

# 物联网开发平台

## 客户端 API



腾讯云

## 【版权声明】

©2013-2026 腾讯云版权所有

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

## 【商标声明】



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

## 【服务声明】

本文档意在向您介绍腾讯云全部或部分产品、服务的当时的相关概况，部分产品、服务的内容可能不时有所调整。您所购买的腾讯云产品、服务的种类、服务标准等应由您与腾讯云之间的商业合同约定，除非双方另有约定，否则，腾讯云对本文档内容不做任何明示或默示的承诺或保证。

## 【联系我们】

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

# 文档目录

## 客户端 API

### 应用端 API 简介

#### 用户管理

##### 用户注册登录

##### 用户注销

#### 配网管理

##### 生成 Wi-Fi 设备配网 Token

##### 查询配网 Token 状态

##### 用户绑定 Wi-Fi 设备

#### 设备管理

##### 查询微信授权票据

##### 获取产品信息

##### 用户删除设备

##### 获取用户绑定设备列表

##### 获取设备当前状态

##### 修改设备名称

##### 获取设备详情

##### 获取设备扩展信息

##### 设备更换房间

##### 添加子设备到用户绑定列表

##### 获取指定网关设备的子设备列表

##### 获取已绑定到家庭下的指定网关的子设备列表

##### 蓝牙上报数据

##### 蓝牙设备发送设备行为的回复消息

##### 蓝牙设备上报事件

##### 配网绑定设备

##### 绑定子设备到网关

##### 解绑子设备网关

#### 设备控制

##### 设备透传指令控制

##### 用户控制设备

##### 同步调用设备行为

##### 异步调用设备行为

#### 设备分享

##### App 端发送设备分享

- 获取设备分享 Token
- 获取设备分享 Token 信息
- 绑定用户分享的设备
- 查询用户分享设备列表
- 删除用户分享的设备
- 查询设备的用户列表
- 删除设备分享的用户
  
- 家庭管理**
- 创建家庭
- 删除家庭
- 修改家庭
- 获取家庭列表
- 获取家庭详情
- 新建房间
- 修改房间
- 删除房间
- 获取房间列表
- 邀请家庭成员
- 成员加入家庭
- 删除家庭成员
- 成员退出家庭
- 获取家庭成员列表
- App 端邀请家庭成员
  
- 设备定时**
- 新建定时任务
- 修改定时任务
- 删除定时任务
- 修改定时任务状态
- 获取定时任务列表
  
- 固件升级**
- 上报设备固件版本
- 查询设备可升级固件版本
- 查询设备固件下载地址
- 确认固件升级任务
- 上报固件升级任务状态
- 查询设备固件升级状态
- 查看固件信息
- 获取设备当前版本信息

## 消息管理

获取消息列表

删除消息

## 手动智能

创建手动智能联动

获取手动智能联动列表

删除手动智能联动

修改手动智能联动

执行手动智能联动

## 自动智能

创建自动智能联动

获取自动智能联动列表

获取自动智能联动详情

删除自动智能联动

修改自动智能联动

修改自动智能联动状态

## 数据查询

获取设备物模型数据

获取设备物模型历史数据

获取设备的历史事件

## 长连接通信

注册监听

心跳

设备状态推送

## 云存服务

拉取云存事件列表

拉取云存事件缩略图

获取具有云存的日期

获取某一天云存时间轴

获取视频防盗链播放 URL

拉取图片流数据

## 音乐服务

歌单歌曲

获取歌曲播放链接

每日推荐

批量查询歌曲信息

推荐歌单

新歌首发

[用户登出](#)  
[用户信息](#)  
[数据结构](#)

# 客户端 API

## 应用端 API 简介

最近更新时间：2025-12-29 15:26:13

### 简介

应用端 API 是开发平台为了满足智能家居场景，为用户开发自有品牌的小程序或 App 而提供的云端服务，用户无需实现用户管理、设备管理、设备定时、家庭管理等基础能力，可通过调用应用端 API 快速完成移动应用端的开发。

### 调用方式

#### 1. 登录前所使用的 API URL 为：

说明：

- 国内平台：<https://iot.cloud.tencent.com/api/exploreropen/appapi>
- 海外平台：<https://oversea.iotcloud.tencentiotcloud.com/api/exploreropen/appapi>

其中公共参数有：Action, RequestId, AppKey, Signature, Timestamp, Nonce。

- Action 用于标识请求的方法名称。
- RequestId 用于标识一个唯一请求，推荐使用 uuid 作为参数值，定位问题时建议提供该参数值。
- AppKey 为应用的密钥。
- Signature 为本次请求的签名，具体计算方法见本文下方示例。
- Timestamp 为本次请求的 UNIX 秒级时间戳。
- Nonce 为随机正整数，用于和时间戳一起，防范 API 重放攻击。

#### 2. 登录后所使用的 API URL 为：

说明：

- 国内平台：<https://iot.cloud.tencent.com/api/exploreropen/tokenapi>
- 海外平台：<https://oversea.iotcloud.tencentiotcloud.com/api/exploreropen/tokenapi>

其中公共参数有：Action, RequestId, AccessToken。

- Action 用于标识请求的方法名称。
- RequestId 用于标识一个唯一请求，推荐使用 uuid 作为参数值，定位问题时建议使用该参数值。
- AccessToken 用于标识一个已经登录的用户。

说明：

调用应用端 API 接口前，需要先调用登录相关接口（应用端 API/用户管理），登录成功后得到 AccessToken。

AccessToken 用于标识一个用户。当用户登录成功后，再携带 AccessToken 调用其他应用端 API。

登录前调用 URL 后缀为 `.../appapi`、登录后调用 URL 后缀为 `.../tokenapi`

## 签名算法

### 获取应用 AppKey 和 AppSecret

如果用户不使用腾讯官方的“腾讯连连”小程序，用户也可通过平台开放能力开发自有品牌小程序。在创建应用的时候，平台会为用户生成小程序对应的安全凭证。安全凭证包括 AppKey 和 AppSecret。AppKey 是用于标识 API 调用者身份，AppSecret 是用于加密签名字符串和服务器端验证签名字符串的密钥。

#### ⚠ 注意：

用户应严格保管其 AppSecret，避免泄露。

具体获取步骤如下：

- 1.1 登录 [物联网开发平台控制台](#)，进入开发中心。
- 1.2 选择左侧菜单[应用开发](#)，新建应用，具体新建步骤参见 [应用开发](#)。
- 1.3 创建小程序成功后，即可获取系统自动生成的 AppKey 与 AppSecret。

## 生成签名字串

有了安全凭证 AppKey 和 AppSecret 后，就可以生成签名字串了。下面给出了一个生成签名字串的详细过程。

假设用户的 AppKey 和 AppSecret 分别是：

- AppKey: ahPxdK\*\*\*\*\*TGrejd
- AppSecret: NcbHqk\*\*\*\*\*TCGbKnQH

#### ⚠ 说明：

本文仅为示例，请您根据自己实际的 AppKey 和 AppSecret 进行后续操作。

以通过手机号注册账号 `AppCreateCellphoneUser` 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	类型	描述	参考数值
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需要提供该次请求的 RequestId。	8b8d499bbba1ac28b6da21b4
Action	String	公共参数，调用的接口方法名称。	AppCreateCellphoneUser

AppKey	String	公共参数，应用 AppKey，用于标识对应的 App。	ahPxdK****TGrejd
Signature	String	公共参数，请求的签名。	Szxai9Qs7****OXa hbFbseZ+uE=
Timestamp	Int64	公共参数，当前的 UNIX 时间戳（秒级）。	1546315200
Nonce	Int	公共参数，随机正整数，与时间戳一起，用于 API 防重放。	71087795
CountryCode	String	国家区码。	86
PhoneNumber	String	手机号码。	13900000000
Password	String	密码。	password
VerificationCode	String	短信验证码。	123456

### 说明:

- 请求参数中的公共请求参数有：RequestId、Action、AppKey、Timestamp、Nonce、Signature。
- AppCreateCellphoneUser 接口特有参数：CountryCode、PhoneNumber、Password、VerificationCode。

而参数 Signature（签名串）正是由上述参数共同生成的，具体步骤如下：

#### 1. 对参数排序。

对所有请求参数按参数名做字典序升序排列，所谓字典序升序排列，直观上就如同在字典中排列单词一样排序，按照字母表或数字表里递增顺序的排列次序，即先考虑第一个“字母”，在相同的情况下考虑第二个“字母”，依此类推。您可以借助编程语言中的相关排序函数来实现这一功能，例如 PHP 中的 ksort 函数。上述示例参数的排序结果如下：

```
{  
    Action=AppCreateCellphoneUser,  
    AppKey=ahPxdK****TGrejd,  
    CountryCode=86,  
    Nonce=71087795,  
    Password=My!P@ssword,  
}
```

```
PhoneNumber=13900000000,  
RequestId=8b8d499bbba1ac28b6da21b4,  
Timestamp=1546315200,  
VerificationCode=123456  
}
```

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. 拼接请求字符串。

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, 例如对 Action 参数, 其参数名称为 "Action", 参数值为 "AppCreateCellphoneUser", 因此格式化后就为

Action=AppCreateCellphoneUser。

- “参数值”为原始值而非 URL 编码后的值。
- 若输入参数中的“键”包含下划线, 则需要将其转换为“.”。对于“值”不需要额外操作。

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=AppCreateCellphoneUser&AppKey=ahPxdK****TGr4jd&CountryCode=86&N  
once=71087795&Password=My!P@ssword  
&PhoneNumber=13900000000&RequestId=8b8d499bbba1ac28b6da21b4&Timestamp=  
1546315200&VerificationCode=123456
```

## 3. 生成签名单。

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, 然后将生成的签名单使用 Base64 进行编码, 即可获得最终的签名单。

具体代码如下, 以 PHP 语言为例:

```
$secretKey = 'NcbHqk****TCGbKnQH';  
$srcStr =  
'Action=AppCreateCellphoneUser&AppKey=ahPxdK****TGr4jd&CountryCode=86&  
Nonce=71087795&Password=My!P@ssword&PhoneNumber=13900000000&RequestId=  
8b8d499bbba1ac28b6da21b4&Timestamp=1546315200&VerificationCode=123456'  
;  
$signStr = base64_encode(hash_hmac('sha1', $srcStr, $secretKey,  
true));  
echo $signStr
```

最终得到的签名单为:

```
Szxai9Qs7O3lBoOXahbFbseZ+uE=
```

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。

## 常见问题

对于参数为字符串且为空的，应将其从签名串中省略，不需要将其参与签名。

# 用户管理

## 用户注册登录

最近更新时间：2025-07-28 12:03:22

### 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（`AppGetTokenByWeiXin`）用于用户注册登录，获取开发平台的用户访问 `Token`，首次调用时，自动为该用户注册对应账号。

### 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值： <code>AppGetTokenByWeiXin</code> 。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需提供该次请求的 <code>RequestId</code> 。
AppKey	String	是	公共参数，应用 <code>AppKey</code> ，用于标识对应的小程序或 <code>App</code> 。
Signature	String	是	公共参数，请求签名，需用户自行生成，用于校验请求的合法性。
Timestamp	Int64	是	公共参数，请求的 <code>UNIX</code> 时间戳（秒级）。
Nonce	Int	是	公共参数，随机正整数，与 <code>Timestamp</code> 联合起来，防止重放攻击。
WxOpenID	String	是	用户的 <code>OpenID</code> 或 <code>UnionID</code> 。
NickName	String	是	昵称。
Avatar	String	是	头像。

### 3. 输出参数

名称	类型	描述

RequestID	String	公共参数，唯一请求 ID，与入参相同。
ExpireAt	Int64	截止时间，UNIX 秒级时间戳。
Token	String	开发平台返回的 AccessToken，通过该 Token 进行登录后的接口请求。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/appapi HTTP/1.1
content-type: application/json
{
  "WxOpenID": "0123456789",
  "NickName": "tests",
  "Avatar": "http://example.com/path/to/avatar.jpg",
  "Signature": "IY0yHX7zWCZKZ97xDonQoy16W4U=",
  "Timestamp": 1552621825,
  "Nonce": 2,
  "Action": "AppGetTokenByWeiXin",
  "AppKey": "ahPxdK*****TGrejd",
  "RequestId": "rest-client"
}
```

