type": "int"}}, "valType": 0, "otherObj": ""}], "Id": 36, "Name": "具体设备", "T": "device"}], "EventRule": "36", "RuleDesc": "联动同一建筑", "RuleId": 29, "RuleName": "联动同一建筑332", "Status": 0, "json": "", "ValidType": 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 联动详情

成功响应

输入示例

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


X-TC-Action: DescribeRuleDetail

<公共请求参数>

```
{
  "WorkspaceId": "1016",
  "Id": "29",
  "ApplicationToken": "baSTzPx0vZ6LPuv2EifNa5CqRBJ9hoY0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "6e77c453-636c-43db-97d8-aa7fac74f37b",
    "Result": {
      "ActionInfoSet": [
        {
          "ApplyDevice": "",
          "CreateTime": "2023-05-23 14:38:13",
          "Desc": "事件范围内同一建筑",
          "Id": 71,
          "MsgContent": "x-json:",
          "MsgType": "origin",
          "Name": "事件范围内同一建筑",
          "SinkConfig": "x-json:{\"nearbySearch\":{\"deviceType\":\"w0201001\",\"pid\":0,\"limit\":2,\"objectLevel\":9,\"scopeLevel\":\"6\",\"scopeRadius\":\"100\",\"deviceAssign\":{\"deviceType\"}}}",
          "Type": "nearbyDevices"
        }
      ],
      "BeginDate": "",
      "EndDate": "",
      "EventInfoSet": [
        {
          "Condition": "x-json:[{\"type\":\"objmodel\",\"attr\":{\"properties.idtest\",\"oper\":\"=\",\"val\":\"348\",\"obj\":{\"id\":\"idtest\",\"name\":\"idtest\",\"type\":\"int\",\"modelName\":\"网关设备_直连类型\",\"modelId\":\"2000085\",\"define\":{\"type\":\"int\"}},\"valType\":0,\"otherObj\":\"\"}],",
          "Id": 36,
          "Name": "具体设备",
          "Type": "device"
        }
      ],
      "EventRule": "36",
      "RuleDesc": "联动同一建筑",
      "RuleId": 29,
      "RuleName": "联动同一建筑332",
      "Status": 0,
      "ValidPeriod": "x-json:",
      "ValidType": 1
    }
  }
}
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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
InvalidParameter	参数错误。
InvalidParameter.ParamNotMatch	参数不匹配
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound	资源不存在。

事件列表查询

最近更新时间：2023-11-30 08:49:43

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

事件列表查询

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数，本接口取值：DescribeEventList。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
PageNumber	是	Integer	分页查询，第几页，必传，大于0 示例值：1
PageSize	是	Integer	每页条数，必传大于0 示例值：1
ApplicationToken	是	String	应用token 示例值：xxx
TriggerType	否	String	事件触发类型，(app, device, timer) 示例值：app
IdSet.N	否	Array of Integer	事件id详情 示例值：112

3. 输出参数

参数名称	类型	描述
Result	DescribeEventListRes	事件列表查询结果 示例值：{"EventDetailSet": [{"CreateTime": "2023-05-13 20:23:37", "DeviceType": "", "Id": 14, "LinkRuleSet": [{"Id": 23, "Name": ""}, {"Name": "13", "TriggerCondition": "x-json", "TriggerType": "timer", "ValidPeriod": "", "WID": ""}], "PageNumber": 1, "PageSize": 1, "TotalPage": 78, "TotalRow": 78}
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 事件列表

成功响应

输入示例

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

```
{
  "WorkspaceId": 1016,
  "PageNumber": 1,
  "PageSize": 1,
  "ApplicationToken": "baSTzPx0vZ6LPuv2EifNa5CqRBJ9hoY0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "529a10d6-0f9d-4ff9-8989-5dd0dc12aef6",
    "Result": {
      "EventDetailSet": [
        {
          "CreateTime": "2023-05-13 20:23:37",
          "DeviceType": "",
          "Id": 14,
          "LinkRuleSet": [
            {
              "Id": 23,
              "Name": ""
            }
          ],
          "Name": "13",
          "TriggerCondition": "x-json:",
          "TriggerType": "timer",
          "ValidPeriod": "",
          "WID": ""
        }
      ],
      "PageNumber": 1,
      "PageSize": 1,
      "TotalPage": 78,
      "TotalRow": 78
    }
  }
}
```

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
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound	资源不存在。

动作列表查询

最近更新时间：2023-11-30 08:49:50

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

动作列表查询

默认接口请求频率限制：20次/秒。

推荐使用 API Explorer

<> 点击调试

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

2. 输入参数

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

参数名称	必选	类型	描述
Action	是	String	公共参数 ，本接口取值：DescribeActionList。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间id 示例值：1016
PageNumber	是	Integer	分页查询，第几页，必传，大于0 示例值：1
PageSize	是	Integer	每页条数，必传大于0 示例值：1
ApplicationToken	是	String	应用token 示例值：baStzPx0vZ6LPuv2EifNa5CqRBj9hoY0
ActionType	否	String	动作类型，（app,device,toAlarm,toNotification） 示例值：device
IdSet.N	否	Array of Integer	事件id详情 示例值：214

3. 输出参数

参数名称	类型	描述
Result	DescribeActionListRes	动作列表查询结果 示例值：{"ActionDetailSet": [{"ActionDesc": "lena测试控制开灯", "ActionType": "deviceControl", "CreateT22", "LinkRuleSet": [], "MsgContent": "x-json:[{"model": "红灯(red)", "value": "10", "obj": {"name": "红灯", "type": "int", "min": 0, "max": 255}, {"model": "绿灯(green)", "value": "10", "type": "int", "obj": {"name": "绿灯", "type": "int", "modelName": "智能灯控模块", "modelId": "w0905001", "type": "int", "min": 0, "max": 255}, {"options": []}], {"MsgType": "model", "Name": "lena测试控制开灯", "SinkC": {"deviceType": "w0905001", "pid": "2000126", "limit": 0, "objectLevel": 0, "scopeLevel": "", "scopeRadius": ""}], "PageNumber": 1, "PageSize": 1, "TotalPage": 158, "TotalRow": 158}
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 获取动作列表

成功响应

输入示例

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

{
  "WorkspaceId": 1016,
  "PageNumber": 1,
  "PageSize": 1,
  "ApplicationToken": "baSTzPx0vZ6LPuv2EifNa5CqRBj9hoY0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "1c116023-0562-4488-933e-84df17c7f08c",
    "Result": {
      "ActionDetailSet": [
        {
          "ActionDesc": "lena测试控制开灯",
          "ActionType": "deviceControl",
          "CreateTime": "2023-05-15 16:29:17",
          "Id": 22,
          "LinkRuleSet": [],
          "MsgContent": "x-json:[{\\"model\\":\\"红灯(red)\",\\"value\\":\\"10\\",\\"obj\\":{\\"name\\":\\"红灯\\",\\"type\\":\\"int\\",\\"modelName\\":\\"智能灯控模块\\",\\"modelId\\":\\"2000126\\",\\"define\\":{\\"type\\":\\"int\\",\\"min\\":0,\\"max\\":255},\\"options\\":[],\\"type\\":\\"int\\"},{\\"model\\":\\"绿灯(green)\",\\"value\\":\\"10\\",\\"type\\":\\"int\\",\\"obj\\":{\\"name\\":\\"绿灯\\",\\"type\\":\\"int\\",\\"modelName\\":\\"智能灯控模块\\",\\"modelId\\":\\"2000126\\",\\"define\\":{\\"type\\":\\"int\\",\\"min\\":0,\\"max\\":255},\\"options\\":[]}}]",
          "MsgType": "model",
          "Name": "lena测试控制开灯",
          "SinkConfig": "x-json:{\\"nearbySearch\\":{\\"deviceType\\":\\"w0905001\\",\\"pid\\":2000126,\\"limit\\":0,\\"objectLevel\\":0,\\"scopeLevel\\":\\"\\",\\"scopeRadius\\":\\"\\",\\"deviceAssign\\":\\"deviceType\\"}}",
          "WID": ""
        }
      ],
      "PageNumber": 1,
      "PageSize": 1,
      "TotalPage": 158,
      "TotalRow": 158
    }
  }
}
```

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
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
ResourceNotFound	资源不存在。

上报应用消息

最近更新时间：2023-11-30 08:49:39

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

上报应用消息

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：ReportAppMessage。
Version	是	String	公共参数，本接口取值：2023-04-27。
Region	是	String	公共参数，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间Id 示例值：1016
Profile	是	MessageProfile	消息定义 示例值：{"modelId": "1000", "poiCode": "t234329", "appType": "security_monitoring"}
ReportTs	是	Integer	数据上报时间 示例值：1
Properties	是	String	属性定义 - KV的json格式,有效字段为x-json:后的字段 示例值：x-json:{"executeld": "RX_TASK"}
ApplicationToken	是	String	应用token 示例值：xxx
EventSet	否	String	事件定义 - KKV的json格式,有效字段为x-json:后的字段 示例值：x-json:{"face_control": {"image": {"data": "8bb1230b-78bb-4635-991b-9746dda24a9e", "type": 4}, "handler": {"type": 0, "data": "222111123123123123"}, "describe": "", "eventType": 0, "eventTs": 1650007523091}}
ServiceSet	否	String	服务定义 - KKV的json格式,有效字段为x-json:后的字段 示例值：x-json:{"}
ExtendTwo	否	String	扩展字段2，如：算法上报id，若为json格式，需传x-json:{"}，有效字段为x-json:后的字段 示例值：xxx
Echo	否	String	透传信息，若为json格式，需传x-json:{"}，有效字段为x-json:后的字段 示例值：xxx

3. 输出参数

参数名称	类型	描述
Result	EmptyRes	上报单条信息结果 示例值：{"Msg": "ok"}
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 单个上报消息

成功响应

输入示例

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

{
  "WorkspaceId": 1016,
  "Profile": {
    "AppType": "security_monitoring"
  },
  "ReportTs": 1,
  "Properties": "x-json:{\"executeld\": \"RX_TASK\"}",
  "ApplicationToken": "baSTzPx0vZ6LPuv2EifNa5CqRBJ9hoY0"
}
```

输出示例

```
{
  "Response": {
    "RequestId": "d542c6ba-6e4b-4d29-be9a-4e48c1c8616d",
    "Result": {
      "Msg": "ok"
    }
  }
}
```

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
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id
ResourceNotFound	资源不存在。

批量上报应用消息

最近更新时间：2023-11-30 08:49:52

1. 接口描述

接口请求域名：weilingwith.tencentcloudapi.com。

批量上报应用消息

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：BatchReportAppMessage。
Version	是	String	公共参数 ，本接口取值：2023-04-27。
Region	是	String	公共参数 ，详见产品支持的 地域列表 。
WorkspaceId	是	Integer	工作空间Id 示例值：1016
ApplicationToken	是	String	应用token 示例值：xxx
ReportSet.N	否	Array of ReportAppMessage	消息上报请求列表 示例值：{"WorkspaceId": 1016, "Profile": {"AppType": "security"}, "ReportTs": 1, "Properties": {"x-json": {"executeld": "RX_TASK"}}

3. 输出参数

参数名称	类型	描述
Result	BatchReportAppMessageRes	批量消息上报结果 示例值：{"Commit": 1, "SpanMap": [{"ReportId": "0000000000000000"}, {"ReportStatus": 1}], "TotalElements": 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 批量上报设备消息

成功响应

输入示例

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

{
```

```

"WorkspaceId": 1016,
"ApplicationToken": "baSTzPx0vZ6LPuv2EifNa5CqRBJ9hoY0",
"ReportSet": [
{
"WorkspaceId": 1016,
"Profile": {
"AppType": "security"
},
"ReportTs": 1,
"Properties": "x-json:{\"executId\": \"RX_TASK\"}"
}
]
}
    
```

输出示例

```

{
"Response": {
"RequestId": "b3cac805-15a2-4766-9ef6-77929e509db9",
"Result": {
"Commit": 1,
"SpanMap": [
{
"ReportId": "0000000000000000",
"ReportStatus": 1
}
],
"TotalElements": 1
}
}
}
    
```