### 输出示例：成功

```
{
  "Response": {
    "Data": {
      "ExpireAt": 1556076201,
      "Token": "d2*****8514"
    },
    "RequestId": "rest-client"
  }
}
```

### 输出示例：失败

```
{
```

```
"Response": {  
    "Error": {  
        "Code": "InvalidParameterValue.ErrorUserNotExists",  
        "Message": "账号不存在"  
    },  
    "RequestId": "rest-client"  
}  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
ErrorRequiredParamNotFound	必选参数缺失。
InvalidAction	Action 非法。
InvalidParameterValue	参数异常。
InvalidParameterValue.InvalidJSON	请求格式不是 JSON。
InvalidParameterValue.NickNameLengthInvalid	昵称长度非法

# 用户注销

最近更新时间：2025-12-19 09:14:51

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppLogoutUser`）用于用户注销。

## 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值： <code>AppLogoutUser</code> 。
AccessToken	String	是	公共参数， <code>AccessToken</code> 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需提供该次请求的 <code>RequestId</code> 。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 <code>RequestId</code> 。
Data	String	响应结果。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "AccessToken": "bf*****d098",
  "Action": "AppLogoutUser",
  "RequestId": "rest-client"
}
```

### 输出示例：成功

```
{  
  "Response": {  
    "Data": "OK",  
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"  
  }  
}
```

### 输出示例：失败

```
{  
  "Response": {  
    "Error": {  
      "Code": "InvalidParameterValue.InvalidAccessToken",  
      "Message": "Token无效"  
    },  
    "RequestId": ""  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
ErrorRequiredParamNotFound	必选参数缺失。
InvalidAction	Action 非法。
InvalidParameterValue	参数非法。
InvalidParameterValue.InvalidJSON	请求格式不是 JSON。
InvalidParameterValue.InvalidAccessToken	Token 无效

# 配网管理

## 生成 Wi-Fi 设备配网 Token

最近更新时间：2025-12-29 15:26:13

### 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)

本接口（AppCreateDeviceBindToken）用于生成 Wi-Fi 配网任务的随机 Token。

### 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值： AppCreateDeviceBindToken。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需要提供该次请求的 RequestId。
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
TokenLengthType	String	可选	默认为空，Token 长度为32字符，若传1，Token 长度为6位字符。

### 3. 输出参数

名称	类型	描述
RequestId	String	RequestId。
Token	String	生成的配网 Token。

### 4. 示例

#### 示例1 生成 Wi-Fi 设备配网 Token

##### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
```

```
{  
  "Action": "AppCreateDeviceBindToken",  
  "AccessToken": "bf*****d098",  
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"  
}
```

### 输出示例：成功

```
{  
  "Response": {  
    "Data": {  
      "Token": "d7*****1b3b",  
      "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"  
    }  
  }  
}
```

### 输出示例：失败

```
{  
  "Response": {  
    "Error": {  
      "Code": "InvalidParameterValue.InvalidAccessToken",  
      "Message": "Token无效"  
    },  
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"  
  }  
}
```

## 5. 错误码

错误码	描述
InvalidParameter	无效参数。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TokenInfoSaveError	Token 保存失败。



# 查询配网 Token 状态

最近更新时间: 2024-09-30 17:54:51

## 1. 接口描述

接口请求域名: `iot.cloud.tencent.com/api/exploreropen/tokenapi`

本接口 (AppGetDeviceBindTokenState) 用于查询配网 Token 的当前状态。

## 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数, 本接口取值: AppGetDeviceBindTokenState。
RequestId	String	是	公共参数, 唯一请求 ID, 可自行生成, 推荐使用 uuid。定位问题时, 需提供该次请求的 RequestId。
AccessToken	String	是	公共参数, 用户通过微信号、手机或邮箱账号登录成功后, 获取的访问 Token。
Token	String	是	由 AppCreateDeviceBindToken 接口生成的配网 Token。

## 3. 输出参数

名称	类型	描述
RequestId	String	RequestId。
State	Uint	Token 状态。 <ul style="list-style-type: none"><li>1: 初始生产。</li><li>2: 可使用状态。</li></ul>

## 4. 示例

### 示例1 查询配网 Token 状态

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
```

```
{  
  "Action": "AppCreateDeviceBindToken",  
  "AccessToken": "bf*****d098",  
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",  
  "Token": "d7a1*****a861b3b"  
}
```

### 输出示例：成功

```
{  
  "Response": {  
    "Data": {  
      "State": 1,  
      "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"  
    }  
  }  
}
```

### 输出示例：失败

```
{  
  "Response": {  
    "Error": {  
      "Code": "InvalidParameterValue.TokenNotExist",  
      "Message": "Token不存在"  
    },  
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"  
  }  
}
```

## 5. 错误码

错误码	描述
InvalidParameter	无效参数。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TokenNotExist	Token 不存在。

InvalidParameterValue.TokenIsExpire	Token 已过期。
InvalidParameterValue.ReadTokenInfoError	Token 读取错误。

# 用户绑定 Wi-Fi 设备

最近更新时间：2024-10-11 18:02:12

## 1. 接口描述

接口请求域名： `iot.cloud.tencent.com/api/exploreropen/tokenapi`

本接口（AppTokenBindDeviceFamily）用于小程序或 App 用户绑定 Wi-Fi 类设备。

## 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值： <code>AppTokenBindDeviceFamily</code> 。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需要提供该次请求的 <code>RequestId</code> 。
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
Token	String	是	配网 Token。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
FamilyId	String	是	家庭 ID。
RoomId	String	是	房间 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需提供该次请求的 <code>RequestId</code> 。

## 4. 示例

### 用户绑定 Wi-Fi 设备

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Action": "AppTokenBindDeviceFamily",
  "AccessToken": "tesetoken",
  "Token": "bf*****d098",
  "ProductId": "*****",
  "DeviceName": "*****",
  "FamilyId": "f*****cc1",
  "RoomId": "0"
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"
}
```

## 输出示例 成功

```
{
  "Response": {
    "Data": {
      "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"
    }
  }
}
```

## 输出示例 失败

```
{
  "Response": {
    "Error": {
      "Code": "InvalidParameterValue.InvalidAccessToken",
      "Message": "Token无效"
    },
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"
  }
}
```

## 5. 错误码

错误码	描述
-----	----

InvalidParameter	无效参数。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TokenNotExist	Token 不存在。
InvalidParameterValue.TokenIsExpired	Token 已过期。
InvalidParameterValue.ReadTokenInfoError	Token 读错误。
InvalidParameterValue.TokenNotBind	Token 未绑定，即在上一步接口 AppGetDeviceBindTokenState 查询返回状态为初始状态，非可使用状态。
InvalidParameterValue.FamilyDeviceCountReadError	读设备数量错误。
UnauthorizedOperation	无操作权限。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。
UnauthorizedOperation.NoPermissionToFamily	操作该家庭无权限。
ResourceNotFound	资源不存在。
ResourceNotFound.StudioFamilyNotExist	家庭未创建或是已删除。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
LimitExceeded	数量限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制。
InternalError	内部错误。
InternalError.InternalServerException	内部错误。

InternalError.InternalServerException	内部 DB 错误。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.CannotBindSameFamily	该设备已经绑定，请勿重复绑定

# 设备管理

## 查询微信授权票据

最近更新时间：2024-06-14 10:35:11

### 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)

本接口（APPGetWechatDeviceTicket）用于获取微信设备票据。

### 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值：APPGetWechatDeviceTicket。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需要提供该次请求的 RequestId。
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
IsThirdApp	Int	是	是否是第三方小程序： 1=是。
ModelId	String	否	模板 ID
MiniProgramAppId	String	否	小程序 APPID

### 3. 输出参数

名称	类型	描述
RequestId	String	RequestId。

Data	WXDeviceInfo	生成的微信设备票据信息。
------	--------------	--------------

## 4. 示例

### 示例1 获取微信设备票据

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Action": "APPGetWechatDeviceTicket",
  "AccessToken": "bf*****d098",
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
  "ProductId": "PRODUCT_ID",
  "Devicename": "DEVICE_NAME",
  "IsThirdApp": 1
}
```

#### 输出示例：成功

```
{
  "Response": {
    "Data": {
      "DeviceId": "xxx",
      "WXIoTDeviceInfo": {
        "SN": "xxx",
        "SNTicket": "xxx",
        "ModelId": "xxx"
      },
      "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"
    }
  }
}
```

#### 输出示例：失败

```
{
  "Response": {
```

```
    "Error": {  
        "Code": "InvalidParameterValue.InvalidAccessToken",  
        "Message": "Token无效"  
    },  
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"  
}  
}
```

## 5. 错误码

错误码	描述
InvalidParameter	无效参数。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TokenInfoSaveError	Token 保存失败。

# 获取产品信息

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetProducts）获取产品信息。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetProducts。
ProductIds	Array of String	是	产品 ID 数组。
DeviceIds	Array of String	否	设备 ID 数组。

## 3. 输出参数

名称	类型	描述
Products	Array	返回产品信息。
RequestId	String	公共参数，唯一请求 ID，与入参相同。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
```

```
"RequestId": "27HPvi1ylv",
"Action": "AppGetProducts",
"ProductIds": [ "FVYYYEL4ON" ],
"DeviceIds": [ "QDA1PZLBNB/aaa", "QDA1PZLBNB/bbb" ]
}
```