5. 开发者资源

腾讯云 API 平台

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

API Inspector

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

SDK

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

- Tencent Cloud SDK 3.0 for Python: [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
InvalidParameter	参数错误。
InvalidParameterValue.InvalidWorkspaceId	错误的工作空间Id
ResourceNotFound	资源不存在。

数据结构

最近更新时间：2024-01-31 01:29:01

Action

动作信息

被如下接口引用：DescribeLinkRuleList。

名称	类型	描述
Id	Integer	动作id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Name	String	动作名 注意：此字段可能返回 null，表示取不到有效值。 示例值：a

ActionDetail

详细动作信息

被如下接口引用：DescribeActionList。

名称	类型	描述
Id	Integer	动作id 注意：此字段可能返回 null，表示取不到有效值。 示例值：22
Name	String	动作名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：a
ActionType	String	动作类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：deviceControl
ActionDesc	String	动作说明 注意：此字段可能返回 null，表示取不到有效值。 示例值：测试控制开灯
MsgType	String	消息类型，origin/custom/model 注意：此字段可能返回 null，表示取不到有效值。 示例值：model
MsgContent	String	消息内容,有效值为x-json:后的字段 注意：此字段可能返回 null，表示取不到有效值。 示例值：x-json:[{"model":"红灯(red)","value":"10","obj":{"name":"红灯","type":"int","modelName":"智能灯控模块","modelId":"2000126","define":{"type":"int","min":0,"max":255},"options":[]},"type":"int"}, {"model":"绿灯(green)","value":"10","type":"int","obj":{"name":"绿灯","type":"int","modelName":"智能灯控模块","modelId":"2000126","define":{"type":"int","min":0,"max":255},"options":[]}]}
CreateTime	String	创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2023-05-15 16:29:17
WID	String	设备唯一标识 注意：此字段可能返回 null，表示取不到有效值。 示例值：xxx
LinkRuleSet	Array of LinkRule	关联故障列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"Id":1,"Name":"a"}

名称	类型	描述
SinkConfig	String	动作下沉配置,有效值为x-json:后的字段 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: x-json:{"nearbySearch": { "deviceType": "w0905001", "pid": "2000126", "limit": "0", "objectLevel": "0", "scopeLevel": "", "scopeRadius": "", "deviceAssig

ActionObj

动作对象

被如下接口引用: DescribeRuleDetail。

名称	类型	描述
Id	Integer	动作id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 71
Name	String	动作名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: a
Type	String	动作类型。(app/推送消息至应用-携带空间设备: 无,appWithNearbyDevices/推送至应用-携带空间设备: 携带,device/推送消息至设备,nearbyDevices/推送消息至设备-事件所在范围内的设备,toAlarm/转换为告警,toNotification/转换为通知) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: nearbyDevices
Desc	String	动作说明 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: test
MsgType	String	消息类型, orgin/custom/model 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: nearbyDevices
MsgContent	String	消息内容 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: x-json:xxx
CreateTime	String	创建时间 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2023-05-23 14:38:13
SinkConfig	String	动作下沉配置 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: x-json:{"nearbySearch": { "deviceType": "w0201001", "pid": "0", "limit": "2", "objectLevel": "9", "scopeLevel": "6", "scopeRadius": "100", "deviceAssign": "
ApplyDevice	String	具体应用 (appid) /具体设备 (DIN/subID) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 10051

AddDeviceInfo

添加设备信息

被如下接口引用: BatchCreateDevice。

名称	类型	必选	描述
ProductId	Integer	是	产品id 示例值: 200022
SN	String	是	设备sn序列号 示例值: test
ParentWID	String	否	父设备wid, 不为空表示导入子设备 示例值: xxx-aaa

名称	类型	必选	描述
KeySource	Integer	否	密钥来源：0-使用产品密钥 1-使用设备特有的密钥 示例值：0

AdministrationData

行政区划数据结构

被如下接口引用：DescribeAdministrationByTag。

名称	类型	描述
AdministrationCode	String	行政区划编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：110000
AdministrationName	String	行政区划名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：北京市

AdministrativeDetail

行政区划详情

被如下接口引用：DescribeCityWorkspaceList。

名称	类型	描述
AdministrativeTypeCode	String	行政区域类型编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：QH03
AdministrativeCode	String	行政区域编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：440305
AdministrativeName	String	行政区域名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：南山区

AlarmInfo

告警信息

被如下接口引用：DescribeAlarmList。

名称	类型	描述
WorkspaceId	Integer	工作空间id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1016
Id	String	告警ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：0c6ec1cc-25d6-43bc-96b1-6b4131f1771a
Status	String	告警状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：unprocessed
Time	Integer	告警时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1693416976
Type	String	告警业务类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：energy

名称	类型	描述
TypeName	String	告警业务类型名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：能源
SubType	String	子告警类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：water_limit
SubTypeName	String	子告警类型名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：c
Level	Integer	告警级别id 注意：此字段可能返回 null，表示取不到有效值。 示例值：3
LevelName	String	告警级别名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：严重
AppId	Integer	上报应用appid 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
WID	String	设备wid 注意：此字段可能返回 null，表示取不到有效值。 示例值：b1fd7aa3-4407-4b82-9dc3-aff9b3034a9f
DeviceName	String	设备名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：20楼水表
Position	String	空间位置 注意：此字段可能返回 null，表示取不到有效值。 示例值：滨海大厦-7F
ReportImg	ReportImg	上报图片 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"Data":"","Type":0}
Desc	String	告警描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：水量使用超限
HandlePersonSet	Array of HandlerPersonInfo	处理人 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"Id":12,"Name":"test"}
HandleRecordSet	Array of HandleRecordInfo	处理记录 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"AppId":0,"Description":"消警-规则消警","ExtendOne":"x-json:","FileSet":[],"Id":308504,"Name":"","OperationType":"消警","Time":"1693416991","Type":"rule_alarm_elimination"}
Extend	String	扩展信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：x-json:{"alarmLevel":3,"alarmValue":13730945.6}
ExtendOne	String	应用扩展字段1 注意：此字段可能返回 null，表示取不到有效值。 示例值：x-json:xxxx
ExtendTwo	String	应用扩展字段2 注意：此字段可能返回 null，表示取不到有效值。 示例值：x-json:xxxx
Echo	String	应用透传字段,有效字段为x-json后的字段 注意：此字段可能返回 null，表示取不到有效值。 示例值：x-json:abc

AlarmLevelInfo

告警级别详情

被如下接口引用: DescribeAlarmLevelList。

名称	类型	描述
LevelId	Integer	级别id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 3
LevelName	String	级别名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 严重

AlarmStatusData

告警状态返回结构体

被如下接口引用: DescribeAlarmStatusList。

名称	类型	描述
StatusID	String	告警状态ID 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: unprocessed
StatusName	String	告警状态名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 待处理
StatusType	String	告警状态类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0

AlarmTypeDetailInfo

告警类型详情信息

被如下接口引用: DescribeAlarmTypeList。

名称	类型	描述
Id	Integer	告警类型id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 26
ParentId	Integer	父节点id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 25
Type	Integer	0-标准告警类型, 1-自定义告警类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
Name	String	告警名称类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 人流数量异常
EnglishName	String	告警类型英文名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: excessive

AlarmTypeInfo

告警类型

被如下接口引用: BatchKillAlarm, DescribeAlarmList。

名称	类型	必选	描述
Type	String	是	告警父类型 示例值: xxx-a
SubType	String	否	告警子类型(如果传告警子类型, 则必传父类型) 示例值: xxx-b

ApiContent

API参数信息

被如下接口引用: DescribeInterfaceList。

名称	类型	描述
Id	String	所属API的id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 741f3abd-639c-49ad-b9ed-b52a0f08a62e
Name	String	参数名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 获取视频流
Type	String	参数类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: string
Dynamic	Boolean	是否为动态值 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: true
Required	Boolean	是否必填 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: true
Value	String	参数值 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 741f3abd-639c-49ad-b9ed-b52a0f08a62e
DefaultValue	String	默认值 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 741f3abd-639c-49ad-b9ed-b52a0f08a62e

ApiInfo

API描述

被如下接口引用: DescribeInterfaceList。

名称	类型	描述
Apild	String	API的id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 741f3abd-639c-49ad-b9ed-b52a0f08a62e
Name	String	API名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 获取视频流
Appid	String	API所属应用的id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 10001
WorkspaceId	String	API所属的项目空间的id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1101

名称	类型	描述
PoiCode	String	API所属目录的编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：api000
Type	Integer	接口分类0. 其他服务 1. IOT服务 2. 空间服务 3. 微应用服务 4. 场景服务 5. AI算法服务 6. 任务算法服务 7. 第三方服务 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
DataAudit	Integer	数据授权 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
ApplyAudit	Integer	是否需要申请 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
Description	String	API详情 注意：此字段可能返回 null，表示取不到有效值。 示例值：获取视频流
Address	String	API地址 注意：此字段可能返回 null，表示取不到有效值。 示例值：/video/ControlCameraPtz
Method	String	请求方法类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：GET
Status	Integer	API状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
PreviewUrl	String	API预览地址 注意：此字段可能返回 null，表示取不到有效值。 示例值：http://t-apigw.twins.tencent.com/proxy/138c04de-4f1e-4bff-9498-4f16f2ed4cd5/video/Contro
QueryParams	Array of ApiContent	query参数 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"DefaultValue": "", "Dynamic": false, "Id": "02066fae-0f05-4256-8554-506479c6cca4", "Name": "id", "Required": false, "Type": "string", "Value": "211"}]
PathParams	Array of ApiContent	路径参数 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"DefaultValue": "", "Dynamic": false, "Id": "02066fae-0f05-4256-8554-506479c6cca4", "Name": "id", "Required": false, "Type": "string", "Value": "211"}]
RequestHeaders	Array of ApiContent	请求头 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"DefaultValue": "", "Dynamic": false, "Id": "02066fae-0f05-4256-8554-506479c6cca4", "Name": "id", "Required": false, "Type": "string", "Value": "211"}]
ResponseHeaders	Array of ApiContent	响应头 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"DefaultValue": "", "Dynamic": false, "Id": "02066fae-0f05-4256-8554-506479c6cca4", "Name": "id", "Required": false, "Type": "string", "Value": "211"}]
IsCommonSpace	Boolean	是否为公共空间接口 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Body	String	请求体（base64编码） 注意：此字段可能返回 null，表示取不到有效值。 示例值： W3tclnV1aWRcljpljZmZDcxNjRjOWY5ZTQ0ZjM5M2I1N2UyZTA2OWJiYjY1XClsXCJrZXIcljplclnJvb3Rc

被如下接口引用: DescribeApplicationList。

名称	类型	描述
ApplicationId	String	应用分配的appid 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 10001
Name	String	应用中文名 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 能源管理
Address	String	应用地址 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: http://www.capi.woa.com
ApplicationLogo	ApplicationLogo	应用logo 注意: 此字段可能返回 null, 表示取不到有效值。
Type	Integer	应用类型, 0:saas应用 1:平台应用 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
EnglishName	String	engine 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 应用英文名
Description	String	能源管理应用 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 应用详情