## 输出示例：成功

```
{
  "Response": {
    "Products": [
      {
        "ProductId": "FVYYYEL4ON",
        "Name": "c1",
        "Description": "yes",
        "State": "",
        "DataTemplate": {
          "version": "1.0",
          "profile": {
            "ProductId": "FVYYYEL4ON",
            "CategoryId": "1",
            "properties": [
              {
                "id": "mac",
                "name": "mac地址",
                "desc": "wifi路由器mac地址",
                "required": true,
                "mode": "r",
                "define": {
                  "type": "string",
                  "min": "0",
                  "max": "30"
                }
              },
              {
                "id": "signal",
                "name": "wifi强度",
                "desc": "wifi信号强度值",
                "mode": "r",
                "define": {
                  "type": "int",
                  "unit": "dbm",
                  "step": "1",
                  "min": "-180",
                  "max": "180",
                  "start": "1"
                }
              },
              {
                "id": "ssid",
                "name": "wifi热点名",
                "desc": "设备连接的wifi热点名",
                "mode": "r",
                "define": {
                  "type": "string",
                  "min": "0",
                  "max": "64"
                }
              },
              {
                "id": "severip",
                "name": "wifi路由器接入的ip",
                "desc": "wifi路由器接入的ip",
                "mode": "r",
                "define": {
                  "type": "string",
                  "min": "0",
                  "max": "64"
                }
              },
              {
                "id": "nmacs",
                "name": "其他热点mac地址",
                "desc": "可接受到的其他热点mac地址",
                "mode": "r",
                "define": {
                  "type": "string",
                  "min": "0",
                  "max": "64"
                }
              }
            ],
            "events": [],
            "actions": []
          },
          "AppTemplate": "",
          "NetType": "wifi",
          "CategoryId": 1,
          "ProductType": 0,
          "UpdateTime": 1600159203
        }
      }
    ]
  }
}
```

```
        "Name": "light5勿删",
        "Description": "",
        "State": "",
        "DataTemplate": {"version": "1.0", "profile": {
            "ProductId": "QDA1PZLBNB", "CategoryId": "141"}, "properties": [
                {"id": "power_switch", "name": "电灯开关", "desc": "控制电灯开
灭", "required": true, "mode": "rw", "define": {"type": "bool", "mapping": {
                    "0": "关", "1": "开"}}, {"id": "color", "name": "颜色", "desc": "灯光颜
色", "mode": "rw", "define": {"type": "enum", "mapping": {
                    "0": "Red", "1": "Green", "2": "Blue"}}, {"id": "brightness", "name": "亮
度", "desc": "灯光亮度", "mode": "rw", "define": {"type": "int", "unit": "%", "step": "1", "min": "0", "max": "100", "start": "1"}, {"id": "name", "name": "灯位置名称", "desc": "灯位置名称: 书房、客厅
等", "mode": "rw", "required": false, "define": {"type": "string", "min": "0", "max": "64"}}, {"id": "Temperature", "name": "温
度", "desc": "", "mode": "rw", "define": {"type": "float", "min": "0", "max": "100", "start": "0", "step": "1", "unit": ""}, "required": false}, {"id": "temp_max", "name": "最高温
度", "desc": "", "mode": "rw", "define": {"type": "int", "min": "0", "max": "100", "start": "0", "step": "1", "unit": ""}, "r
equired": false}, {"id": "LBS_BS", "name": "LBS_BS", "desc": "", "mode": "rw", "define": {"type": "string", "min": "0", "max": "2048"}, "required": false}, {"id": "FOO", "name": "FOO", "desc": "", "mode": "rw", "define": {"type": "string", "min": "0", "max": "2048"}, "required": false}, {"id": "GPS_Ext", "name": "GPS_Ext", "desc": "", "mode": "rw", "define": {"type": "string", "min": "0", "max": "2048"}, "required": false}], "events": [
            {"id": "status_report", "name": "DeviceStatus", "desc": "Report the device
status", "type": "info", "required": false, "params": [
                {"id": "status", "name": "running_state", "desc": "Report current device
running state", "define": {"type": "bool", "mapping": {
                    "0": "normal", "1": "fault"}}, {"id": "message", "name": "Message", "desc": "Some extra message", "define": {"type": "string", "min": "0", "max": "64"}}, {"id": "low_voltage", "name": "LowVoltage", "desc": "Alert for device voltage
is low", "type": "alert", "required": false, "params": [
                    {"id": "voltage", "name": "Voltage", "desc": "Current voltage", "define": {"type": "float", "unit": "V", "step": "1", "min": "0.0", "max": "24.0", "start": "1"}], {"id": "hardware_fault", "name": "Hardware_fault", "desc": "Report
hardware fault", "type": "fault", "required": false, "params": [
                    {"id": "fault", "name": "Fault", "desc": "Fault details", "define": {"type": "string", "min": "0", "max": "1024"}]}]}]}]}]
```

```
[{"id": "name", "name": "Name", "desc": "Name like: memory, tf card, censors", "define": {"type": "string", "min": "0", "max": "64"}}, {"id": "error_code", "name": "Error_Code", "desc": "Error code for fault", "define": {"type": "int", "unit": "", "step": "1", "min": "0", "max": "2000", "start": "1"}}], "actions": []}, {"AppTemplate": "", "NetType": "else", "CategoryId": 141, "ProductType": 0, "UpdateTime": 1600265188}, {"}, {" "ProductId": "QDA1PZLBNB", "Name": "light5勿删", "Description": "", "State": "", "DataTemplate": {"version": "1.0", "profile": {"ProductId": "QDA1PZLBNB", "CategoryId": "141"}, "properties": [{"id": "power_switch", "name": "电灯开关", "desc": "控制电灯开灭", "required": true, "mode": "rw", "define": {"type": "bool", "mapping": {"0": "关", "1": "开"}}, {"id": "color", "name": "颜色", "desc": "灯光颜色", "mode": "rw", "define": {"type": "enum", "mapping": {"0": "Red", "1": "Green", "2": "Blue"}}, {"id": "brightness", "name": "亮度", "desc": "灯光亮度", "mode": "rw", "define": {"type": "int", "unit": "%", "step": "1", "min": "0", "max": "100", "start": "1"}}, {"id": "name", "name": "灯位置名称", "desc": "灯位置名称: 书房、客厅等", "mode": "rw", "required": false, "define": {"type": "string", "min": "0", "max": "64"}}, {"id": "Temperature", "name": "温度", "desc": "", "mode": "rw", "define": {"type": "float", "min": "0", "max": "100", "start": "0", "step": "1", "unit": ""}, "required": false}, {"id": "temp_max", "name": "最高温度", "desc": "", "mode": "rw", "define": {"type": "int", "min": "0", "max": "100", "start": "0", "step": "1", "unit": ""}, "required": false}, {"id": "LBS_BS", "name": "LBS_BS", "desc": "", "mode": "rw", "define": {"type": "string", "min": "0", "max": "2048"}, "required": false}, {"id": "FOO", "name": "FOO", "desc": "", "mode": "rw", "define": {"type": "string", "min": "0", "max": "2048"}, "required": false}, {"id": "GPS_Ext", "name": "GPS_Ext", "desc": "", "mode": "rw", "define": {"type": "string", "min": "0", "max": "2048"}]}]}]
```

```
{"type":"string","min":"0","max":"2048"},"required":false}], "events": [{"id": "status_report", "name": "DeviceStatus", "desc": "Report the device status", "type": "info", "required": false, "params": [{"id": "status", "name": "running_state", "desc": "Report current device running state", "define": {"type": "bool", "mapping": {"0": "normal", "1": "fault"}}, {"id": "message", "name": "Message", "desc": "Some extra message", "define": {"type": "string", "min": "0", "max": "64"}}], {"id": "low_voltage", "name": "LowVoltage", "desc": "Alert for device voltage is low", "type": "alert", "required": false, "params": [{"id": "voltage", "name": "Voltage", "desc": "Current voltage", "define": {"type": "float", "unit": "V", "step": "1", "min": "0.0", "max": "24.0", "start": "1"}}], {"id": "hardware_fault", "name": "Hardware_fault", "desc": "Report hardware fault", "type": "fault", "required": false, "params": [{"id": "name", "name": "Name", "desc": "Name like: memory, tf card, censors ...", "define": {"type": "string", "min": "0", "max": "64"}}, {"id": "error_code", "name": "Error_Code", "desc": "Error code for fault", "define": {"type": "int", "unit": "", "step": "1", "min": "0", "max": "2000", "start": "1"}]}], "actions": []}, {"AppTemplate": "", "NetType": "else", "CategoryId": 141, "ProductType": 0, "UpdateTime": 1600265188}], [{"RequestId": "27HPvi1ylv"}]}]
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。



# 用户删除设备

最近更新时间：2023-07-17 18:02:12

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppDeleteDeviceInFamily`）为用户提供删除设备的功能，删除后用户需要重新配网进行绑定。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppDeleteDeviceInFamily</code> 。
FamilyId	String	是	家庭 ID，成功创建家庭后，返回的 FamilyId。
RoomId	String	是	房间 ID。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
}
```

```
"RequestId": "42a96dbf-bdca-48f0-8f44-e36146d44e17",
"Action": "AppDeleteDeviceInFamily",
"AccessToken": "bf*****d098",
"FamilyId": "1",
"ProductId": "R320NVL0EU",
"DeviceName": "df2eSJyY"
}
```

## 输出示例 成功

```
{
  "Response": {
    "RequestId": "42a96dbf-bdca-48f0-8f44-e36146d44e17"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccess Token	Token 无效。

# 获取用户绑定设备列表

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetFamilyDeviceList）用于获取用户已绑定设备列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetFamilyDeviceList。
FamilyId	String	是	家庭 ID。
RoomId	String	是	房间 ID。
Offset	Int	否	消息偏移量。
Limit	Int	否	最大返回消息条数，最大值为50。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Total	Int	设备总数。
DeviceList	Array	设备列表，请参见 <a href="#">DeviceList</a> 。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
    "RequestId": "req_1",
    "Action": "AppGetFamilyDeviceList",
    "FamilyId": "e0e23a2b33a24652b6*****a1cf",
    "RoomId": "5", //如果要获取家庭所有设备，则该字段为"”，"0" 则为默认房间
    "Offset": 0,
    "Limit": 2,
    "AccessToken": "bf*****d098"
}
```

## 输出实例

```
{
    "Response": {
        "DeviceList": [
            {
                "ProductId": "R32*****EU",
                "DeviceName": "df2eSJyY",
                "DeviceId": "R32****0EU/df****yY",
                "AliasName": "df2eSJyY",
                "UserID": "1",
                "RoomId": "r_4b27c753ef774d458e*****86ad",
                "IconUrl": "",
                "CreateTime": 1574664969,
                "UpdateTime": 1574668779
            }
        ],
        "RequestId": "req_1",
        "Total": 1
    }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误
InvalidParameterValue	参数取值错误

InvalidParameterValue.InvalidAccessToken

Token 无效

# 获取设备当前状态

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetDeviceStatuses）用于查询设备状态。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetDeviceStatuses。
DeviceIds	Array of String	是	设备 ID 数组。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Devices	Array of Devices	返回设备。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "892497d8-a733-480b-91d5-5559c6b49551",
  "Action": "AppGetDeviceStatuses",
  "DeviceIds": ["HY4DHFM5P6/81386276"],
  "AccessToken": "6bf4003d7f*****400f3186f"
```

}

## 输出示例：成功

```
{  
  "Response": {  
    "DeviceStatuses": [ {  
      "DeviceId": "22F9Y6II70/light1",  
      "ProductId": "22F9Y6II70",  
      "DeviceName": "light1",  
      "Online": 0 //0 在线; 1: 离线  
    } ],  
    "RequestId": "892497d8-a733-480b-91d5-5559c6b49551"  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 修改设备名称

最近更新时间：2023-07-17 18:02:12

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppUpdateDeviceInFamily`）用于修改设备名称。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppUpdateDeviceInFamily</code> 。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
AliasName	String	是	设备别名。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
DeviceId	String	设备 ID。
ProductId	String	产品 ID。
DeviceName	String	设备名称。
AliasName	String	设备别名。

IconUrl	String	图标 URL。
FamilyId	String	家庭 ID。
RoomId	String	房间 ID。
CreateTime	Int64	创建时间。
UpdateTime	Int64	更新时间。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "641150a3-4238-430d-888b-c01b11298351",
  "Action": "AppUpdateDeviceInFamily",
  "AccessToken": "xxxxv2",
  "ProductId": "VZS14OZSN5",
  "DeviceName": "~virtualDev",
  "AliasName": "12345"
}
```

### 输出示例：成功

```
{
  "Response": {
    "Data": {
      "DeviceId": "R32ONVL0EU/df2eSJyY",
      "ProductId": "R32ONVL0EU",
      "DeviceName": "df2eSJyY",
      "AliasName": "12345",
      "IconUrl": "",
      "FamilyId": "",
      "RoomId": 0,
      "CreateTime": 1574859101,
      "UpdateTime": 1574859144
    }
  }
}
```

```
    },
    "RequestId": "641150a3-4238-430d-888b-c01b11298351"
}
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 获取设备详情

最近更新时间：2023-07-17 18:26:01

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetDeviceInFamily）用于查询设备详情。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetDeviceInFamily。
ProductId	String	是	产品 ID。
FamilyId	String	是	家庭 ID。
DeviceName	String	是	设备名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
DeviceId	String	设备 ID，是产品 ID/设备名称组合。
ProductId	String	产品 ID。
DeviceName	String	设备名称。
AliasName	String	设备别名。

IconUrl	String	图标 URL。
FamilyId	String	家庭 ID。
RoomId	String	房间 ID。
CreateTime	Int64	创建时间, UNIX 秒级时间戳。
UpdateTime	Int64	最后一次更新时间, UNIX 秒级时间戳。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
    "RequestId": "082b70e-3d05-4179-9c94-6c70082e4a7",
    "Action": "AppGetDeviceInFamily",
    "ProductId": "R32ONVL0EU",
    "FamilyId": "xxx",
    "DeviceName": "df2eSJyY",
    "AccessToken": "xxxxv2"
}
```

### 输出示例: 成功

```
{
    "Response": {
        "Data": {
            "DeviceId": "R32ONVL0EU/df2eSJyY",
            "ProductId": "R32ONVL0EU",
            "DeviceName": "df2eSJyY",
            "AliasName": "12345",
            "IconUrl": "",
            "FamilyId": "",
            "RoomId": "",
            "DeviceType": 0, // 0 普通 1 网关设备 2 子设备
            "CreateTime": 1574931773,
            "LastUpdate": 1574931773
        }
    }
}
```

```
        "UpdateTime": 1574931945
    },
    "RequestId": "082b70e-3d05-4179-9c94-6c70082e4a7"
}
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。
ResourceNotFound	资源不存在。
ResourceNotFound.StudioMemberNotExist	家庭成员未创建或是已删除。

# 获取设备扩展信息

最近更新时间：2024-09-29 18:26:01

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetDeviceExtInfo）用于获取设备的扩展信息。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetDeviceExtInfo。
DeviceId	String	是	设备 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
ManufacturerName	String	厂家名称。
ProductModel	String	产品型号。
Mac	String	设备上报基础信息 中的 MAC 地址。
FwVer	String	设备上报基础信息 中的固件版本。
IP	String	设备的 IP 地址。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b",
  "Action": "AppGetDeviceExtInfo",
  "DeviceId": "PR*****ID/de***01",
  "AccessToken": "871d2d42*****692afae0"
}
```

### 输出示例: 成功

```
{
  "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b",
  "ManufacturerName": "厂家名称",
  "ProductModel": "产品型号",
  "Mac": "AABBCCDDEEFF",
  "FwVer": "1.2.0",
  "IP": ""
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 设备更换房间

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppModifyFamilyDeviceRoom）设备更换房间。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： AppModifyFamilyDeviceRoom。
FamilyId	String	是	家庭 ID。
RoomId	String	是	房间 ID。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
  "RequestId": "8c012671-3934-4dc2-b07c-82c5ba3766f2",
  "Action": "AppModifyFamilyDeviceRoom",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "FamilyId": "1",
  "RoomId": "r_4b27c753ef774d458e*****86ad",
  "ProductId": "R32****0EU",
  "DeviceName": "df2eSJyY"
}
```

### 输出示例：成功

```
{
  "Response": {
    "RequestId": "8c012671-3934-4dc2-b07c-82*****766f2"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 添加子设备到用户绑定列表

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreroopen/tokenapi`。

本接口（AppBindSubDeviceInFamily）用于将已经绑定到网关设备的子设备绑定到家庭。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppBindSubDeviceInFamily</code> 。
GatewayProductId	String	是	网关产品 ID。
GatewayDeviceName	String	是	网关设备名称。
ProductId	String	是	子产品 ID。
DeviceName	String	是	子设备名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
AppDeviceInfo	Object of <a href="#">AppDeviceInfo</a>	返回结果。

## 4. 示例

### 示例1

## 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
  "RequestId": "keyi20191119-012345",
  "Action": "AppBindSubDeviceInFamily",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "ProductId": "LAEG4YJE1A",
  "DeviceName": "subdev2",
  "GatewayProductId": "NJ270VLZT4",
  "GatewayDeviceName": "gwdev"
}
```

## 输出示例: 成功

```
{
  "Response": {
    "Data": [
      {
        "AppDeviceInfo": {
          "DeviceId": "R32ONVL0EU/df2eSJyY",
          "ProductId": "R32ONVL0EU",
          "DeviceName": "df2eSJyY",
          "AliasName": "",
          "IconUrl": "",
          "FamilyId": "1",
          "RoomId": "0",
          "CreateTime": 1574152773,
          "UpdateTime": 1574152773
        }
      },
      {
        "RequestId": "keyi20191119-012345"
      }
    ]
  }
}
```

## 5. 错误码

错误码	描述
-----	----

InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 获取指定网关设备的子设备列表

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)。

本接口（AppGetGatewayBindDeviceList）用于获取指定网关设备的子设备列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetGatewayBindDeviceList。
ProductId	String	是	子设备的 ProductId。
GatewayProductId	String	是	网关设备的 ProductId。
GatewayDeviceName	String	是	网关设备的 DeviceName。
Offset	Int	是	消息偏移量。
Limit	Int	是	最大返回消息条数，最大值为50。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Total	String	总数量。
DeviceList	Array of DeviceList	设备列表。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppGetGatewayBindDeviceList",
  "ProductId": "LAEG4YJE1A", //子设备的productId
  "GatewayProductId": "NJ270VLZT4", //网关设备的productId
  "GatewayDeviceName": "gwdev", //网关设备的deviceName
  "Offset": 0,
  "Limit": 2,
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec"
}
```

### 输出示例

```
{
  "Response": {
    "DeviceList": [
      {
        "ProductId": "LAEG4YJE1A",
        "DeviceName": "subdev2", //子设备的deviceName
        "DeviceId": "LAEG4YJE1A/subdev2", //子设备的deviceId
        "AliasName": "",
        "IconUrl": "",
        "BindStatus": 0 //未绑定到家庭
      },
      {
        "ProductId": "LAEG4YJE1A",
        "DeviceName": "subdev1",
        "DeviceId": "LAEG4YJE1A/subdev1",
        "AliasName": "",
        "IconUrl": "",
        "BindStatus": 1 //已经绑定到家庭
      }
    ],
    "RequestId": "req_1",
    "Total": 2
  }
}
```

}

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalRPCError	调用超时。
InternalError.InternalRPCError	调用返回失败。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.NotSupportVirtualDevice	不支持虚拟设备。
UnsupportedOperation.DeviceNotGateway	不是网关设备。
ResourceNotFound	资源不存在。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.StudioMemberNotExist	家庭成员未创建或是已删除。

# 获取已绑定到家庭下的指定网关的子设备列表

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)。

本接口（AppGetFamilySubDeviceList）用于获取家庭下网关的子设备列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用uuld。定位问题时，需提供该次请求的RequestId。
Action	String	是	公共参数，本接口取值：AppGetFamilySubDeviceList。
GatewayProductId	String	是	网关设备产品 ID。
GatewayDeviceName	String	是	网关设备 Name。
Offset	Int	是	消息偏移量。
Limit	Int	是	最大返回消息条数，最大值为50。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用uuld。定位问题时，需提供该次请求的RequestId。
Total	String	总数量。
DeviceList	Array of <a href="#">DeviceList</a>	设备列表。

## 4. 示例

## 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
    "RequestId": "req_1",
    "Action": "AppGetFamilySubDeviceList",
    "GatewayProductId": "NJ270VLZT4",
    "GatewayDeviceName": "gwdev",
    "Offset": 0,
    "Limit": 2,
    "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec"
}
```

## 输出示例

```
{
    "Response": {
        "DeviceList": [
            {
                "FamilyId": "f_9b309d84c962****0a11b4c3d9588fcc1",
                "ProductId": "LAEG4YJE1A",
                "DeviceName": "subdev1",
                "DeviceId": "LAEG4YJE1A/subdev1",
                "AliasName": "",
                "RoomId": "0",
                "IconUrl": "",
                "DeviceType": 2,
                "CreateTime": 1583492992,
                "UpdateTime": 1583492992
            }
        ],
        "RequestId": "req_1",
        "Total": 1
    }
}
```

## 5. 错误码

错误码	描述
-----	----

InternalError	内部错误。
InternalError.InternalRPCError	调用超时。
InternalError.InternalRPCError	调用返回失败。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.ERR_APP_INVALID_IOTAPPID	App 无效 lotAppID。
ResourceNotFound	资源不存在
ResourceNotFound.DeviceNotExist	设备未创建或是已删除

# 蓝牙上报数据

最近更新时间：2024-10-11 18:26:01

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppReportDataAsDevice）用于小程序或App进行数据上报。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppReportDataAsDevice。
DeviceId	String	是	设备 ID。
Data	String	是	修改的物模型数据。
TimeStamp	Int64	是	Unix 毫秒级时间戳。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Data	String	返回的数据信息。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "123e4567-e89b-12d3-a456-426614174000",
```

```
"Action": "AppReportDataAsDevice",
"DeviceId": "22*****0/light1",
"Data": "{\"light_switch\":0}",
"TimeStamp": 1581422682000,
"AccessToken": "8b4a70dd16*****2826868b18edd4e78a3bb8ec"
}
```

### 输出示例：成功

```
{
  "Response": {
    "Data": "",
    "RequestId": "123e4567-e89b-12d3-a456-426614174000"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.BindDeviceNotConnected	设备近期没有连接到云。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 蓝牙设备发送设备行为的回复消息

最近更新时间：2024-10-11 18:26:02

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppPublishMsgAsDevice）提供小程序或App模拟设备行为的回复消息功能。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppPublishMsgAsDevice。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
Topic	String	是	上报的 Topic。
Payload	String	是	消息内容。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "123e4567-e89b-12d3-a456-426614174000",
```

```
"Action": "AppPublishMsgAsDevice",
"ProductId": "22*****0",
"DeviceName": "light1",
"Topic": "$thing/up/action/22*****0/light1"
"Payload": "e1wibGlnaHRfc3dpdGNoXCI6MH0="
"AccessToken": "8b4a70dd16*****2826868b18edd4e78a3bb8ec"
}
```

**输出示例: 成功**

```
{
  "Response": {
    "RequestId": "123e4567-e89b-12d3-a456-426614174000"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.BindDeviceNotConnected	设备近期没有连接到云。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 蓝牙设备上报事件

最近更新时间：2024-10-11 18:26:02

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppReportDeviceEvent）用于蓝牙设备通过 App 或小程序上报事件。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppReportDeviceEvent。
DeviceId	String	是	设备 ID，由 ProductId/DeviceName 拼接组成。
EventId	String	是	事件 ID。
Params	String	是	JSON 格式的 Event 参数。
Method	String	是	上报方法，可以是 ReportEventAsDevice、reported（默认）。
EventTimeStamp	Int64	是	Unix 毫秒级时间戳。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
```

```
content-type: application/json
{
  "RequestId": "123e4567-e89b-12d3-a456-426614174000",
  "Action": "AppReportDeviceEvent",
  "DeviceId": "QD*****BNB/dev01",
  "EventId": "low_voltage",
  "Params": "{\"voltage\":3.1}",
  "Method": "ReportEventAsDevice",
  "AccessToken": "ningtoken*****"
}
```

### 输出示例：成功

```
{
  "Response": {
    "Data": "",
    "RequestId": "123e4567-e89b-12d3-a456-426614174000"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.BindDeviceNotConnected	设备近期没有连接到云。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 配网绑定设备

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppSigBindDeviceInFamily`）用于小程序或 App 进行 Wi-Fi 设备配网绑定操作。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppSigBindDeviceInFamily</code> 。
FamilyId	String	是	家庭 ID。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
RoomId	String	否	房间 ID。
DeviceTimeStamp	Int64	是	设备时间戳，Unix 秒级时间戳。
ConnId	String	否	随机字符串，建议5个字节长度。
Signature	String	是	动态签名，由设备根据配网协议传输到小程序、App 端的签名。计算示例见下文。
BindType	String	是	绑定类型。 <ul style="list-style-type: none"><li>wifi_sign: WIFI 绑定（默认）。</li><li>bluetooth_sign: 蓝牙绑定。</li><li>other_sign: 其他。</li></ul>
SignMethod	String	否	签名算法。 <ul style="list-style-type: none"><li>hmacsha1: HMACSHA1 加密算法（默认）。</li></ul>

- hmacsha256: HMACSHA256 加密算法。

### 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
AppDeviceInfo	Object of <a href="#">AppDeviceInfo</a>	设备信息。

### 4. 动态签名计算

签名计算步骤：

1. 确认是WIFI配网还是蓝牙或者其他配网，需要加密的明文串拼接略有不同。
2. 获取设备的 psk，可以在控制台查看或者通过 API 获取。
3. 将 psk 通过 base64 解码后作为 hash 的 key 对明文编码。
4. hash 后的结果转十六进制字符串拼接即可。

WIFI 绑定和蓝牙或其他绑定签名计算步骤一致，只是加密明文拼接略有不同，详见示例代码。

#### WIFI 配网绑定签名

明文拼接格式： "DeviceName=%s&DeviceTimestamp=%d&ProductId=%s&ConnId=%s"

##### Java

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;

public class AppBind {
    private static final String HMAC_SHA1_ALGORITHM = "HmacSHA1";
    private static final String HMAC_SHA256_ALGORITHM = "HmacSHA256";
    public static final String DEVICE_PSK = "4B***==";
    public static final String PRODUCT_ID = "productId";
    public static final String DEVICE_NAME = "d1";
    public static final String CONN_ID = "12345";
    public static final int TIMESTAMP = 1694141664;
    public static final String ContentFormat =
"DeviceName=%s&DeviceTimestamp=%d&ProductId=%s&ConnId=%s";
```