ApplicationList

应用列表

被如下接口引用: DescribeApplicationList。

名称	类型	描述
ApplicationInfoList	Array of ApplicationInfo	应用列表 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: [{"Address": "http://checkproject-a6f6718d.dtwins.tencent.com/testcodingpush", "ApplicationId": "10041", "ApplicationLogo": {"FileId": "359cc1b5-88c1-4dd6-8ff4-21d0b36ac7e2", "Url": "https://wetwin-manager-1258344699.cos.ap-guangzhou.myqcloud.com/%2F100055/0/20230717/359cc1b5-88c1-4dd6-8ff4-21d0b36ac7e2/%E8%B5%84%E4%BA%A7%E7%AE%A1%E7%90%86-%E9%BB%91.png?q-sign-algorithm=sha1&q-ak=AKIDiFWWFo67rLZXFgrINinb8NL2SaR2M0OU&q-sign-time=1693452005%3B1693453205&q-key-time=1693452005%3B1693453205&q-header-list=host&q-url-param-list=&q-signature=6bf531d1b08d6f75eb7e153c24746a61d7be37d0"}, {"Description": "1234561111", "EnglishName": "zctest1111", "Name": "智能资产管理", "Type": 0}, {"Address": "http://t-ws.twins.tencent.com/proxy/29098b25-447d-437e-987a-d493d55e84f5/weiling-pubsub/test", "ApplicationId": "10048", "ApplicationLogo": {"FileId": "", "Url": ""}, {"Description": "", "EnglishName": "MsgCenter", "Name": "消息服务", "Type": 0}, {"Address": "http://t.twins.tencent.com/", "ApplicationId": "10051", "ApplicationLogo": {"FileId": "", "Url": ""}, {"Description": "", "EnglishName": "IOT", "Name": "孪生中台(物联感知中心)", "Type": 0}]
TotalCount	String	当前查询条件命中的数据总条数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 符合查询条件的数据总条数

ApplicationLogo

应用logo

被如下接口引用: DescribeApplicationList。

名称	类型	描述

名称	类型	描述
FileId	String	logo图片对应的fileId 注意：此字段可能返回 null，表示取不到有效值。 示例值：d442d82a-ded2-49f1-92e7-81a7f0f45ac6
Url	String	logo图片地址 注意：此字段可能返回 null，表示取不到有效值。 示例值：https://wetwin-manager-1258344699.cos.ap-guangzhou.myqcloud.com/%2F100055/0/20231030/72f6c306-5819-42dc-98a9-758b4e7892aa/%E7%A7%9F%E6%88%B7%E5%88%9D%E5%A7%8B%E5%8C%96%E8%84%9A%E6%9C%AC-1698664462444.sql?q-sign-algorithm=sha1&q-ak=AKIDiFWWFo67rLZXFgrINinb8NL2SaR2M0OU&q-sign-time=1698664464%3B1698665664&q-key-time=1698664464%3B1698665664&q-header-list=host&q-url-param-list=&q-signature=45bb085655d6a9722df5ec4dfd1aa80b1cdb08f8

ApplicationTokenInfo

应用Token令牌信息

被如下接口引用：CreateApplicationToken, DescribeEdgeApplicationToken。

名称	类型	描述
Token	String	应用申请调用API的令牌 注意：此字段可能返回 null，表示取不到有效值。 示例值：Oq9lShbmhCT3FLnemDCecrXAnAHQe6SJ

BatchCreateDeviceRes

批量新增设备接口返回结果

被如下接口引用：BatchCreateDevice。

名称	类型	描述
SuccessSet	Array of CreateDeviceSucceeded	新增成功的设备列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"ParentWID": "", "ProductId": 2000054, "SN": "atest", "WID": "75c83fbd-4862-4c4e-8091-18ccba3e2419"}
FailSet	Array of CreateDeviceFailed	新增失败的设备列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"ParentWID": "", "ProductId": 2000054, "SN": "atest", "Reason": "xxxx"}

BatchReportAppMessageRes

批量消息上报结果

被如下接口引用：BatchReportAppMessage。

名称	类型	描述
TotalElements	Integer	上报数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Commit	Integer	提交数量（推送成功） 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
SpanMap	Array of ReportMsgRes	消息推送结果列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"ReportId": "0000000000000000", "ReportStatus": 1}

BuildingListRes

建筑列表响应体

被如下接口引用: DescribeBuildingList。

名称	类型	描述
BuildingProfileList	Array of BuildingProfile	建筑列表 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: [{ "Address": "北京市,丰台区样例模型", "BuildingId": "2da7d294-483e-4033-a99a-49f7323fd6ef", "BuildingName": "样例模型1", "Latitude": 39.927326, "Longitude": 116.39177, "SpaceCode": "000009" }, { "Address": "1#楼", "BuildingId": "dfad5b19-79d8-4e40-9a8c-8efa32a1f425", "BuildingName": "1#楼", "Latitude": 39.92442, "Longitude": 116.38269, "SpaceCode": "000007" }]

BuildingModel

建筑模型信息

被如下接口引用: DescribeBuildingModel。

名称	类型	描述
ElementId	String	构件ID 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 82d5fb5a-52d0-4636-a225-d46245e911ef
ElementName	String	构件名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 验证专用
ModelType	String	模型类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: MODEL_ORIGIN
ModelUrl	String	模型URL 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/cb6bad63-0855-4dc2-8908-20518627479b.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqqVvLagaL8IBZq5896tTxOfjBL2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3Ddbac70b180e5a2d8c343506dae755a01669ffb04

BuildingModelRes

建模模型信息响应体

被如下接口引用: DescribeBuildingModel。

名称	类型	描述
----	----	----

名称	类型	描述
Models	Array of BuildingModel	建模模型信息出参 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{ "ElementId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ElementName": "验证专用", "ModelType": "MODEL_ORIGIN", "ModelUrl": "https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/cb6bad63-0855-4dc2-8908-20518627479b.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqVvLagaL8IBZq5896tTxOfjBL2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3Ddbac70b180e5a2d8c343506dae755a01669ffb04" }, { "ElementId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ElementName": "验证专用-外壳", "ModelType": "MODEL_HULL", "ModelUrl": "https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/ff847c36-89f8-4644-9c1f-d265bd5d5167.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqVvLagaL8IBZq5896tTxOfjBL2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3D7af0ab15cab83314e44026fedc0927467d07b776" }, { "ElementId": "0b00f88f-36f1-4ae3-a7b4-5db12332c005", "ElementName": "-1F", "ModelType": "MODEL_STOREY", "ModelUrl": "https://bim-1301743406.cos.ap-guangzhou.myqcloud.com/100055/1016/2ee86ae4-41a1-4c45-840b-4e9eae1b83a1.glb?sign=q-sign-algorithm%3Dsha1%26q-ak%3DAKIDsyqVvLagaL8IBZq5896tTxOfjBL2h8L%26q-sign-time%3D1692758001%3B1787366001%26q-key-time%3D1692758001%3B1787366001%26q-header-list%3D%26q-url-param-list%3D%26q-signature%3De9b58a5293201d24c19e2bc28b49404035b569c1" }]

BuildingProfile

建筑概要信息

被如下接口引用：DescribeBuildingList, DescribeBuildingProfile。

名称	类型	描述
BuildingId	String	建筑id 注意：此字段可能返回 null，表示取不到有效值。 示例值：2da7d294-483e-4033-a99a-49f7323fd6ef
BuildingName	String	建筑名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：样例模型1
SpaceCode	String	空间编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：000009
Longitude	Float	经度 注意：此字段可能返回 null，表示取不到有效值。 示例值：116.38269
Latitude	Float	纬度 注意：此字段可能返回 null，表示取不到有效值。 示例值：39.92442
Address	String	地址 注意：此字段可能返回 null，表示取不到有效值。 示例值：北京市,丰台区样例地址1#楼

BuildingProfileRes

查询建筑信息响应体

被如下接口引用：DescribeBuildingProfile。

名称	类型	描述
BuildingProfile	BuildingProfile	建筑概要信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：{ "Address": "广东省,深圳市,南山区测试建筑", "BuildingId": "2e374df6-7ba7-45d2-a44d-2a92bb2a6668", "BuildingName": "测试建筑", "Latitude": 22.533192, "Longitude": 113.93048, "SpaceCode": "000003" }

CameraExtendInfoRes

视频扩展信息结果

被如下接口引用: DescribeCameraExtendInfo。

名称	类型	描述
SaveType	String	存储方式 (nvr或cosmtav) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: nvr
SaveDay	Integer	云存储天数 (save_type是cosmtav时这个参数才有效) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
LiveResolution	Integer	实时分辨率 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
HistoryResolution	Integer	历史分辨率 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0

ControlDeviceRes

设备控制结果

被如下接口引用: ControlDevice。

名称	类型	描述
WID	String	设备Id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 无=70e673c7-11b8-4ca0-b11c-dfa360a44638
Code	Integer	指令接受, 0表示成功 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
Result	String	控制结果 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: success
Seq	String	批量大于1时, 可用此seq进行链路追踪 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 21909094-b304-4942-af6a-eb781ddd0a2a

ControlDeviceSet

设备控制后返回结果集合

被如下接口引用: ControlDevice。

名称	类型	描述
Set	Array of ControlDeviceRes	设备控制后返回结果集合 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"Code": 0, "Result": "success", "Seq": "21909094-b304-4942-af6a-eb781ddd0a2a", "WID": "70e673c7-11b8-4ca0-b11c-dfa360a44638"}

CreateDeviceFailed

导入失败设备信息

被如下接口引用: BatchCreateDevice。

名称	类型	描述
ProductId	Integer	产品id 注意：此字段可能返回 null，表示取不到有效值。 示例值：200022
ParentWID	String	父设备wid，不为空表示导入自设备 注意：此字段可能返回 null，表示取不到有效值。 示例值：xxx-aaaa
Reason	String	失败原因 注意：此字段可能返回 null，表示取不到有效值。 示例值：atest
SN	String	设备sn序列号 注意：此字段可能返回 null，表示取不到有效值。 示例值：atest

CreateDeviceSucceeded

导入成功设备信息

被如下接口引用：BatchCreateDevice。

名称	类型	描述
ProductId	Integer	产品id 注意：此字段可能返回 null，表示取不到有效值。 示例值：200200
ParentWID	String	父设备wid，不为空表示导入自设备 注意：此字段可能返回 null，表示取不到有效值。 示例值：xxx-aaa
WID	String	设备编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：75c83fbd-4862-4c4e-8091-18ccba3e2419
SN	String	设备sn序列号 注意：此字段可能返回 null，表示取不到有效值。 示例值：atest

CustomField

自定义字段

被如下接口引用：DescribeDeviceList。

名称	类型	必选	描述
Id	Integer	否	字段id 示例值：1
Val	String	否	字段内容 示例值：test

CustomFieldInfo

自定义字段

被如下接口引用：DescribeDeviceList。

名称	类型	描述
Id	Integer	字段id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

名称	类型	描述
Key	String	字段key 注意：此字段可能返回 null，表示取不到有效值。 示例值：color
Name	String	字段名 注意：此字段可能返回 null，表示取不到有效值。 示例值：颜色
Val	String	字段值 注意：此字段可能返回 null，表示取不到有效值。 示例值：a

DescribeActionListRes

动作列表查询结果

被如下接口引用：DescribeActionList。

名称	类型	描述
PageNumber	Integer	第几页 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
PageSize	Integer	每页条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
TotalPage	Integer	总页数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
TotalRow	Integer	总条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
ActionDetailSet	Array of ActionDetail	动作列表查询集合 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"ActionDesc": "lena测试控制开灯", "ActionType": "deviceControl", "CreateTime": "2023-05-15 16:29", "MsgContent": "x-json:[{"model": "红灯(red)", "value": "10", "obj": {"name": "红灯", "type": "int", "modelName": "块", "modelId": "2000126", "define": {"type": "int", "min": 0, "max": 255}, "options": [], "type": "int"}, {"model": "绿灯(green)", "value": "10", "type": "int", "obj": {"name": "绿灯", "type": "int", "modelName": "智能灯控模块", "modelId": "20", "type": "int", "min": 0, "max": 255}, "options": []}]", "MsgType": "model", "Name": "lena测试控制开灯", "SinkConfig": {"deviceType": "w0905001", "pid": 2000126, "limit": 0, "objectLevel": 0, "scopeLevel": "", "scopeRadius": "", "devi": ""}}

DescribeAdministrationByTagRes

根据Tag获取行政区划列表返回结构

被如下接口引用：DescribeAdministrationByTag。

名称	类型	描述
List	Array of AdministrationData	行政区划列表 注意：此字段可能返回 null，表示取不到有效值。

DescribeAlarmLevelListRes

告警级别枚举获取

被如下接口引用：DescribeAlarmLevelList。

名称	类型	描述

名称	类型	描述
AlarmLevelSet	Array of AlarmLevelInfo	告警级别枚举获取数组 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"LevelId": 3, "LevelName": "严重"}