```
public static void main(String[] args) throws Exception {
    String content = String.format(ContentFormat, DEVICE_NAME,
    TIMESTAMP, PRODUCT_ID, CONN_ID);
    // sha1
    System.out.println(calculateHMACSHA1(content,
    Base64.getDecoder().decode(DEVICE_PSK)));
    // sha256
    System.out.println(calculateHMACSHA256(content,
    Base64.getDecoder().decode(DEVICE_PSK)));
}

// 计算sha1签名
public static String calculateHMACSHA1(String data, byte[] keyBytes)
throws Exception {
    SecretKeySpec signingKey = new SecretKeySpec(keyBytes,
    HMAC_SHA1_ALGORITHM);
    Mac mac = Mac.getInstance(HMAC_SHA1_ALGORITHM);
    mac.init(signingKey);
    byte[] dataBytes = data.getBytes();
    byte[] signatureBytes = mac.doFinal(dataBytes);

    return bytesToHex(signatureBytes);
}

// 计算sha256签名
public static String calculateHMACSHA256(String data, byte[]
keyBytes) throws Exception {
    SecretKeySpec signingKey = new SecretKeySpec(keyBytes,
    HMAC_SHA1_ALGORITHM);
    Mac mac = Mac.getInstance(HMAC_SHA256_ALGORITHM);
    mac.init(signingKey);
    byte[] dataBytes = data.getBytes();
    byte[] signatureBytes = mac.doFinal(dataBytes);

    return bytesToHex(signatureBytes);
}

private static String bytesToHex(byte[] bytes) {
    StringBuilder builder = new StringBuilder();
    for (byte b : bytes) {
        String hex = Integer.toHexString(b & 0xFF);
        if (hex.length() == 1) {
            hex = "0" + hex;
        }
        builder.append(hex);
    }
    return builder.toString();
}
```

```
        for (byte b : bytes) {
            builder.append(String.format("%02x", b));
        }

        return builder.toString();
    }
}
```

## Go

```
import (
    "crypto/hmac"
    "crypto/sha1"
    "encoding/base64"
    "encoding/hex"
    "fmt"
    "hash"
)

var (
    devicePsk = "Re*****Q=="
    contentFmt = "DeviceName=%s&DeviceTimestamp=%d&ProductId=%s&ConnId=%s"
)

func main() {
    fmt.Println(getSign("deviceName", "productId", "connId",
1693898532))
}

// 获取sha1签名
func getSha1Sign() string {
    psk, _ := base64.StdEncoding.DecodeString(devicePsk)
    var mac hash.Hash
    mac = hmac.New(sha1.New, psk)
    mac.Write([]byte(contentFmt))
    result := mac.Sum(nil)

    return hex.EncodeToString(result)
}
```

```
// 获取sha256签名
func getSha256Sign() string {
    psk, _ := base64.StdEncoding.DecodeString(devicePsk)
    var mac hash.Hash
    mac = hmac.New(sha256.New, psk)
    mac.Write([]byte(contentFmt))
    result := mac.Sum(nil)

    return hex.EncodeToString(result)
}
```

## C

```
#include <stdio.h>
#include <string.h>
#include <openssl/hmac.h>

const char *devicePsk = "Re5***==";
const char *contentFmt =
"DeviceName=%s&DeviceTimestamp=%ld&ProductId=%s&ConnId=%s";

char *getSign(char *deviceName, char *productId, char *connId, long
deviceTimestamp);

int main() {
    printf("%s\n", getSign("d1", "BRS**F", "9102344", 1693898532));
    return 0;
}

char *getSign(char *deviceName, char *productId, char *connId, long
deviceTimestamp) {
    unsigned char psk[16];
    int psk_len = EVP_DecodeBlock(psk, devicePsk, strlen(devicePsk));
    printf("%s\n", psk);

    char content[256];
    sprintf(content, contentFmt, deviceName, deviceTimestamp, productId,
    connId);
}
```

```
printf("%s\n", content);

unsigned char result[20];
unsigned int result_len;
HMAC(EVP_sha1(), psk, psk_len, (unsigned char *) content,
      strlen(content), result, &result_len);

char *hexResult = malloc(result_len * 2 + 1);
for (int i = 0; i < result_len; i++) {
    sprintf(hexResult + i * 2, "%02X", result[i]);
}
hexResult[result_len * 2] = '\0';
return hexResult;
}
```

## 蓝牙和其他配网绑定签名

明文拼接格式：`"%s%s;%s;%d"`，字段依次是 `ProductId`、`DeviceName`、`ConnId`、`DeviceTimestamp`。

### Java

```
import javax.crypto.Mac;
import javax.crypto.spec.SecretKeySpec;
import java.util.Base64;

public class AppBind {
    private static final String HMAC_SHA1_ALGORITHM = "HmacSHA1";
    private static final String HMAC_SHA256_ALGORITHM = "HmacSHA256";
    public static final String DEVICE_PSK = "4B*****";
    public static final String PRODUCT_ID = "productId";
    public static final String DEVICE_NAME = "d1";
    public static final String CONN_ID = "12345";
    public static final int TIMESTAMP = 1694141664;
    public static final String ContentFormat = "%s%s;%s;%d"; // ProductId;DeviceName;ConnId;DeviceTimestamp

    public static void main(String[] args) throws Exception {
        String content = String.format(ContentFormat, PRODUCT_ID,
                DEVICE_NAME, CONN_ID, TIMESTAMP);
```

```
// sha1
System.out.println(calculateHMACSHA1(content,
Base64.getDecoder().decode(DEVICE_PSK)));
// sha256
System.out.println(calculateHMACSHA256(content,
Base64.getDecoder().decode(DEVICE_PSK)));
}

// 计算sha1签名
public static String calculateHMACSHA1(String data, byte[] keyBytes)
throws Exception {
    SecretKeySpec signingKey = new SecretKeySpec(keyBytes,
HMAC_SHA1_ALGORITHM);
    Mac mac = Mac.getInstance(HMAC_SHA1_ALGORITHM);
    mac.init(signingKey);
    byte[] dataBytes = data.getBytes();
    byte[] signatureBytes = mac.doFinal(dataBytes);

    return bytesToHex(signatureBytes);
}

// 计算sha256签名
public static String calculateHMACSHA256(String data, byte[]
keyBytes) throws Exception {
    SecretKeySpec signingKey = new SecretKeySpec(keyBytes,
HMAC_SHA1_ALGORITHM);
    Mac mac = Mac.getInstance(HMAC_SHA256_ALGORITHM);
    mac.init(signingKey);
    byte[] dataBytes = data.getBytes();
    byte[] signatureBytes = mac.doFinal(dataBytes);

    return bytesToHex(signatureBytes);
}

private static String bytesToHex(byte[] bytes) {
    StringBuilder builder = new StringBuilder();
    for (byte b : bytes) {
        builder.append(String.format("%02x", b));
    }
}
```

```
        return builder.toString();
    }
}
```

## Go

```
import (
    "crypto/hmac"
    "crypto/sha1"
    "encoding/base64"
    "encoding/hex"
    "fmt"
    "hash"
)

var (
    devicePsk          = "Re*****"
    ProductId         = "productId"
    DeviceName        = "d1"
    ConnId            = "12345"
    DeviceTimestamp int64 = 1694141664
    contentFmt        = fmt.Sprintf("%s%s;%s;%d", ProductId,
    DeviceName, ConnId, DeviceTimestamp) //
    ProductId;DeviceName;ConnId;DeviceTimestamp)
func main() {
    fmt.Println(getSign())
}

func getSign() string {
    psk, _ := base64.StdEncoding.DecodeString(devicePsk)
    var mac hash.Hash
    mac = hmac.New(sha1.New, psk)
    mac.Write([]byte(contentFmt))
    result := mac.Sum(nil)

    return hex.EncodeToString(result)
}
```

## C

```
#include <stdio.h>
#include <string.h>
#include <openssl/hmac.h>

const char *devicePsk = "Re5****";
const char *contentFmt = "%s%s;%s;%d"; // ProductId;DeviceName;ConnId;DeviceTimestamp

char *getSign(char *deviceName, char *productId, char *connId, long deviceTimestamp);

int main() {
    printf("%s\n", getSign("d1", "BRS**F", "9102344", 1693898532));
    return 0;
}

char *getSign(char *deviceName, char *productId, char *connId, long deviceTimestamp) {
    unsigned char psk[16];
    int psk_len = EVP_DecodeBlock(psk, devicePsk, strlen(devicePsk));
    printf("%s\n", psk);

    char content[256];
    sprintf(content, contentFmt, productId, deviceName, connId,
deviceTimestamp);
    printf("%s\n", content);

    unsigned char result[20];
    unsigned int result_len;
    HMAC(EVP_sha1(), psk, psk_len, (unsigned char *) content,
strlen(content), result, &result_len);

    char *hexResult = malloc(result_len * 2 + 1);
    for (int i = 0; i < result_len; i++) {
        sprintf(hexResult + i * 2, "%02X", result[i]);
    }
    hexResult[result_len * 2] = '\0';

    return hexResult;
}
```

## }5. 示例

## 默认 WIFI 配网绑定请求输入示例1:

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
  "DeviceTimestamp": 1694141664,
  "Action": "AppSigBindDeviceInFamily",
  "ConnId": "1938",
  "RequestId": "dd7b-1014bc4-rey76",
  "AccessToken": "51dbxxx4d4fdb",
  "FamilyId": "f_8xxxxfb1a",
  "Signature": "7CE3518***69EA8C",
  "ProductId": "productId",
  "RoomId": "",
  "BindType": "wifi_sign",
  "DeviceName": "d1"
}
```

## 默认 WIFI 配网绑定请求输入示例2:

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
  "DeviceTimestamp": 1694141664,
  "Action": "AppSigBindDeviceInFamily",
  "ConnId": "1938",
  "RequestId": "dd7b-1014bc4-rey76",
  "AccessToken": "51dbxxx4d4fdb",
  "FamilyId": "f_8xxxxfb1a",
  "Signature": "7CE3518***69EA8C",
  "ProductId": "productId",
  "RoomId": "",
  "BindType": "wifi_sign",
  "SignMethod": "hmacsha256",
  "DeviceName": "d1"
}
```

### 默认蓝牙配网绑定请求输入示例3：

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
  "DeviceTimestamp": 1694141664,
  "Action": "AppSigBindDeviceInFamily",
  "ConnId": "1938",
  "RequestId": "dd7b-1014bc4-rey76",
  "AccessToken": "51dbxxx4d4fdb",
  "FamilyId": "f_8xxxxfb1a",
  "Signature": "7CE3518***69EA8C",
  "ProductId": "productId",
  "RoomId": "",
  "BindType": "bluetooth_sign",
  "DeviceName": "d1"
}
```

### 默认蓝牙配网绑定请求输入示例4：

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
  "DeviceTimestamp": 1694141664,
  "Action": "AppSigBindDeviceInFamily",
  "ConnId": "1938",
  "RequestId": "dd7b-1014bc4-rey76",
  "AccessToken": "51dbxxx4d4fdb",
  "FamilyId": "f_8xxxxfb1a",
  "Signature": "7CE3518***69EA8C",
  "ProductId": "productId",
  "RoomId": "",
  "BindType": "bluetooth_sign",
  "SignMethod": "hmacsha256",
  "DeviceName": "d1"
}
```

**输出示例：成功。**

```
{  
  "Response": {  
    "RequestId": "req_1",  
    "data": {  
      "AppDeviceInfo": {  
        "DeviceId": "45454/d2",  
        "ProductId": "45454",  
        "DeviceName": "d2",  
        "AliasName": "",  
        "CreateTime": 1552910676,  
        "UpdateTime": 1552910676,  
        "FamilyId": "1",  
        "RoomId": "0",  
        "IconUrl": ""  
      }  
    }  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.BindDeviceNotConnected	设备近期没有连接到云。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 绑定子设备到网关