DescribeAlarmListRes

告警列表回包

被如下接口引用：DescribeAlarmList。

名称	类型	描述
PageNumber	Integer	第几页 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
PageSize	Integer	每页条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
TotalPage	Integer	总页数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
TotalRow	Integer	总条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：20
AlarmInfoSet	Array of AlarmInfo	告警列表集合 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"Appld": 0, "Desc": "水量使用超限", "DeviceName": "20楼水表", "Echo": "", "Extend": "x-json: {"alarmLevel": 3, "alarmValue": 13730945.6}, "ExtendOne": "x-json:", "ExtendTwo": "x-json:", "HandlePersonSet": [], "HandleRecordSet": [], "Id": "0c6ec1cc-25d6-43bc-96b1-6b4131f1771a", "Level": 3, "LevelName": "严重", "Position": "滨海大厦-7F", "ReportImg": {"Data": "", "Type": 0}, "Status": "unprocessed", "SubType": "water_limit", "SubTypeName": "水量使用超限", "Time": 1693416976, "Type": "energy", "TypeName": "能源", "WID": "b1fd7aa3-4407-4b82-9dc3-aff9b3034a9f", "Workspaceld": 1016}

DescribeAlarmStatusListRes

告警状态列表返回

被如下接口引用：DescribeAlarmStatusList。

名称	类型	描述
List	Array of AlarmStatusData	告警状态返回结构 注意：此字段可能返回 null，表示取不到有效值。

DescribeAlarmTypeListRes

告警类型列表回包

被如下接口引用：DescribeAlarmTypeList。

名称	类型	描述
AlarmTypeSet	Array of AlarmTypeDetailInfo	告警类型查询列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"EnglishName": "excessive", "Id": 26, "Name": "人流数量异常", "ParentId": 25, "Type": 1}

DescribeCityWorkspaceListRes

通过城市id查询工作空间列表

被如下接口引用: DescribeCityWorkspaceList。

名称	类型	描述
WorkspaceSet	Array of WorkspaceInfo	通过城市id查询工作空间列表结果 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"workspaceId": 1016, "nameCh": "华南园区项目", "description": "华南园区项目", "status": 0, "parkName": "园区简称", "parkNum": "001", "administDetail": {}}

DescribeDeviceListRes

设备列表查询结果

被如下接口引用: DescribeDeviceList。

名称	类型	描述
PageNumber	Integer	第几页 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
PageSize	Integer	每页条数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
TotalPage	Integer	总页数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2
TotalRow	Integer	总条数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 20
DeviceDataSet	Array of DeviceDataInfo	设备信息集合 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"ActiveTime": "2023-04-13T09:41:38+08:00", "DeviceName": "1c902558-d360-4a93-9b6f-1b49970534f7", "DeviceTagSet": ["中文标签测试"], "DeviceTypeCode": "1c902558-d360-4a93-9b6f-1b49970534f7", "DeviceTypeName": "1c902558-d360-4a93-9b6f-1b49970534f7", "FieldList": [{"Id": 1, "Key": "color", "Name": "颜色", "Val": ""}], "IsActive": 1, "IsLive": false, "Location": {"X": -197697.515625, "Y": 22156.90234375, "Z": 0}, "ModelId": "", "ModelName": "", "OtaVersion": "v1", "ParentWID": "", "ParentWIDName": "", "ProductAbility": 1, "ProductId": 2000053, "ProductName": "其它网关", "SN": "tony001", "SpaceInfoSet": [{"Code": "001234", "Id": "7e0eba7e-aa9b-43bd-8a68-2ea7153828de", "Level": 6, "Name": "验证专用"}], "WID": "1c902558-d360-4a93-9b6f-1b49970534f7"}

DescribeEventListRes

事件列表查询结果

被如下接口引用: DescribeEventList。

名称	类型	描述
PageNumber	Integer	第几页 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
PageSize	Integer	每页条数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
TotalPage	Integer	总页数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2
TotalRow	Integer	总条数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2

名称	类型	描述
EventDetailSet	Array of EventDetail	事件信息列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"CreateTime": "2023-05-13 20:23:37", "DeviceType": "", "Id": 14, "LinkRuleSet": [{"Id": 23, "Name": ""}], "Name": "13", "Tr

DescribeLinkRuleListRes

联动规则列表查询结果

被如下接口引用：DescribeLinkRuleList。

名称	类型	描述
PageNumber	Integer	第几页 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
PageSize	Integer	每页条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
TotalPage	Integer	总页数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
TotalRow	Integer	总条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
LinkRuleSet	Array of LinkRuleInfo	联动规则列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"ActionSet": [{"Id": 28, "Name": "瞿杨动作"}], "BeginDate": "2023-05-03", "EndDate": "2023-05-22", "EventSet": [{"Id": 33, "Name": "体验"}], "Id": 24, "Name": "体验", "Status": 24, "ValidPeriod": ""}

DeviceDataInfo

设备数据信息

被如下接口引用：DescribeDeviceList。

名称	类型	描述
WID	String	设备ID， wid 注意：此字段可能返回 null，表示取不到有效值。 示例值：1c902558-d360-4a93-9b6f-1b49970534f7中文标签测试
DeviceName	String	设备名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：1c902558-d360-4a93-9b6f-1b49970534f7
DeviceTypeCode	String	设备类型Id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1c902558-d360-4a93-9b6f-1b49970534f7
DeviceTypeName	String	设备类型名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：1c902558-d360-4a93-9b6f-1b49970534f7
ProductId	Integer	产品Id 注意：此字段可能返回 null，表示取不到有效值。 示例值：2000053
ProductName	String	产品名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：其它网关

名称	类型	描述
ProductAbility	Integer	产品能力:信令数据、音视频。二进制数值中第0位表示信令数据、第1位表示音视频。1(信令数据), 3(具有信令数据以及音视频能力)。 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
SpaceInfoSet	Array of DeviceSpaceInfo	设备位置信息 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"Code": "001234", "Id": "7e0eba7e-aa9b-43bd-8a68-2ea7153828de", "Level": 6, "Name": "验证专用"}
ModelId	String	模型id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1232
ModelName	String	模型名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: a
DeviceTagSet	Array of String	设备标签名, 非必填 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 中文标签测试
IsActive	Integer	激活状态(1激活、0未激活) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
ActiveTime	String	激活时间 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2023-04-13T09:41:38+08:00
IsLive	Boolean	推流状态(推流中、未推流) 仅摄像机有的状态 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: false
ParentWID	String	设备所属父设备id(子设备才有) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: xxxxx
ParentWIDName	String	设备所有父设备名称(子设备才有) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: atest
SN	String	序列号 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: tony001
Location	DeviceLocation	设备点位坐标值 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"X": -197697.515625, "Y": 22156.90234375, "Z": 0}
FieldList	Array of CustomFieldInfo	自定义字段 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"Id": 1, "Key": "color", "Name": "颜色", "Val": ""}
GroupInfo	String	分组信息 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: xx/xx/xx
DeviceStatus	String	通信在/离线状态(online=normal+fault, offline) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: online
Status	String	设备业务状态(normal、fault、offline) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: normal

DeviceLocation

设备点位坐标值

被如下接口引用：DescribeDeviceList。

名称	类型	描述
X	Float	点位X坐标值 注意：此字段可能返回 null，表示取不到有效值。 示例值：-197697.515625
Y	Float	点位Y坐标值 注意：此字段可能返回 null，表示取不到有效值。 示例值：-197697.515625
Z	Float	点位Z坐标值 注意：此字段可能返回 null，表示取不到有效值。 示例值：0

DeviceModifyInfo

设备修改信息入参

被如下接口引用：ModifyDeviceName。

名称	类型	必选	描述
WID	String	是	设备id 示例值：be2ea997-b7f4-4167-9545-9aa275158a92
DeviceName	String	是	修改后的设备名字 示例值：a

DeviceShadowInfo

设备影子信息

被如下接口引用：DescribeDeviceShadowList。

名称	类型	描述
WID	String	设备ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：026d1dd8-a6d7-4fb7-9a94-0df07edaaff8
DeviceShadow	String	设备影子数据,返回有效数据为"x-json:"后字段 注意：此字段可能返回 null，表示取不到有效值。 示例值：x-json:{"desired":{},"reported":{"events":{},"profile":{"modelId":"w0745001"},"properties":{"Ept":96,"IA":32,"IB":42},"reportTs":1693370542126}}
DeviceShadowUpdateTime	String	设备影子更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2023-08-30 12:42:22

DeviceShadowRes

设备影子查询列表

被如下接口引用：DescribeDeviceShadowList。

名称	类型	描述
PageNumber	Integer	第几页 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
PageSize	Integer	每页条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：5

名称	类型	描述
TotalPage	Integer	总页数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
TotalRow	Integer	总条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：10
Set	Array of DeviceShadowInfo	设备影子列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"DeviceShadow": "x-json":{"desired":{},"reported":{"events":{},"profile":{"modelId":"w0745001"},"properties":{"Ept":96,"IA":32,"IB":42},"reportTs":1693370542126}},"DeviceShadowUpdateTime": "2023-08-30 12:42:22","WID": "026d1dd8-a6d7-4fb7-9a94-0df07edaaff8"}

DeviceSpaceInfo

设备位置信息

被如下接口引用：DescribeDeviceList。

名称	类型	描述
Id	String	空间Id 注意：此字段可能返回 null，表示取不到有效值。 示例值：7e0eba7e-aa9b-43bd-8a68-2ea7153828de
Name	String	空间名字 注意：此字段可能返回 null，表示取不到有效值。 示例值：验证专用
Level	Integer	空间级别 注意：此字段可能返回 null，表示取不到有效值。 示例值：5
Code	String	空间编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：001234

DeviceSpaceInfoRes

设备挂接的空间信息

被如下接口引用：DescribeSpaceInfoByDeviceId。

名称	类型	描述
BuildingId	String	建筑id 注意：此字段可能返回 null，表示取不到有效值。 示例值：82d5fb5a-52d0-4636-a225-d46245e911ef
ElementId	String	构件id 注意：此字段可能返回 null，表示取不到有效值。 示例值：8a74aba0-46e3-4097-89e6-7adbeb9ccaef
EntityType	String	构件类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：TBIM_CAMERA
ElementName	String	构件名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：1232533半球摄像头
Level	Integer	构件级别 注意：此字段可能返回 null，表示取不到有效值。 示例值：9

名称	类型	描述
BottomHeight	Integer	底部标高（单位mm） 注意：此字段可能返回 null，表示取不到有效值。 示例值：3000
SpaceCode	String	空间编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：00003000300000008

DeviceStatusInfo

设备状态信息

被如下接口引用：DescribeDeviceStatusList。

名称	类型	描述
WID	String	设备ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：be2ea997-b7f4-4167-9545-9aa275158a92
DeviceStatus	String	设备状态（online=normal+fault、offline） 注意：此字段可能返回 null，表示取不到有效值。 示例值：online
DeviceStatusUpdateTime	String	设备状态更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2023-08-31 00:57:28
Status	String	设备业务状态（normal、fault、offline） 注意：此字段可能返回 null，表示取不到有效值。 示例值：normal
IsAlive	Boolean	推流状态。推流中=true，未推流=false 注意：此字段可能返回 null，表示取不到有效值。 示例值：true

DeviceStatusRes

设备状态获取接口结果

被如下接口引用：DescribeDeviceStatusList。

名称	类型	描述
PageNumber	Integer	第几页 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
PageSize	Integer	每页条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：10
TotalPage	Integer	总页数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
TotalRow	Integer	总条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：20
DeviceStatusSet	Array of DeviceStatusInfo	设备状态信息列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"DeviceStatus": "online", "DeviceStatusUpdateTime": "2023-08-31 00:57:28", "IsAlive": true, "Status": "normal", "WID": "be2ea997-b7f4-4167-9545-9aa275158a92"}

DeviceStatusStatRes

设备状态统计结果

被如下接口引用: DescribeDeviceStatusStat。

名称	类型	描述
WorkspaceId	Integer	工作空间Id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1016
Total	Integer	汇总数。在线 (正常+故障) + 离线 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 8791
NormalSum	Integer	正常数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 87
OfflineSum	Integer	离线数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 8689
FaultSum	Integer	故障数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 15
DeviceTypeOverviewSet	Array of DeviceTypeOverview	设备类型概览列表 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"DeviceType":"string","Fault":0,"Name":"string","Offline":0,"Normal":0,"Total":0}
StatLevelSet	Array of StatLevel	设备类型统计列表 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"StatDeviceTypeSet":[{"DeviceType":"string","Fault":0,"Name":"string","Offline":0,"Normal":0,"Total":0}],"Fau

DeviceTagInfo

设备标签信息

被如下接口引用: DescribeDeviceTagList。

名称	类型	描述
TagId	Integer	标签Id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 35
TagName	String	标签名字 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 西安真实摄像头

DeviceTagRes

设备标签列表查询结果

被如下接口引用: DescribeDeviceTagList。

名称	类型	描述
PageNumber	Integer	第几页 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
PageSize	Integer	每页条数 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 20

名称	类型	描述
TotalPage	Integer	总页数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
TotalRow	Integer	总条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：40
Set	Array of DeviceTagInfo	设备标签列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"TagId": 35,"TagName": "西安真实摄像头"}

DeviceType

设备类型

被如下接口引用：DescribeDeviceTypeList。

名称	类型	描述
Code	String	设备类型编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：w0301006
Name	String	设备类型名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：烟感报警器
ParentCode	String	父设备类型编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：0301
ParentName	String	父设备类型名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：火灾自动报警及联动系统
IsSubsystem	Integer	是否子系统，1是 注意：此字段可能返回 null，表示取不到有效值。 示例值：0

DeviceTypeOverview

设备类型概览信息

被如下接口引用：DescribeDeviceStatusStat。

名称	类型	描述
DeviceType	String	设备类型值 注意：此字段可能返回 null，表示取不到有效值。 示例值：w02002
Name	String	设备类型名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：atest
Total	Integer	汇总数。在线（正常+故障）+ 离线 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
Normal	Integer	正常数 注意：此字段可能返回 null，表示取不到有效值。 示例值：0

名称	类型	描述
Offline	Integer	离线数 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
Fault	Integer	故障数 注意：此字段可能返回 null，表示取不到有效值。 示例值：0

DeviceTypeSet

设备类型列表

被如下接口引用：DescribeDeviceTypeList。

名称	类型	描述
Set	Array of DeviceType	设备类型列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"Code": "w0301006", "IsSubsystem": 0, "Name": "烟感报警器", "ParentCode": "0301", "ParentName": "火灾自动报警及联动系统"}

ElementCoordinates

构件地理坐标

被如下接口引用：DescribePropertyList。

名称	类型	描述
Longitude	Float	经度 注意：此字段可能返回 null，表示取不到有效值。 示例值：117.25139600000003
Latitude	Float	纬度 注意：此字段可能返回 null，表示取不到有效值。 示例值：39.12852507853361
Altitude	Float	高程 注意：此字段可能返回 null，表示取不到有效值。 示例值：0

ElementProfile

构件概要信息

被如下接口引用：DescribeElementProfilePage, DescribeElementProfileTree。

名称	类型	描述
BuildingId	String	建筑id 注意：此字段可能返回 null，表示取不到有效值。 示例值：82d5fb5a-52d0-4636-a225-d46245e911ef
ElementId	String	构件id 注意：此字段可能返回 null，表示取不到有效值。 示例值：61049c1b-158d-41bd-9f0b-df30a50401bd
EntityType	String	构件类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：TBIM_BUILDINGSTOREY
ElementName	String	构件名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：RF

名称	类型	描述
Level	Integer	构件空间级别 注意：此字段可能返回 null，表示取不到有效值。 示例值：7
BottomHeight	Integer	底部标高（单位mm） 注意：此字段可能返回 null，表示取不到有效值。 示例值：3000
Sort	Integer	排序 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
SpaceCode	String	空间编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：000030003
SpaceTypeCode	String	空间分类编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：""
SpaceTypeName	String	空间分类名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：""
ParentElementId	String	父级构件id 注意：此字段可能返回 null，表示取不到有效值。 示例值：82d5fb5a-52d0-4636-a225-d46245e911ef
SpacePoild	String	空间层级类型编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：""
ElementDesc	String	构件描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：""
IsDelete	Integer	删除标记 注意：此字段可能返回 null，表示取不到有效值。 示例值：0

ElementProfilePageRes

构件分页查询响应体

被如下接口引用：DescribeElementProfilePage。

名称	类型	描述
TotalCount	Integer	构件总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
List	Array of ElementProfile	构件列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"BottomHeight": -100, "BuildingId": "956bd069-c802-4bbb-b325-18d30d7bcd3c", "ElementDesc": "", "ElementId": "1Kjh8u2qf4uRSkCPoKigTI", "ElementName": "1F（结）-0.100", "EntityType": "TBIM_BUILDINGSTOREY", "IsDelete": 0, "Level": 7, "ParentElementId": "956bd069-c802-4bbb-b325-18d30d7bcd3c", "Sort": 1, "SpaceCode": "000185004", "SpacePoild": "", "SpaceTypeCode": "", "SpaceTypeName": ""}]

ElementProfileTreeNode

构件树节点信息

被如下接口引用：DescribeElementProfileTree。

名称	类型	描述
ElementProfile	ElementProfile	构件概要信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：{ "BottomHeight": 3000, "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ElementDesc": "", "ElementId": "6042d426-6973-47ff-8ce1-3a63ef9f3c2f", "ElementName": "", "EntityType": "TBIM_SLASHWALL", "IsDelete": 0, "Level": 8, "ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd", "Sort": 1, "SpaceCode": "00003000300000001", "SpacePoild": "", "SpaceTypeCode": "", "SpaceTypeName": "" }
Children	Array of ElementProfileTreeNode	子节点信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]

ElementProfileTreeRes

构件树响应体

被如下接口引用：DescribeElementProfileTree。

名称	类型	描述
BuildingId	String	建筑id 注意：此字段可能返回 null，表示取不到有效值。 示例值：82d5fb5a-52d0-4636-a225-d46245e911ef
ParentElementId	String	父级构件id 注意：此字段可能返回 null，表示取不到有效值。 示例值：61049c1b-158d-41bd-9f0b-df30a50401bd
Root	ElementProfileTreeNode	构件树 注意：此字段可能返回 null，表示取不到有效值。 示例值：{ "Children": [{ "Children": [], "ElementProfile": { "BottomHeight": 3000, "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ElementDesc": "", "ElementId": "6042d426-6973-47ff-8ce1-3a63ef9f3c2f", "ElementName": "", "EntityType": "TBIM_SLASHWALL", "IsDelete": 0, "Level": 8, "ParentElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd", "Sort": 1, "SpaceCode": "00003000300000001", "SpacePoild": "", "SpaceTypeCode": "", "SpaceTypeName": "" } }], "ElementProfile": { "BottomHeight": 3000, "BuildingId": "82d5fb5a-52d0-4636-a225-d46245e911ef", "ElementDesc": "", "ElementId": "61049c1b-158d-41bd-9f0b-df30a50401bd", "ElementName": "RF", "EntityType": "TBIM_BUILDINGSTOREY", "IsDelete": 0, "Level": 7, "ParentElementId": "", "Sort": 1, "SpaceCode": "000030003", "SpacePoild": "", "SpaceTypeCode": "", "SpaceTypeName": "" } }

ElementProperty

构件属性信息

被如下接口引用：DescribePropertyList。

名称	类型	描述
Name	String	属性名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：Materials and Finishes
Description	String	属性描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：""
Content	String	属性内容 注意：此字段可能返回 null，表示取不到有效值。 示例值：{Material=None}

ElementPropertyRes

构件属性信息响应体

被如下接口引用: DescribePropertyList。

名称	类型	描述
BuildingId	String	建筑id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "377e4f9d-d20e-4209-b1e8-0ddb32e888c8"
ElementId	String	构件id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "b66aece1-62e9-49d3-872e-6e66156a853b"
PropertySet	Array of ElementProperty	构件属性集合 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: [{"Content": "{Elevation=9500, Story Above=None}", "Description": "", "Name": "Constraints"}, {"Content": "{Computation Height=0}", "Description": "", "Name": "Dimensions"}, {"Content": "{Is ground plane=0}", "Description": "", "Name": "Display"}, {"Content": "{storeyType=TOP_FLOOR, floorHeight=3000, bottomHeight=9500}", "Description": "", "Name": "EDITOR_BUILDING_STOREY_EXTEND"}, {"Content": "{Scope Box=None}", "Description": "", "Name": "Extents"}, {"Content": "{IfcGUID=1LvAmy3\$j18hHP\$GEcP3rk, Export to IFC=0}", "Description": "", "Name": "IFC Parameters"}, {"Content": "{Structural=No, Design Option=None, Building Story=Yes}", "Description": "", "Name": "Identity Data"}, {"Content": "{Name=Level : F4, Type=C_上标高+层标, Childs=[Roofs], Family=Level, Type Id=87266, Category=OST_Levels, ClassName=OdBmCDANode, ODAUniqueID=214592, LocalizedName=, Family and Type=Level : C_上标高+层标}", "Description": "", "Name": "Invalid group"}, {"Content": "{Material=None}", "Description": "", "Name": "Materials and Finishes"}]
Coordinates	ElementCoordinates	构件地理坐标 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"Altitude": 0, "Latitude": 39.12852507853361, "Longitude": 117.25139600000003 }
Translate	ElementTranslate	构件偏移量 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"X": 0, "Y": 0, "Z": 0 }
ElementName	String	构件名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "Level : F4"
EntityTypeCode	String	构件类型代码 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "TBIM_BUILDINGSTOREY"
EntityTypeName	String	构件类型名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "楼层"

ElementTranslate

构件平移信息

被如下接口引用: DescribePropertyList。

名称	类型	描述
X	Float	X方向偏移量 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
Y	Float	Y方向偏移量 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
Z	Float	Z方向偏移量 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0

EmptyRes

空结果返回

被如下接口引用: AddAlarmProcessRecord, BatchKillAlarm, ChangeAlarmStatus, ControlCameraPTZ, ModifyDeviceName, ReportAppMessage, StopVideoStreaming, UpdateWorkspaceParkAttributes。

名称	类型	描述
Msg	String	返回请求状态,成功ok, 失败error 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: ok

Event

事件信息

被如下接口引用: DescribeLinkRuleList。

名称	类型	描述
Id	Integer	事件id或动作Id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 24
Name	String	事件名称或动作名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: a

EventDetail

事件详细信息

被如下接口引用: DescribeEventList。

名称	类型	描述
Id	Integer	事件id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 14
Name	String	事件名 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: a
TriggerType	String	事件触发类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: timer
TriggerCondition	String	事件触发条件, 返回为x-json后的字段 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: x-json:xxx
ValidPeriod	String	有效期 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: xxx
LinkRuleSet	Array of LinkRule	关联规则列表 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"Id": 23, "Name": ""}
CreateTime	String	创建时间 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2023-05-13 20:23:37
DeviceType	String	设备类型, 当触发类型为deviceType时返回 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: xxx

名称	类型	描述
WID	String	设备的wid, 当触发类型是device返回 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: xxx-xxx-xxx

EventObj

事件对象

被如下接口引用: DescribeRuleDetail。

名称	类型	描述
Id	Integer	事件id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 36
Name	String	事件名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 具体设备
Type	String	事件触发类型名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: device
Condition	String	时间触发条件 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "x-json:[{"type":"objmodel","attr":"properties.idtest","oper":"=","val":"348","obj":{"id":"idtest","name":"idtest","type":"int","modelName":"网关设备_直连类型","modelId":"2000085","define":{"type":"int"},"valType":0,"otherObj":""}]"