最近更新时间：2024-10-11 18:26:02

## 1. 接口描述

接口请求域名：接口请求域名替换

本接口(AppSigGatewayBindDevice)绑定子设备到网关设备

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppSigGatewayBindDevice。
ProductId,	String	是	产品 ID。
DeviceName	String	是	设备名称。
GatewayProductId	String	是	网关产品 ID。
GatewayDeviceName	String	是	网关设备 Name。
DeviceTimeStamp	Int64	是	设备时间戳，Unix 秒级时间戳。
Nonce	String	是	随机字符串，建议 5 个字节长度。
Signature	String	是	动态签名。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 签名生成步骤：

1. 对参数 ( deviceName、nonce、productId、timestamp ) 按字典序升序排序。
2. 将以上参数，按参数名称 = 参数值 & 参数名称 = 参数值拼接成字符串。
3. 使用 HMAC-sha1 算法对上一步中获得的字符串进行计算，密钥为 ProductSecret。
4. 将生成的结果使用 Base64 进行编码，即可获得最终的签名串放入 signature。

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
  "RequestId": "zzY1pZb3sz1N",
  "Action": "AppSigGatewayBindDevice",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "GatewayProductId": "NJ27OVLZT4",
  "GatewayDeviceName": "gwdev",
  "ProductId": "LAEG4YJE1A",
  "DeviceName": "subdev3",
  "DeviceTimestamp": 1583810805,
  "Nonce": "123456",
  "Signature": "YZGf0E***WGUEJGn+SxUf23HYA="
}
```

#### 输出示例：成功

```
{
  "Response": {
    "RequestId": "zzY1pZb3sz1N"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。

InternalError.InternalServerException	发生错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.BindDeviceNotConnected	设备近期没有连接到云。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.InvalidAppParameter	App 请求的参数错误。
InvalidParameterValue.BindDeviceSigMismatch	绑定设备动态签名错误。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.NotBindVirtualDevice	无法绑定虚拟设备。
UnsupportedOperation.DeviceType	设备类型错误。
ResourceNotFound	资源不存在。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.AppNotExists	应用未创建或是已删除。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。
UnauthorizedOperation.NoPermissionToFamiliy	操作该家庭无权限。
LimitExceeded	超出限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制。

# 解绑子设备网关

最近更新时间：2024-11-05 15:02:52

## 1. 接口描述

接口请求域名：接口请求域名替换

本接口(AppGatewayUnbindDevice) 从网关中解绑子设备

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求ID，可自行生成，推荐使用uuld。定位问题时，需提供该次请求的RequestId。
Action	String	是	公共参数，本接口取值：AppGatewayUnbindDevice。
ProductId,	String	是	产品 ID。
DeviceName	String	是	设备名称。
GatewayProductId	String	是	网关产品 ID。
GatewayDeviceName	String	是	网关设备 Name。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求ID，可自行生成，推荐使用uuld。定位问题时，需提供该次请求的RequestId。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
  "RequestId": "zzY1pZb3sz1N",
  "Action": "AppGatewayUnbindDevice",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "GatewayProductId": "NJ****T4",
  "GatewayDeviceName": "gwdev",
  "ProductId": "LA*****1A",
  "DeviceName": "dev"
}
```

**输出示例: 成功**

```
{
  "Response": {
    "RequestId": "zzY1pZb3sz1N"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerException	发生错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.BindDeviceNotConnected	设备近期没有连接到云。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.InvalidAppParameter	App 请求的参数错误。
InvalidParameterValue.BindDeviceSigMismatch	绑定设备动态签名错误。
UnsupportedOperation	不支持的操作。

UnsupportedOperation.NotBindVirtualDevice	无法绑定虚拟设备。
UnsupportedOperation.DeviceType	设备类型错误。
ResourceNotFound	资源不存在。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.AppNotExists	应用未创建或是已删除。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。
UnauthorizedOperation.NoPermissionToFamily	操作该家庭无权限。
LimitExceeded	超出限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制

# 设备控制

## 设备透传指令控制

最近更新时间：2025-08-18 18:09:12

### 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（AppApiPublishMessage）用于使用自定义透传协议进行设备远控。

说明：

Signature 鉴权方式参见 [应用端 API 简介](#)。

## 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值：AppApiPublishMessage。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
AppKey	String	是	公共参数，应用 AppKey，用于标识对应的小程序或 App。
Signature	String	是	公共参数，请求签名，需用户自行生成，用于校验请求的合法性。
Timestamp	Int64	是	公共参数，请求的 UNIX 时间戳（秒级）。
Nonce	Int	是	公共参数，随机正整数，与 Timestamp 联合起来，防止重放攻击。
ProductId	String	是	产品 ID。 示例值：Nlasdf****ABCd
DeviceName	String	是	设备名称。 示例值：dev1
Topic	String	是	消息发往的主题。 示例值：RL0BAZKZ6V/dev1/control

Payload	String	是	云端下发到设备的控制报文。 示例值: AASDFASFSADFASDF
Qos	Integer	否	消息服务质量等级, 取值为0或1。 示例值: 1
PayloadEncoding	String	否	Payload 的内容编码格式, 取值为 base64 或空。base64 表示云端将接收到的 base64 编码后的报文再转换成二进制报文下发至设备, 为空表示不作转换, 透传下发至设备。 示例值: base64

## 输出参数

名称	类型	描述
RequestId	String	公共参数, 唯一请求 ID, 与入参相同。

## 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/appapi HTTP/1.1
content-type: application/json
{
  "Signature": "IY0yHX7zWCZKZ97xDonQoy16W4U=",
  "Timestamp": 1552621825,
  "Nonce": 2,
  "Action": "AppApiPublishMessage",
  "AppKey": "ahPxdK*****TGrejd",
  "RequestId": "rest-client",
  "Topic": "RL0BAZKZ6V/dev1/control",
  "Payload": "AASDFASFSADFASDF",
  "ProductId": "Nlasdf*****ABCd",
  "DeviceName": "dev1",
  "Qos": 1
}
```

### 输出示例: 成功

{

```
"Response": {  
    "RequestId": "rest-client"  
}  
}
```

### 输出示例：失败

```
{  
    "Response": {  
        "Error": {  
            "Code": "InvalidParameterValue.ErrorUserNotExists",  
            "Message": "账号不存在"  
        },  
        "RequestId": "rest-client"  
    }  
}
```

## 错误码

错误码	描述
InternalError	内部错误。
ErrorRequiredParamNotFound	必选参数缺失。
InvalidAction	Action 非法。
InvalidParameterValue	参数异常。
InvalidParameterValue.InvalidJSON	请求格式不是 JSON。
InvalidParameterValue.NickNameLengthInvalid	昵称长度非法

# 用户控制设备

最近更新时间：2023-07-19 10:57:11

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppControlDeviceData）用于用户对绑定的设备发起控制操作。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppControlDeviceData。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名。
Data	String	是	控制设备报文。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Data	String	返回数据。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
}
```

```
"RequestId": "req_1",
"AppId": 1111,
"Action": "AppControlDeviceData",
"ProductId": "22F9Y6II7O",
"DeviceName": "light1",
"Data": "{\"light_switch\":0}",
"AccessToken": "8b*****b8ec"
}
```

**输出示例：成功**

```
{
  "Response": {
    "Data": "",
    "RequestId": "req_1"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 同步调用设备行为

最近更新时间：2024-10-08 18:37:11

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppCallDeviceActionSync`）用于用户向已成功绑定的设备发起同步行为调用。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	<code>AppCallDeviceActionSync</code>
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
ActionId	String	是	行为功能的标识符，ActionId 在产品物模型的行为中定义。
InputParams	String	否	输入参数。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
ClientToken	String	调用 ID。
Status	String	返回状态。

OutputParams	String	输出参数。 注意：此字段可能返回 null，表示取不到有效值。
--------------	--------	------------------------------------

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "AccessToken": "accesstoken*****acc",
  "RequestId": "026fe48b-0c31-4a7c-aaf6-83ba3e9s4bcf",
  "ProductId": "AB*****2345",
  "DeviceName": "DEV",
  "ActionId": "actionid",
  "InputParams": "{\"color\":6}",
  "Action": "AppCallDeviceActionSync"
}
```

#### 输出示例：成功

```
{
  "Response": {
    "ClientToken": "1ee12*****7df703e0d",
    "OutputParams": "",
    "RequestId": "026fe48b-0c31-4a7c-aaf6-83ba3e9s4bcf",
    "Status": "succ"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。

InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 异步调用设备行为

最近更新时间：2024-10-08 18:37:11

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppCallDeviceActionAsync`）用于向用户所绑定的设备发起异步行为调用请求，异步方式调用设备行为返回结果可通过长连接通信实时获取。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	AppCallDeviceActionAsync
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
ActionId	String	是	行为功能的标识符。
InputParams	String	否	输入参数。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
ClientToken	String	调用 ID。
Status	String	返回状态。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "AccessToken": "accesstoken*****acc",
  "RequestId": "026fe48b-0c31-4a7c-aaf6-83ba3e9s4bcf",
  "ProductId": "ABCDE12345",
  "DeviceName": "DEV",
  "ActionId": "actionid",
  "InputParams": "{\"color\":6}",
  "Action": "AppCallDeviceActionAsync"
}
```

#### 输出示例: 成功

```
{
  "Response": {
    "ClientToken": "1ee12*****7df703e0d",
    "RequestId": "026fe48b-0c31-4a7c-aaf6-83ba3e9s4bcf",
    "Status": "Sent"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。



# 设备分享

## App 端发送设备分享

最近更新时间：2024-10-11 16:01:45

### 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppSendShareDeviceInvite）用于发送设备分享邀请。

### 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppSendShareDeviceInvite。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
ToUserId	String	是	被分享用户 ID。

### 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
Status	String	请求结果。

### 4. 示例

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Action": "AppSendShareDeviceInvite",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "RequestId": "1555507435215",
  "ProductId": "22F9Y6II7O",
  "DeviceName": "light1",
  "ToUserID": "123456"
}
```

### 输出示例: 成功

```
{
  "Response": {
    "RequestId": "1555507435215",
    "Status": "OK"
  }
}
```

### 输出示例: 失败

```
{
  "Response": {
    "Error": {
      "Code": "InvalidParameterValue.InvalidAccessToken",
      "Message": "Token无效"
    },
    "RequestId": "1555507435215"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。

InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。

# 获取设备分享 Token

最近更新时间：2024-10-11 16:01:45

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppCreateShareDeviceToken）用于获取设备分享 Token。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppCreateShareDeviceToken。
FamilyId	String	是	家庭 ID，成功创建家庭后，返回的 FamilyId
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
TokenContent	String	否	分享的上下文。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
ShareDeviceToken	String	设备分享 Token。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Action": "AppCreateShareDeviceToken",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "RequestId": "155550****15",
  "ProductId": "22F9****70",
  "DeviceName": "light1"
}
```

### 输出示例：成功

```
{
  "Response": {
    "RequestId": "1555507****15",
    "ShareDeviceToken": "49d2b71dc5814****7c68a4c756d3ce9"
  }
}
```

### 输出示例：失败

```
{
  "Response": {
    "Error": {
      "Code": "InvalidParameterValue.InvalidAccessToken",
      "Message": "Token无效"
    },
    "RequestId": "1555507****15"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。

InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。

# 获取设备分享 Token 信息

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppDescribeShareDeviceToken）用于获取设备分享 Token 信息。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppDescribeShareDeviceToken</code> 。
ShareDeviceToken	String	是	设备分享 Token。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
ShareDeviceTokenInfo	Object of <a href="#">ShareDeviceTokenInfo</a>	响应的结果。
AliasName	String	设备别名
IconUrl	String	图标 URL

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
```

```
content-type: application/json
{
  "Action": "AppDescribeShareDeviceToken",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "RequestId": "1555507****15",
  "ShareDeviceToken": "49d2b71dc5814354****8a4c756d3ce9"
}
```

### 输出示例: 成功

```
// UserId、UserNick 在被分享者接受前为空
// 被分享者接受分享后, UserId 保存的是被分享者的信息
// 前端可以通过UserId是否为空, 来判断绑定情况。
{
  "Response": {
    "RequestId": "1555507****15",
    "ShareDeviceInfo": {
      "FromUserId": "1",
      "FromUserNick": "tests",
      "UserId": "2",
      "UserNick": "test2",
      "ProductId": "22F9Y6II7O",
      "DeviceName": "light1",
      "AliasName": "my_light",
      "IconUrl": "",
      "BindTime": 0,
      "ExpireTime": 1574757262,
      "CreateTime": 1574152462
    }
  }
}
```

### 输出示例: 失败

```
{
  "Response": {
    "Error": {
      "Code": "InvalidParameterValue.InvalidShareDeviceToken",
      "Message": "设备分享Token无效"
    }
  }
}
```

```
    },
    "RequestId": "1555507****15"
}
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidShareDeviceToken	设备分享 Token 无效。

# 绑定用户分享的设备

最近更新时间：2024-10-11 16:01:46

## 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)。

本接口（AppBindUserShareDevice）用于绑定用户分享的设备。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： AppBindUserShareDevice。
ShareDeviceToken	String	是	设备分享的 Token。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
Status	String	响应结果。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
```

```
{  
  "Action": "AppBindUserShareDevice",  
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",  
  "RequestId": "1555507****15",  
  "ProductId": "22F9Y6II7O",  
  "DeviceName": "light1",  
  "ShareDeviceToken": "0820728466424d8f****422ec06bd398"  
}
```

### 输出示例: 成功

```
{  
  "Response": {  
    "RequestId": "1555507****15",  
    "Status": "OK"  
  }  
}
```

### 输出示例: 失败

```
{  
  "Response": {  
    "Error": {  
      "Code": "InvalidParameterValue.DeviceBinded",  
      "Message": "设备已经和当前用户绑定"  
    },  
    "RequestId": "1555507****15"  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.DeviecBinded	设备已经和当前用户绑定。

InvalidParameterValue.UserBindExceedLimit	用户绑定分享设备数已达限制。
InvalidParameterValue.DeviceShareExceededLimit	设备已达最大分享用户数限制。
InvalidParameterValue.CannotBindOwnDevice	不能绑定自己分享的设备。
InvalidParameterValue.ShareDeviceTokenUsed	设备分享 Token 已使用。
InvalidParameterValue.InvalidShareDeviceToken	设备分享 Token 无效。

# 查询用户分享设备列表

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)。

本接口（AppListUserShareDevices）用于查询用户分享的设备列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppListUserShareDevices。
Offset	Int	是	分页偏移，0起始，最大不超过500。
Limit	Int	是	单次拉取数量，默认为10，最大不超过100。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
Total	Int	返回的消息总条数。
ShareDevices	Array of ShareDevices	响应设备信息。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
}
```

```
"Action": "AppListUserShareDevices",
"AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
"RequestId": "1555507****15",
"Offset": 0,
"Limit": 10
}
```

## 输出示例: 成功

```
{
  "Response": {
    "RequestId": "1555507****15",
    "ShareDevices": [
      {
        "ProductId": "WTXXXX0MF2",
        "DeviceName": "device_XXXX001",
        "DeviceId": "WTXXXX0MF2/device_XXXX001",
        "AliasName": "test_device",
        "IconUrl": "https://yyy-xxxx-xxx.cos.ap-
guangzhou.myqcloud.com/113xxxx7a0b273bxxxxc0b9040feeb/1xxxxxxxxx0.png",
        "CreateTime": "2019-12-17T19:39:21+08:00"
      }
    ],
    "Total": 1
  }
}
```

## 输出示例: 失败

```
{
  "Response": {
    "Error": {
      "Code": "InvalidParameterValue.InvalidAccessToken",
      "Message": "Token无效"
    },
    "RequestId": "1555507****15"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccess Token	Token 无效。

# 删除用户分享的设备

最近更新时间：2024-10-11 16:01:46

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppRemoveUserShareDevice`）用于删除用户分享的设备。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求ID，可自行生成，推荐使用uuid。定位问题时，需提供该次请求的RequestId。
Action	String	是	公共参数，本接口取值： <code>AppRemoveUserShareDevice</code> 。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的RequestId。
Status	String	响应结果。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
```

```
"Action": "AppRemoveUserShareDevice",
"AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
"RequestId": "1555507****15",
"ProductId": "22F9Y6II7O",
"DeviceName": "light1"
}
```

### 输出示例: 成功

```
{
  "Response": {
    "RequestId": "1555507****15",
    "Status": "OK"
  }
}
```

### 输出示例: 失败

```
{
  "Response": {
    "Error": {
      "Code": "InvalidParameterValue.AppDeviceNotExist",
      "Message": "没有找到绑定设备"
    },
    "RequestId": "1555507****15"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.AppDeviceNotExist	没有找到绑定设备。

# 查询设备的用户列表

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名： `iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppListShareDeviceUsers`）用于查询设备用户列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppListShareDeviceUsers</code> 。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
Offset	Int	是	分页偏移，0起始，最大不超过500。
Limit	Int	是	单次拉取数量，默认为10，最大不超过100。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
Total	Int	返回的数据总数。
Users	Array of <code>ShareUserInfo</code>	用户列表。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Action": "AppListShareDeviceUsers",
  "AccessToken": "user2",
  "RequestId": "1555507****15",
  "ProductId": "22F9Y6II7O",
  "DeviceName": "light1",
  "Offset": 0,
  "Limit": 10
}
```

### 输出示例: 成功

```
{
  "Response": {
    "RequestId": "1555507****15",
    "Users": [
      {
        "UserId": "1",
        "CountryCode": "86",
        "PhoneNumber": "139****5678",
        "NickName": "tests",
        "Avatar": "",
        "BindTime": 1574153536
      }
    ],
    "Total": 1
  }
}
```

### 输出示例: 失败

```
{
  "Response": {
    "Error": {
      "Code": "InvalidParameterValue.InvalidAccessToken",
      "Message": "Token无效"
    }
  }
}
```

```
    },
    "RequestId": "1555507****15"
}
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 删除设备分享的用户

最近更新时间：2023-07-14 16:01:46

## 1. 接口描述

接口请求域名： `iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppRemoveShareDeviceUser`）用于删除设备分享的用户。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求ID，可自行生成，推荐使用uuid。定位问题时，需提供该次请求的RequestId。
Action	String	是	公共参数，本接口取值： <code>AppRemoveShareDeviceUser</code> 。
RemoveUserId	String	是	待删除用户 ID。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的RequestId。
Status	String	响应结果。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
```

```
{  
  "Action": "AppRemoveShareDeviceUser",  
  "AccessToken": "user2",  
  "RequestId": "15555*****15",  
  "RemoveUserId": "1234555",  
  "ProductId": "22F9Y6II70",  
  "DeviceName": "light1"  
}
```

### 输出示例: 成功

```
{  
  "Response": {  
    "RequestId": "155550*****15",  
    "Status": "OK"  
  }  
}
```

### 输出示例: 失败

```
{  
  "Response": {  
    "Error": {  
      "Code": "InvalidParameterValue.AppDeviceNotExist",  
      "Message": "没有找到绑定设备"  
    },  
    "RequestId": "15555*****15"  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.AppDeviceNotExist	没有找到绑定设备。

InvalidParameterValue.NoPermission

用户对该设备无权限。

# 家庭管理

## 创建家庭

最近更新时间：2024-11-01 16:30:32

### 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppCreateFamily）用于创建家庭。

### 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppCreateFamily。
Name	String	是	家庭名称。
Address	String	是	家庭地址。

### 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
FamilyId	String	成功创建的家庭 ID。

### 4. 示例

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
```

```
"Action": "AppCreateFamily",
  "Name": "family_name",
  "Address": "family_address",
  "AccessToken": "c1*****847e"
}
```

**输出示例：成功**

```
{
  "Response": {
    "Data": {
      "FamilyId": "e0e23a2b33a24652b6*****a1cf"
    },
    "RequestId": "req_1"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 删除家庭

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppDeleteFamily`）用于删除家庭。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppDeleteFamily</code> 。
Name	String	是	需要删除的家庭名称。
FamilyId	String	是	需要删除的家庭 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppDeleteFamily",
  "Name": "XXX的家庭",
  "FamilyId": "e0e23a2b33a24652b6*****a1cf",
  "AccessToken": "c1*****847e"
```

}

### 输出示例：成功

```
{  
  "Response": {  
    "RequestId": "req_1"  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 修改家庭

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppModifyFamily）用于修改家庭。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppModifyFamily。
Name	String	是	修改后的家庭名称。
FamilyId	String	是	需要修改的家庭 ID。
Address	String	是	修改后的家庭地址。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppModifyFamily",
  "Name": "family_name",
```

```
"Address": "family_address",
"FamilyId": "e0e23a2b33a24652b6*****a1cf",
"AccessToken": "c1*****847e"
}
```

**输出示例：成功**

```
{
  "Response": {
    "RequestId": "req_1"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 获取家庭列表

最近更新时间：2024-11-01 17:48:52

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetFamilyList）用于获取家庭列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetFamilyList。
Offset	Int	否	所需要查询数据的偏移量。
Limit	Int	否	返回条数的限制量，最多为50条。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同。
Total	Int	返回的家庭数量总数。
FamilyList	Array	返回的家庭列表信息数组，请参见 <a href="#">FamilyList</a> 。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppGetFamilyList",
```

```
"Offset":0,  
"Limit":2,  
"AccessToken": "c1*****847e"  
}
```

### 输出示例：成功

```
{  
  "Response": {  
    "RequestId": "req_1",  
    "FamilyList": [{  
      "FamilyId": "a1c6939b39d345b897*****a12c",  
      "Name": "family_name",  
      "Role": 0, // 1:管理员 0: 普通成员  
      "CreateTime": 1570786578,  
      "UpdateTime": 1570790807  
    }],  
    "Total":1  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 获取家庭详情

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppDescribeFamily`）用于获取家庭详情。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppDescribeFamily</code> 。
FamilyId	String	是	所需查询详情的家庭 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
FamilyId	String	家庭 ID。
FamilyName	String	家庭名称。
Address	String	家庭地址。
CreateTime	Int64	创建时间，UNIX 时间戳（秒级）。
UpdateTime	Int64	最后一次更新时间，UNIX 时间戳（秒级）。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppDescribeFamily",
  "FamilyId": "a1c6939b39d345b897*****a12c",
  "AccessToken": "c1*****847e"
}
```

### 输出示例：成功

```
{
  "Response": {
    "RequestId": "req_1",
    "Data": {
      "FamilyId": "a1c6939b39d345b897*****a12c",
      "FamilyName": "family_name",
      "Address": "family_address",
      "CreateTime": 1570786578,
      "UpdateTime": 1570790807
    }
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 新建房间

最近更新时间：2023-07-19 10:57:12

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppCreateRoom）用于新建房间。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppCreateRoom。
FamilyId	String	是	新建房间的所属家庭 ID。
Name	String	是	新建的房间名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
RoomId	String	成功创建的房间 ID。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppCreateRoom",
  "Name": "name",
```

```
"FamilyId": "family_address",
"AccessToken": "c1*****847e"
}
```

### 输出示例：成功

```
{
  "Response": {
    "Data": {
      "RoomId": 123456
    },
    "RequestId": "req_1"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 修改房间

最近更新时间：2024-10-11 10:57:12

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppModifyRoom）用于修改房间。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppModifyRoom。
FamilyId	String	是	需要修改的房间所属家庭 ID。
Name	String	否	房间名称。
RoomId	String	是	需要修改的房间 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
    "RequestId": "req_1",
    "Action": "AppModifyRoom",
    "FamilyId": "fa070f59a2fe4****43815310c794fdf",
```

```
"RoomId": "5",
"Name": "TestName",
"AccessToken":
"8b4a70dd16105f*****18edd4e78a3bb8ec"
}
```

**输出示例：成功**

```
{
  "Response": {
    "RequestId": "req_1"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 删除房间

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名： `iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppDeleteRoom）用于删除房间。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppDeleteRoom。
FamilyId	String	是	需要删除的房间所属家庭 ID。
RoomId	String	是	需要删除的房间 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppDeleteRoom",
  "FamilyId": "e0e23a2b33a24652b6*****a1cf",
  "RoomId": "123456",
  "AccessToken": "c1*****847e"
```