FileDownloadURL

文件下载URL

被如下接口引用: DescribeFileDownloadURL。

名称	类型	描述
FileURL	String	下载地址 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: https://xxx.xxx

FileInfo

文件信息

被如下接口引用: DescribeAlarmList。

名称	类型	描述
FileId	String	文件id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 192
ReportName	String	名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: test

FileUploadURL

获取文件上传URL接口回包

被如下接口引用: DescribeFileUploadURL。

名称	类型	描述
UploadURL	String	上传地址 注意：此字段可能返回 null，表示取不到有效值。 示例值：http://xxx.xxx
FileId	String	文件Id 注意：此字段可能返回 null，表示取不到有效值。 示例值：xxx
DownloadURL	String	下载地址 注意：此字段可能返回 null，表示取不到有效值。 示例值：http://xxx.xxx

HandleRecordInfo

告警处理记录

被如下接口引用：DescribeAlarmList。

名称	类型	描述
Id	Integer	告警处理记录id 注意：此字段可能返回 null，表示取不到有效值。 示例值：30293
Description	String	描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：test
Name	String	名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：test
OperationType	String	操作类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：消警
Time	String	处理时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1693416991
Type	String	类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：rule_alarm_elimination
FileSet	Array of FileInfo	文件列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"Id":123,"Name":"a"}
AppId	Integer	应用appid 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
ExtendOne	String	扩展字段1，存非孛生中台用户id 注意：此字段可能返回 null，表示取不到有效值。 示例值：x-json:xxx

HandlerPersonInfo

告警处理人列表

被如下接口引用：DescribeAlarmList。

名称	类型	描述
----	----	----

名称	类型	描述
Id	String	用户id 注意：此字段可能返回 null，表示取不到有效值。 示例值：12
Name	String	用户名 注意：此字段可能返回 null，表示取不到有效值。 示例值：test

LinkRule

关联规则信息

被如下接口引用：DescribeActionList, DescribeEventList。

名称	类型	描述
Id	Integer	关联联动规则id 注意：此字段可能返回 null，表示取不到有效值。 示例值：23
Name	String	关联联动规则名字 注意：此字段可能返回 null，表示取不到有效值。 示例值：a

LinkRuleInfo

联动规则信息

被如下接口引用：DescribeLinkRuleList。

名称	类型	描述
Id	Integer	联动id 注意：此字段可能返回 null，表示取不到有效值。 示例值：24
Name	String	联动名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：体验
EventSet	Array of Event	事件列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"Id": 33,"Name": "体验"}
ActionSet	Array of Action	动作列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"Id": 28,"Name": "动作"}
Status	Integer	状态：0开，-1关 注意：此字段可能返回 null，表示取不到有效值。 示例值：24
BeginDate	String	起始时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2023-05-03
EndDate	String	结束时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：2023-05-22
ValidPeriod	String	有效周期内容,有效字段为x-json后的字段 注意：此字段可能返回 null，表示取不到有效值。 示例值：x-json:xxxx

MessageProfile

上报消息概要

被如下接口引用: BatchReportAppMessage, ReportAppMessage。

名称	类型	必选	描述
AppType	String	是	应用类型 示例值: security_monitoring
ModelId	String	否	模型Id 示例值: 1000
PoiCode	String	否	设备类型 示例值: t234329

ModelInfo
模型基础信息

被如下接口引用: DescribeModelList。

名称	类型	描述
WorkspaceId	Integer	工作空间id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1016
ModelName	String	模型名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 边缘温控器
ModelId	String	物模型id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2000055
RelatedProduct	Array of RelatedProduct	关联产品信息 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"Id": 2000055, "Name": "边缘温控器"}
DeviceTypeName	String	设备类型名 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 空调温控器
DeviceType	String	设备类型id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: w0404024
ModelType	Integer	物模型类型, 产品模型/标准模型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
ModelParams	String	模型参数内容,有效字段为"x-json:"后的字段 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: x-json:"xxxx"

ModelSet
模型列表查询结果

被如下接口引用: DescribeModelList。

名称	类型	描述
PageNumber	Integer	第几页 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1

名称	类型	描述
PageSize	Integer	每页条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
TotalPage	Integer	总页数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
TotalRow	Integer	总条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：2
Set	Array of ModellInfo	模型基础信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"DeviceType": "w0404024", "DeviceTypeName": "空调温控器", "ModelId": "2000055", "ModelName": "边缘温控器", "ModelParams": "x-json:{}", "ModelType": 1, "RelatedProduct": [{"id": "2000055", "Name": "边缘温控器"}], "WorkspaceId": "1016"}, {"TotalPage": 156, "TotalRow": 156}

ProcessRecordInfo

处理记录项

被如下接口引用：AddAlarmProcessRecord。

名称	类型	必选	描述
Id	String	是	告警的id 示例值：cdf85c7c-cf04-41b5-9631-fa4a7290e615
ProcessTime	Integer	是	处理时间，毫秒 示例值：1688114263000
ProcessType	String	是	处理类型，此处操作类型固定填add_record 示例值：add_record-other
Processor	String	否	注：此字段填写的是孪生中台的用户，非孪生中台用户不填该字段 [{"id": "123", "name": "tes"}] 示例值：[{"id": "123", "name": "tes"}]
ProcessDescription	String	否	处理描述信息 示例值：test
AttachedFileId	String	否	附加文件标识 示例值：[{"fileId": "b87ae7ec-6874-4103-bfa0-2489374476eb", "fileName": "ecs.png"}]

ProductInfo

产品信息

被如下接口引用：DescribeProductList。

名称	类型	描述
WorkspaceId	Integer	工作空间id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1016
ProductId	Integer	产品PID 注意：此字段可能返回 null，表示取不到有效值。 示例值：200055
ProductName	String	产品名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：atest

名称	类型	描述
DeviceTypeName	String	设备类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：温控器
DeviceTypeId	String	设备类型id 注意：此字段可能返回 null，表示取不到有效值。 示例值：w04042
Attribute	Integer	产品属性，如：网关（1）、直连设备（2）、子设备（3） 注意：此字段可能返回 null，表示取不到有效值。 示例值：3
ProductType	String	产品型号 注意：此字段可能返回 null，表示取不到有效值。 示例值：atest
ProductAbility	Integer	产品能力:信令数据、音视频。二进制数值中第0位表示信令数据、第1位表示音视频。1（信令数据），3（具有信令数据以及音视频能力）。 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Manufacturer	String	生产厂商 注意：此字段可能返回 null，表示取不到有效值。 示例值：atest
MaintenanceMfr	String	维保厂商 注意：此字段可能返回 null，表示取不到有效值。 示例值：atest
ModelName	String	物模型名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：温控
ModelId	String	物模型id 注意：此字段可能返回 null，表示取不到有效值。 示例值：200055
ModelType	Integer	物模型类型，产品模型/标准模型 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

ProductSet

产品列表查询结果

被如下接口引用：DescribeProductList。

名称	类型	描述
PageNumber	Integer	第几页 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
PageSize	Integer	每页条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：20
TotalPage	Integer	总页数 注意：此字段可能返回 null，表示取不到有效值。 示例值：3
TotalRow	Integer	总条数 注意：此字段可能返回 null，表示取不到有效值。 示例值：100

名称	类型	描述
Product	Array of ProductInfo	产品信息列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"WorkspaceId": 1016, "ProductId": 2000055, "ProductName": "边缘温控器", "DeviceTypeName": "空调温控器", "DeviceType": "w0404024", "Attribute": 3, "ProductType": "", "ProductAbility": 1, "Manufacturer": "dagakng", "MaintenanceMfr": "", "ModelName": "边缘温控器", "ModelId": "2000055", "ModelType": 1}

RawInfo

视频流Raw协议信息

被如下接口引用：DescribeVideoLiveStream。

名称	类型	描述
SM4Vector	String	加密向量（如果视频网关选择流为非加密传输这个参数可忽略） 注意：此字段可能返回 null，表示取不到有效值。 示例值：LT50CB5unVXLDjkM
NATIP	String	专线ip（非专线接入可忽略） 注意：此字段可能返回 null，表示取不到有效值。 示例值：1.47.110.0
StreamToken	String	客户端握手鉴权参数 注意：此字段可能返回 null，表示取不到有效值。 示例值：18c434ca
Port	Integer	拉流端口 注意：此字段可能返回 null，表示取不到有效值。 示例值：9001
StreamEnKey	String	视频流加密key，目前为AES128加密KEY（如果视频网关选择流为非加密传输这个参数可忽略） 注意：此字段可能返回 null，表示取不到有效值。 示例值：VRK0JsivUkHWon7K
IP	String	拉流公网地址（非公网接入时，这个地址是内网地址） 注意：此字段可能返回 null，表示取不到有效值。 示例值：185.47.110.0
InnerIP	String	拉流内网地址 注意：此字段可能返回 null，表示取不到有效值。 示例值：11.121.500.32

RecordInfo

录像信息

被如下接口引用：DescribeVideoCloudRecord。

名称	类型	描述
StartTime	Integer	本录像片段开始时间（s） 注意：此字段可能返回 null，表示取不到有效值。 示例值：1616466060
EndTime	Integer	本录像片段结束时间（s） 注意：此字段可能返回 null，表示取不到有效值。 示例值：1616466098
VideoURL	String	录像片段文件url 注意：此字段可能返回 null，表示取不到有效值。 示例值： http://1304886180.vod2.myqcloud.com/cac4da69vodcq1304886180/618f1d115285890815878290403/f0.mp4

RelatedProduct

关联产品信息

被如下接口引用: DescribeModelList。

名称	类型	描述
Id	Integer	关联产品pid 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2000055
Name	String	关联产品名字 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 边缘温控器

ReportAppMessage

单条消息上报请求

被如下接口引用: BatchReportAppMessage。

名称	类型	必选	描述
WorkspaceId	Integer	是	工作空间Id 示例值: 1016
Profile	MessageProfile	否	消息定义 示例值: {"modelId": 1000, "poiCode": "t234329", "appType": "security_monitoring"}
ReportTs	Integer	否	数据上报时间 示例值: 1
Properties	String	否	属性定义 - KVV, 若为json格式, 需在前加上x-json:, 有效字段为x-json:后的字段 示例值: x-json:{"executeld": "RX_TASK"}
EventSet	String	否	事件定义 - KKV, 若为json格式, 需在前加上x-json:, 有效字段为x-json:后的字段 示例值: x-json:{"face_control": {"image": {"data": "8bb1230b-78bb-4635-991b-9746dda24a9e", "type": 4}, "handler": {"type": 0, "data": "22211123123123123"}, "describe": "", "eventType": 0, "eventTs": 1650007523091}}
ServiceSet	String	否	服务定义 - KKV, 若为json格式, 需在前加上x-json:, 有效字段为x-json:后的字段 示例值: x-json:xxx
ExtendTwo	String	否	扩展字段2, 如: 算法上报id, 若为json格式, 需在前加上x-json: 示例值: x-json:xxx
Echo	String	否	透传信息, 若为json格式, 需在前加上x-json:, 有效字段为x-json:后的字段 示例值: x-json:{"executeld": "RX_TASK"}

Reporting

上报图片列表

被如下接口引用: DescribeAlarmList。

名称	类型	描述
Type	Integer	类型 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 4
Data	String	数据 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: testurl

ReportMsgRes

上报消息结果

被如下接口引用: BatchReportAppMessage。

名称	类型	描述
ReportId	String	上报消息对应下标的16位标识Id, 即第几个消息 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0000000000000000
ReportStatus	Integer	上报消息结果, 1表示成功推送, 0表示推送失败 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1

RuleDetailRes

规则详情查询结果

被如下接口引用: DescribeRuleDetail。

名称	类型	描述
RuleId	Integer	联动id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 29
RuleName	String	联动名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 联动同一建筑332
RuleDesc	String	联动说明 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 联动同一建筑
ValidType	Integer	1 全天有效, 0: 固定时间段有效 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
ValidPeriod	String	有效期, json字符串(全天有效时空) 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: x-json:xxxx
BeginDate	String	起始时间 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2023-8-22
EndDate	String	结束时间 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 2023-8-24
Status	Integer	启用状态。1-启用, 0-停用 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 0
EventRule	String	触发规则, 事件的组合 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 26
EventInfoSet	Array of EventObj	事件对象集合 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: "Condition": "x-json:[{"type":"objmodel","attr":"properties.idtest","oper":"=","val":"348","obj":{"id":"idtest","name":"idtest","type":"int","modelName":"网关设备_直连类型","modelId":"2000085","define":{"type":"int"},"valType":0,"otherObj":""}],"Id": 36,"Name": "具体设备","Type": "device"}]
ActionInfoSet	Array of ActionObj	动作对象集合 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: {"ApplyDevice":"","CreateTime":"2023-05-23 14:38:13","Desc":"事件范围内同一建筑","Id": 71,"MsgConte":"origin","Name":"事件范围内同一建筑","SinkConfig":{"x-json":{"nearbySearch":{"deviceType":"w0201001","pid":0,"limit":2,"objectLevel":9,"scopeLevel":6,"scopeRadius":100,"deviceAssig":"nearbyDevices"}}

SceneInfo

场景信息

被如下接口引用：DescribeSceneList。

名称	类型	描述
Sceneld	String	场景id 注意：此字段可能返回 null，表示取不到有效值。 示例值：0abe73dd-d06f-44f5-af82-43dea23af230
SceneName	String	场景名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：三层商务办公楼单位修正
Version	String	场景版本 注意：此字段可能返回 null，表示取不到有效值。 示例值：20230705160410780

SceneListRes

场景列表响应体

被如下接口引用：DescribeSceneList。

名称	类型	描述
SceneList	Array of SceneInfo	场景列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"Sceneld": "0abe73dd-d06f-44f5-af82-43dea23af230", "SceneName": "三层商务办公楼单位修正", "Version": "20230705160410780"}, {"Sceneld": "e9f2b87f-9078-4546-8d4f-99356ec3f82c", "SceneName": "商务办公楼模型", "Version": "20230630151240729"}]

SpaceDataListStatsRes

查询项目空间楼栋数量与建筑面积响应体

被如下接口引用：DescribeWorkSpaceBuildingCountAndArea。

名称	类型	描述
List	Array of SpaceDataStats	楼栋数量与建筑面积列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"BuildingArea": 0, "BuildingCount": 2, "Workspaceld": "1018", "WorkspaceName": "华北园区项目"}, {"BuildingArea": 202928.77732501808, "BuildingCount": 127, "Workspaceld": "1016", "WorkspaceName": "华南园区项目"}, {"BuildingArea": 0, "BuildingCount": 8, "Workspaceld": "1015", "WorkspaceName": "公共空间"}]

SpaceDataStats

项目空间楼栋数量与建筑面积出参

被如下接口引用：DescribeWorkSpaceBuildingCountAndArea。

名称	类型	描述
Workspaceld	String	工作空间ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：1016
WorkspaceName	String	工作空间名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：华南园区项目
BuildingCount	Integer	楼栋数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：127

名称	类型	描述
BuildingArea	Float	建筑面积 注意：此字段可能返回 null，表示取不到有效值。 示例值：202928.77732501808

SpaceDataTotalStatsRes

查询租户楼栋数量和楼栋建筑面积相应体

被如下接口引用：DescribeTenantBuildingCountAndArea。

名称	类型	描述
BuildingCount	Integer	总楼栋数量 注意：此字段可能返回 null，表示取不到有效值。 示例值：137
BuildingArea	Float	总建筑面积 注意：此字段可能返回 null，表示取不到有效值。 示例值：202928.77732501808

SpaceDeviceIdListRes

查询指定空间id列表响应

被如下接口引用：DescribeSpaceDeviceIdList。

名称	类型	描述
DeviceIds	Array of String	设备id列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：["2b538c57-933f-4820-b1d0-4df70d64aee3", "ad655ba1-975a-4238-94d7-5d2015d1cf1f", "5a1a0b73-7aa3-4699-a893-21846e7853ec", "0a198d6e-40f0-499f-bf99-7a959b7f4ef2"]

SpaceDeviceRelation

设备-空间绑定关系

被如下接口引用：DescribeSpaceDeviceRelationList。

名称	类型	描述
DeviceId	String	设备id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1c902558-d360-4a93-9b6f-1b49970534f7
ElementId	String	构件id 注意：此字段可能返回 null，表示取不到有效值。 示例值：8a74aba0-46e3-4097-89e6-7adbeb9ccaef

SpaceDeviceRelationRes

设备-空间绑定关系响应体

被如下接口引用：DescribeSpaceDeviceRelationList。

名称	类型	描述
SpaceDeviceRelationList	Array of SpaceDeviceRelation	设备空间绑定关系列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{ "DeviceId": "1c902558-d360-4a93-9b6f-1b49970534f7", "ElementId": "8a74aba0-46e3-4097-89e6-7adbeb9ccaef" }, { "DeviceId": "b66603af-7f8c-46c0-8712-1353a08d5806", "ElementId": "af7f7d83-17f8-4dff-80e2-a5129a39eab0" }]

SpaceInfo

项目空间详细信息

被如下接口引用: DescribeWorkspaceList。

名称	类型	描述
WorkspaceId	Integer	项目空间id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1101
TenantId	Integer	租户id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 100000
EnglishName	String	英文名 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: testspace
ChineseName	String	中文名 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 华南项目空间
Description	String	项目空间描述 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 华南项目空间
Status	Integer	项目空间状态:0 启用 1 停用 -1 已删除 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
IsCommWorkspace	Boolean	是否是公共空间 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: true
ValidityStartTime	String	有效期开始时间 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1693193293
ValidityEndTime	String	有效期结束时间 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1693193293
Selected	Integer	选中状态 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1
IsSystem	Integer	系统生成状态 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: 1

SpaceRelation

空间层级关系

被如下接口引用: DescribeSpaceRelationByDeviceId。

名称	类型	描述
ElementId	String	构件id 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: bdbbf537-d6c4-4b1e-80ae-e0bcaab8b06f
ElementName	String	构件名称 注意: 此字段可能返回 null, 表示取不到有效值。 示例值: test_no_100

名称	类型	描述
Level	Integer	空间层级 注意：此字段可能返回 null，表示取不到有效值。 示例值：9
SpaceCode	String	空间编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：00016200200000003
ParentSpaceCode	String	父级空间编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：000162002
Children	Array of SpaceRelation	子构件信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：[]

SpaceRelationRes

空间层级关系响应体

被如下接口引用：DescribeSpaceRelationByDeviceId。

名称	类型	描述
SpaceRelation	SpaceRelation	空间层级关系 注意：此字段可能返回 null，表示取不到有效值。 示例值：{ "Children": [{ "Children": [{ "Children": [], "ElementId": "bdbbf537-d6c4-4b1e-80ae-e0bcaab8b06f", "ElementName": "test_no_100", "Level": 9, "ParentSpaceCode": "000162002", "SpaceCode": "00016200200000003" }], "ElementId": "f3c5ffa5-05e3-4c26-a447-0edf71aa9783", "ElementName": "1F", "Level": 7, "ParentSpaceCode": "000162", "SpaceCode": "000162002" }], "ElementId": "377a53a6-c5a7-445d-a3b7-c2d721f95408", "ElementName": "小智新设备挂接0628", "Level": 6, "ParentSpaceCode": "", "SpaceCode": "000162" }

SpaceType

空间分类

被如下接口引用：DescribeSpaceTypeList。

名称	类型	描述
SpaceTypeCode	String	空间分类编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：DIANTIJIAN
SpaceTypeName	String	空间分类名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：电梯间

SpaceTypeListRes

空间分类列表响应体

被如下接口引用：DescribeSpaceTypeList。

名称	类型	描述
SpaceTypeList	Array of SpaceType	空间分类列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{ "SpaceTypeCode": "DIANTIJIAN", "SpaceTypeName": "电梯间" }, { "SpaceTypeCode": "LOUTIJIAN", "SpaceTypeName": "楼梯间" }, { "SpaceTypeCode": "WODEFANGJIAN", "SpaceTypeName": "我的房间" }, { "SpaceTypeCode": "XIUXISHI", "SpaceTypeName": "休息室" }]

SsoDepartment

部门用户

被如下接口引用：DescribeTenantDepartmentList。

名称	类型	描述
DepartmentId	String	部门ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：21
Name	String	部门名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：财务部
ParentDepartmentId	String	父级部门ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：9

SsoDepartmentsResult
部门用户结果

被如下接口引用：DescribeTenantDepartmentList。

名称	类型	描述
Total	Integer	总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Departments	Array of SsoDepartment	部门列表 注意：此字段可能返回 null，表示取不到有效值。 示例值： [{"DepartmentId": "21", "Name": "财务部", "ParentDepartmentId": "9"}, {"DepartmentId": "22", "Name": "商务部", "ParentDepartmentId": "10"}]

SsoTeamUser
部门用户

被如下接口引用：DescribeWorkspaceUserList。

名称	类型	描述
UserId	String	用户ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：0d0a98561b3c4f72963eba5c5fcf56e1
RealName	String	用户名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：张三
UserType	String	用户类型，1-超级管理员；2-1号管理员；3-普通管理员；99-普通用户 注意：此字段可能返回 null，表示取不到有效值。 示例值：3
TenantId	String	所属租户ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：10001
Email	String	邮箱 注意：此字段可能返回 null，表示取不到有效值。 示例值：test@163.com
Phone	String	电话 注意：此字段可能返回 null，表示取不到有效值。 示例值：186xxxx234

名称	类型	描述
Status	Integer	用户状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：用户状态，0待审核，1正常启用，2禁用
CreateAt	Integer	创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1680836659
DepartmentId	String	部门ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：12
DepartmentName	String	部门名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：财务部
LinkFilter	Integer	是否关联权限 注意：此字段可能返回 null，表示取不到有效值。 示例值：1

SsoTeamUserResult

空间用户结果

被如下接口引用：DescribeWorkspaceUserList。

名称	类型	描述
Total	Integer	总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
Users	Array of SsoTeamUser	部门用户列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"UserId":"0d0a98561b3c4f72963eba5c5fcf56e1","RealName":"张三","UserType":"3","TenantId":"10001","Email":"test@163.com","Phone":"186xxxx234","Status":0,"CreateAt":"1680836659","LinkFilter":1}]

SsoUser

用户结果

被如下接口引用：DescribeTenantUserList。

名称	类型	描述
UserId	String	用户ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：0d0a98561b3c4f72963eba5c5fcf56e1
UserName	String	用户昵称 注意：此字段可能返回 null，表示取不到有效值。 示例值：张三
RealName	String	用户名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：张三
UserType	String	用户类型，1-超级管理员；2-1号管理员；3-普通管理员；99-普通用户 注意：此字段可能返回 null，表示取不到有效值。 示例值：3
TenantId	String	所属租户ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：100055

名称	类型	描述
UserGroup	String	所属组ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：12
Email	String	邮箱 注意：此字段可能返回 null，表示取不到有效值。 示例值：test@163.com
Phone	String	电话 注意：此字段可能返回 null，表示取不到有效值。 示例值：18612341234
Status	Integer	用户状态，0待审核，1正常启用，2禁用 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
CreateAt	Integer	创建时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1680836659
UpdateAt	Integer	更新时间 注意：此字段可能返回 null，表示取不到有效值。 示例值：1680836659
BelongTeam	Integer	是否属于团队，0不可用，1属于，2不属 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
DepartmentId	String	部门ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：123
DepartmentName	String	部门名称 注意：此字段可能返回 null，表示取不到有效值。 示例值：财务部
DepartmentUserId	Integer	子账户ID 注意：此字段可能返回 null，表示取不到有效值。 示例值：123
Password	String	密码 注意：此字段可能返回 null，表示取不到有效值。 示例值：***

SsoUserResult

租户人员结果

被如下接口引用：DescribeTenantUserList。

名称	类型	描述
Total	Integer	总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：28
Users	Array of SsoUser	租户人员数据 注意：此字段可能返回 null，表示取不到有效值。

StatDeviceType

设备类型统计

被如下接口引用：DescribeDeviceStatusStat。

名称	类型	描述
----	----	----

名称	类型	描述
Total	Integer	汇总数。在线（正常+故障）+ 离线 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
Normal	Integer	正常数 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
Offline	Integer	离线数 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
Fault	Integer	故障数 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
Name	String	设备名 注意：此字段可能返回 null，表示取不到有效值。 示例值：test
DeviceType	String	设备类型 注意：此字段可能返回 null，表示取不到有效值。 示例值：w020202

StatLevel

层级统计

被如下接口引用：DescribeDeviceStatusStat。

名称	类型	描述
Total	Integer	汇总数。在线（正常+故障）+ 离线 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
NormalSum	Integer	正常数 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
OfflineSum	Integer	离线数 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
FaultSum	Integer	故障数 注意：此字段可能返回 null，表示取不到有效值。 示例值：0
SpaceCode	String	空间id 注意：此字段可能返回 null，表示取不到有效值。 示例值：100110
StatDeviceTypeSet	Array of StatDeviceType	设备类型统计列表 注意：此字段可能返回 null，表示取不到有效值。 示例值： { "DeviceType": "string", "Fault": 0, "Name": "string", "Offline": 0, "Normal": 0, "Total": 0 }

VideoCloudRecordRes

云录像接口结果

被如下接口引用：DescribeVideoCloudRecord。

名称	类型	描述
----	----	----

名称	类型	描述
TotalCount	Integer	录像信息总数 注意：此字段可能返回 null，表示取不到有效值。 示例值：1
RecordSet	Array of RecordInfo	录像信息列表 注意：此字段可能返回 null，表示取不到有效值。 示例值： { "StartTime": "1616466060", "EndTime": "1616466098", "VideoURL": "http://1304886180.vod2.myqcloud.com/cac4dat" }

VideoRecordStreamRes

视频流查询结果

被如下接口引用：DescribeVideoLiveStream。

名称	类型	描述
FLV	String	FLV协议格式视频流 注意：此字段可能返回 null，表示取不到有效值。 示例值："https://t-video.twins.tencent.com:443/hls1935/76d8109e-64cd-4a85-b011-4a0c29a654e8.flv?ticket=75f3"
RTMP	String	RTMP协议格式视频流 注意：此字段可能返回 null，表示取不到有效值。 示例值：rtmp://11.141.177.239:8080/hls1935/76d8109e-64cd-4a85-b011-4a0c29a654e8?ticket=fb63fb0c65c45748
HLS	String	HLS协议格式视频流 注意：此字段可能返回 null，表示取不到有效值。 示例值：https://t-video.twins.tencent.com:443/hls1935/76d8109e-64cd-4a85-b011-4a0c29a654e8.m3u8?ticket=35
WebRTC	String	WebRtc协议格式视频流 注意：此字段可能返回 null，表示取不到有效值。 示例值：webrtc://t-video.twins.tencent.com:443/hls1935/76d8109e-64cd-4a85-b011-4a0c29a654e8?ticket=bb3f73
RAW	RawInfo	RAW协议格式视频流 注意：此字段可能返回 null，表示取不到有效值。 示例值： { "IP": "183.47.110.99", "InnerIP": "11.141.200.32", "NATIP": "", "Port": "9001", "SM4Vector": "LT50CB5unVXLDjkM", "StreamE" }
Stream	String	视频流的唯一标识 注意：此字段可能返回 null，表示取不到有效值。 示例值：76d8109e-64cd-4a85-b011-4a0c29a654e8

WorkspaceInfo

工作空间信息描述

被如下接口引用：DescribeCityWorkspaceList。

名称	类型	描述
WorkspaceId	Integer	工作空间Id 注意：此字段可能返回 null，表示取不到有效值。 示例值：1016
ChineseName	String	工作空间中文名字 注意：此字段可能返回 null，表示取不到有效值。 示例值：空间
Description	String	工作空间描述 注意：此字段可能返回 null，表示取不到有效值。 示例值：腾讯大厦
Status	Integer	工作空间是否删除状态 注意：此字段可能返回 null，表示取不到有效值。 示例值：0

名称	类型	描述
ParkName	String	该工作空间绑定的区/县的行政区名字 注意：此字段可能返回 null，表示取不到有效值。 示例值：园区简称
ParkNum	String	该工作空间绑定的区/县的行政区编码 注意：此字段可能返回 null，表示取不到有效值。 示例值：001
AdministrativeDetailSet	Array of AdministrativeDetail	获取该工作空间绑定的区/县的上级行政区划信息 注意：此字段可能返回 null，表示取不到有效值。 示例值：{"administTypeCode": "QH01", "administCode": "440000", "administName": "广东省"}

WorkspaceInfoList

项目空间列表

被如下接口引用：DescribeWorkspaceList。

名称	类型	描述
List	Array of SpaceInfo	项目空间列表 注意：此字段可能返回 null，表示取不到有效值。 示例值：[{"ChineseName": "lv测试空间", "Description": "lv测试空间", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "1744387200", "ValidityStartTime": "1682524800", "WorkspaceId": 1092}, {"ChineseName": "RayDataWeb", "Description": "RayDataWeb测试使用测完后删除", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "2000390400", "ValidityStartTime": "1684684800", "WorkspaceId": 1166}, {"ChineseName": "华东项目空间", "Description": "演练项目空间", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "1693411200", "ValidityStartTime": "1682870400", "WorkspaceId": 1175}, {"ChineseName": "ces", "Description": "123", "EnglishName": "", "IsCommWorkspace": false, "Status": 1, "TenantId": 100055, "ValidityEndTime": "1688400000", "ValidityStartTime": "1688313600", "WorkspaceId": 1524}, {"ChineseName": "公共空间", "Description": "租户默认工作空间", "EnglishName": "comm_workspace", "IsCommWorkspace": true, "Status": 0, "TenantId": 100055, "ValidityEndTime": "4834882335", "ValidityStartTime": "1681282335", "WorkspaceId": 1015}, {"ChineseName": "华南园区项目", "Description": "华南园区项目", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "2153664000", "ValidityStartTime": "1680278400", "WorkspaceId": 1016}, {"ChineseName": "华北园区项目", "Description": "华北园区项目", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "2153664000", "ValidityStartTime": "1680278400", "WorkspaceId": 1018}, {"ChineseName": "test", "Description": "ceshi", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "1747584000", "ValidityStartTime": "1685289600", "WorkspaceId": 1263}, {"ChineseName": "report2test", "Description": "report2test", "EnglishName": "", "IsCommWorkspace": false, "Status": 0, "TenantId": 100055, "ValidityEndTime": "2192976000", "ValidityStartTime": "1687795200", "WorkspaceId": 1439}]

错误码

最近更新时间：2023-12-05 02:49:36

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。
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。

错误码	说明
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.ApiAuthorizationNotFound	未找到此应用该api授权信息
AuthFailure.TokenExpired	token已过期,请重新申请
AuthFailure.TokenNotFound	未找到该token信息
FailedOperation.ApiRuleConfigError	Api规则配置信息错误
FailedOperation.DBError	操作DB失败
FailedOperation.GetTagsFailed	获取标签错误
FailedOperation.LockRedisCacheFailed	锁定Redis缓存错误
FailedOperation.ModifyDeviceError	修改设备信息失败
FailedOperation.RedisOperationFailed	redis操作错误
FailedOperation.SendMsgError	消息发送错误
FailedOperation.SendMsgToIoTFailed	向IoT服务发送消息失败
FailedOperation.StorageInitFailed	对象存储 - 初始化失败
FailedOperation.URLGenerateFailed	对象存储 - url生成失败
InternalServerError.ApiGatewayInternalError	API网关内部错误
InternalServerError.ApiRequestPathParameterError	该api访问路径匹配参数异常，请检查请求路径
InternalServerError.AppApiNoSpacePermission	该应用未关联该项目空间数据，无法获取该项目空间数据

错误码	说明
InternalError.BIMAppInternalError	BIM服务内部错误
InternalError.BusinessLogicError	业务逻辑错误
InternalError.ManagerSrvFailed	manager服务操作失败
InternalError.Timeout	超时
InternalError.UnknownError	未知错误
InternalError.UpdateParkInfoFailed	更新工作空间下园区信息错误
InvalidParameter.DuplicateParkCode	园区编码重复
InvalidParameter.ExceedParkLengthLimit	园区名字或园区编码长度超过50字符
InvalidParameter.InvalidAlarmContent	告警content无效
InvalidParameter.InvalidMediaData	错误的媒体数据
InvalidParameter.InvalidMetaData	错误的Meta数据, 无法构建握手参数
InvalidParameter.InvalidRequestTime	请求时间非法
InvalidParameter.ParamNotMatch	参数不匹配
InvalidParameter.StatusNotMatchProcessType	状态与处理类型不符
InvalidParameter.TokenFieldNotFound	未找到token字段
InvalidParameterValue.EmptyAdministrativeCode	行政区编码长度为0
InvalidParameterValue.InvalidAlarmStatus	告警状态无效
InvalidParameterValue.InvalidApplicationId	应用id非法
InvalidParameterValue.InvalidElementIdOrProjectId	项目id或构件id取值错误
InvalidParameterValue.InvalidGB28181Config	网络视频录像机GB28181协议错误
InvalidParameterValue.InvalidNonce	申请应用token时, 传递的nonce参数非法
InvalidParameterValue.InvalidProtocol	非法protocol协议参数
InvalidParameterValue.InvalidRequestTime	请求时间非法
InvalidParameterValue.InvalidSaveType	保存类型参数错误
InvalidParameterValue.InvalidSignature	签名非法
InvalidParameterValue.InvalidStreamId	非法StreamId, StreamId仅支持0或1
InvalidParameterValue.InvalidTenantId	metadata租户信息无效
InvalidParameterValue.InvalidTimeRange	时间范围参数错误
InvalidParameterValue.InvalidVideoPlayRate	播放速率错误, 需要等于0.5、1、1.5、2、4、8或16
InvalidParameterValue.InvalidWorkspaceld	错误的工作空间Id
LimitExceeded.ApiLimitExceeded	api接口批量最大个数超限
LimitExceeded.DeviceLimitExceeded	设备数量超过限制
LimitExceeded.VideoStreamThresholdExceeded	视频流超过阈值
LimitExceeded.VideoTranscode	视频转码超出阈值
MissingParameter.EmptyAlarmId	告警id为空
MissingParameter.EmptyStream	stream参数缺失

错误码	说明
MissingParameter.EmptyWID	WID参数缺失
OperationDenied.ApiPermissionDenied	Api没有操作权限
ResourceNotFound.AlarmIDNotFound	告警id找不到
ResourceNotFound.AlarmIdNotExist	告警id不存在
ResourceNotFound.CameraBaseInfoNotExist	摄像头基础信息不存在
ResourceNotFound.CameraInfoNotExist	摄像头信息不存在
ResourceNotFound.CameraStatusNotExist	摄像头状态不存在
ResourceNotFound.ElementNotFound	未查询到构件信息
ResourceNotFound.EmptyDeviceList	空设备列表
ResourceNotFound.NVROrCVRConfigNotExist	网络视频录像机或中心级视频网络存储设备相关配置信息缺失
ResourceNotFound.SRSHookServiceNode	srs_hook服务节点缺失
ResourceNotFound.SRSNotExist	该deviceId的srs信息缺失
ResourceNotFound.StreamNotExist	视频流缺失
ResourceNotFound.VideoPushServiceNode	VideoPush服务节点缺失
ResourceNotFound.WIDNotExist	设备Wid不存在
ResourceUnavailable.DeviceOffline	设备已离线
UnauthorizedOperation.APIAuthFailed	wid 未找到或被应用api未被授权
UnauthorizedOperation.UnauthorizedApi	应用api的数据授权未配置，请先配置权限
UnsupportedOperation.NotStreaming	未推流
UnsupportedOperation.UnsupportedCMD	不支持的控制指令