}

### 输出示例：成功

```
{  
  "Response": {  
    "RequestId": "req_1"  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 获取房间列表

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetRoomList）用于获取房间列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetRoomList。
Offset	Int	否	所需要查询的数据的偏移量。
Limit	Int	否	所需要查询的总限制量，最大返回50条。
FamilyId	String	否	需要获取的家庭 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
Total	Int	返回的设备总数
RoomList	Array	返回的家庭列表信息数组，请参见 <a href="#">Roomlist</a> 。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
}
```

```
"RequestId": "req_1",
"Action": "AppGetRoomList",
"FamilyId": "fa070f59a2fe4f1da4*****4fdf",
"Offset": 0,
"Limit": 2,
"AccessToken": "xxxx"
}
```

### 输出示例：成功

```
{
  "Response": {
    "RequestId": "req_1",
    "RoomList": [
      {
        "RoomId": "123456",
        "RoomName": "name",
        "DeviceNum": 2,
        "CreateTime": 1570786578,
        "UpdateTime": 1570790807
      }
    ],
    "Total": 1
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效

# 邀请家庭成员

最近更新时间：2025-12-29 15:26:13

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppInviteMember`）用于向微信好友发送邀请加入家庭请求。

## 2. 输入参数

名称	类型	必选	描述
<code>AccessToken</code>	<code>String</code>	是	公共参数， <code>AccessToken</code> 用于对一个已经登录的用户鉴权。
<code>RequestId</code>	<code>String</code>	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需提供该次请求的 <code>RequestId</code> 。
<code>Action</code>	<code>String</code>	是	公共参数，本接口取值： <code>AppInviteMember</code> 。
<code>FamilyId</code>	<code>String</code>	是	家庭 ID。

## 3. 输出参数

名称	类型	描述
<code>RequestId</code>	<code>String</code>	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 <code>RequestId</code> 。
<code>ShareToken</code>	<code>String</code>	分享 Token，被邀请者可凭此分享 Token 调用 <a href="#">成员加入家庭</a> 接口以加入家庭。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppInviteMember",
  "FamilyId": "e0e23a2b33a24652b6*****a1cf",
  "AccessToken": "c1*****847e"
```

}

## 输出示例：成功

```
{  
  "Response": {  
    "ShareToken": "e0*****a1cf",  
    "RequestId": "req_1"  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 成员加入家庭

最近更新时间：2025-12-29 15:26:14

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppJoinFamily）用于成员加入家庭。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppJoinFamily。
ShareToken	String	是	分享 Token，由家庭管理员调用 <a href="#">邀请家庭成员</a> 生成，被邀请者可凭此分享 Token 加入相应的家庭。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "550e8400-e29b-41d4-a716-446655440000",
  "Action": "AppJoinFamily",
  "ShareToken": "e0*****a1cf",
  "AccessToken": "c1*****847e"
}
```

## 输出示例：成功

```
{  
  "Response": {  
    "RequestId": "550e8400-e29b-41d4-a716-446655440000"  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 删除家庭成员

最近更新时间：2024-10-11 10:57:13

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppDeleteFamilyMember`）用于管理员删除家庭成员。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppDeleteFamilyMember</code> 。
FamilyId	String	否	要移除成员的家庭 ID。
MemberId	String	是	需要移除的成员 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "550e8400-e29b-41d4-a716-446655440000",
  "Action": "AppDeleteFamilyMember",
  "FamilyId": "f_abcd****abcd",
  "MemberId": "12345678",
```

```
"AccessToken": "c1*****847e"  
}
```

**输出示例：成功**

```
{  
  "Response": {  
    "RequestId": "550e8400-e29b-41d4-a716-446655440000"  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 成员退出家庭

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppExitFamily`）用于成员主动退出某个家庭。

## 2. 输入参数

名称	类型	必选	描述
<code>AccessToken</code>	<code>String</code>	是	公共参数， <code>AccessToken</code> 用于对一个已经登录的用户鉴权。
<code>RequestId</code>	<code>String</code>	是	公共参数，唯一请求ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需提供该次请求的 <code>RequestId</code> 。
<code>Action</code>	<code>String</code>	是	公共参数，本接口取值： <code>AppExitFamily</code> 。
<code>FamilyId</code>	<code>String</code>	是	需要退出的家庭 ID。

## 3. 输出参数

名称	类型	描述
<code>RequestId</code>	<code>String</code>	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 <code>RequestId</code> 。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "550e8400-e29b-41d4-a716-44665*****00",
  "Action": "AppExitFamily",
  "FamilyId": "e0e23a2b33a24652b6*****a1cf",
  "AccessToken": "c1*****847e"
}
```

## 输出示例：成功

```
{  
  "Response": {  
    "RequestId": "550e8400-e29b-41d4-a716-44665****00"  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 获取家庭成员列表

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetFamilyMemberList）用于获取家庭成员列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetFamilyMemberList。
Offset	Int	否	所需要查询的数据的偏移量。
Limit	Int	否	所需要查询的总限制量，最大返回50条。
FamilyId	String	是	需要获取的家庭 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
Total	Int	返回的家庭成员总数。
MemberList	Array	返回的家庭成员数组，请参见 <a href="#">MemberList</a> 。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
```

```
{  
    "FamilyId": "f_b3ac7e6b1cd24392b41e*****5a",  
    "Offset": 0,  
    "Limit": 100,  
    "Action": "AppGetFamilyMemberList",  
    "RequestId": "hojkE0Hx88",  
    "AccessToken": "3c*****40c5"  
}
```

### 输出示例：成功

```
{  
    "MemberList": [  
        {  
            "UserID": "7267",  
            "NickName": "deny",  
            "Avatar":  
                "https://wx.qlogo.cn/mmopen/vi_32/V2kIBBE35phMrFKu5sra16bzpUuPtokyWTbMRP  
                t7bRGBjx4fEqQrZxoDYyxNVbtkXMosuxQeIqiafIZJ8ADr8cw/132",  
            "Role": 1  
        }],  
    "Total": 1,  
    "RequestId": "hojkE0Hx88"  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# App 端邀请家庭成员

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppSendShareFamilyInvite）用于发送家庭分享邀请。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppSendShareFamilyInvite
FamilyId	String	是	家庭 ID
ToUserId	String	是	被分享用户 ID

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Action": "AppSendShareFamilyInvite",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
```

```
"RequestId": "1555*****5215",
"FamilyId": "e0e23a2b33a24652b6*****a1cf",
"ToUserID": "123456"
}
```

### 输出示例: 成功

```
{
  "Response": {
    "RequestId": "1555*****5215"
  }
}
```

### 输出示例: 失败

```
{
  "Response": {
    "Error": {
      "Code": "InvalidParameterValue.InvalidAccessToken",
      "Message": "Token无效"
    },
    "RequestId": "1555*****5215"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该家庭无权限。

# 设备定时

## 新建定时任务

最近更新时间：2024-11-01 16:30:32

### 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppCreateTimer）用于用户创建设备的定时任务。

### 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值 AppCreateTimer。
ProductId	String	是	新建定时任务所属产品 ID。
DeviceName	String	是	新建定时任务控制的设备名称。
TimerName	String	是	定时器的名称。
Days	String	是	定时器开启时间，每一位，0：关闭，1：开启，从左至右依次表示：周日、周一、周二、周三、周四、周五、周六。
TimePoint	String	是	定时器开启时间点。
Repeat	int	是	是否循环。 <ul style="list-style-type: none"><li>0：不需要。</li><li>1：需要。</li></ul>
Data	String	是	定时器启动时下发的控制报文。

### 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
TimerId	String	定时器 ID。
TimerName	String	定时器名称。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
    "RequestId": "req_1",
    "Action": "AppCreateTimer",
    "ProductId": "US4*****IK",
    "DeviceName": "411_3",
    "AccessToken": "c1*****847e",
    "TimerName": "timer_test",
    "Days": "1000000", // 每一位 0:关闭,1:开启, 从左至右依次表示: 周日 周一 周二 周三 周四 周五 周六
    "TimePoint": "06:30",
    "Repeat": 1, // 是否需要循环 0:不需要 1 需要
    "Data": "{\"brightness\": 25}"
}
```

### 输出示例：成功

```
{
    "Response": {
        "Data": {
            "TimerId": "e0e23a2b33a24652b6*****a1cf",
            "TimerName": "timer_test"
        },
        "RequestId": "req_1"
    }
}
```

}

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 修改定时任务

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppModifyTimer）用于修改设备的定时任务。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值 AppModifyTimer。
ProductId	String	是	修改定时器对应的产品 ID。
DeviceName	String	是	定时器对应的设备名称。
TimerName	String	是	定时器的名称。
TimerId	String	是	定时器 ID。
TimePoint	String	是	定时器开启时间点。
Days	String	是	定时器开启时间，每一位，0：关闭，1：开启，从左至右依次表示：周日、周一、周二、周三、周四、周五、周六。
Repeat	Int	是	是否循环。 <ul style="list-style-type: none"><li>0：不需要。</li><li>1：需要。</li></ul>
Data	String	是	定时器启动时，下发的控制报文。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
    "RequestId": "req_1",
    "Action": "AppModifyTimer",
    "ProductId": "US*****DIK",
    "DeviceName": "411_3",
    "AccessToken": "c1*****847e",
    "TimerId": "a1c6939b39d345b897*****a12c",
    "TimerName": "timer_test_modify",
    "Days": "1100000",
    "TimePoint": "9:30",
    "Repeat": 0,
    "Data": "{\"brightness\": 28}"
}
```

### 输出示例：成功

```
{
    "Response": {
        "RequestId": "req_1"
    }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。

InvalidParameterValue.InvalidAccessToken

Token 无效。

# 删除定时任务

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppDeleteTimer）用于删除设备定时任务。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需要提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值 AppDeleteTimer。
ProductId	String	是	修改定时器对应的产品 ID。
DeviceName	String	是	定时器对应的设备名称。
TimerId	String	是	定时器 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
```

```
"Action": "AppDeleteTimer",
"ProductId": "US4*****IK",
"DeviceName": "411_3",
"TimerId": "e0e23a2b33a24652b6*****a1cf",
"AccessToken": "c1*****847e"
}
```

### 输出示例：成功

```
{
  "Response": {
    "RequestId": "req_1"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 修改定时任务状态

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppModifyTimerStatus）用于修改定时任务的启停状态。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值 AppModifyTimerStatus。
ProductId	String	是	需要修改定时任务列表的产品 ID。
DeviceName	String	是	需修改定时器状态设备名称。
TimerId	String	是	定时器 ID。
Status	Int	否	需要修改的定时器状态。 <ul style="list-style-type: none"><li>0：关闭。</li><li>1：开启。</li></ul>

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
```

```
content-type: application/json
{
    "RequestId": "req_1",
    "Action": "AppModifyTimerStatus",
    "ProductId": "US4*****IK",
    "DeviceName": "411_3",
    "TimerId": "a1c6939b39d345b897*****a12c",
    "Status": 0,
    "AccessToken": "c1*****847e"
}
```

### 输出示例：成功

```
{
    "Response": {
        "RequestId": "req_1"
    }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 获取定时任务列表

最近更新时间：2024-11-01 16:30:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetTimerList）用于获取定时器列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值 AppGetTimerList。
ProductId	String	是	需要获取定时任务列表的产品 ID。
DeviceName	String	是	需要获取定时任务列表的设备名称。
Offset	Int	否	获取的列表偏移量。
Limit	Int	否	获取到的总消息条数，最大为50。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
TimerList	Array	获取到的产品列表，请参见 <a href="#">TimerList</a> 。
Total	Int	总共拥有的条数。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
    "RequestId": "req_1",
    "Action": "AppGetTimerList",
    "ProductId": "US4*****IK",
    "DeviceName": "411_3",
    "Offset": 0,
    "Limit": 2,
    "AccessToken": "c1*****847e"
}
```

## 输出示例：成功

```
{
    "Response": {
        "RequestId": "req_1",
        "Total": 2,
        "TimerList": [
            {
                "TimerId": "a1c6939b39d345b897*****a12c",
                "TimerName": "timer_test_modify",
                "ProductId": "US4*****IK",
                "DeviceName": "411_3",
                "Days": "1100000",
                "TimePoint": "09:30",
                "Repeat": 0,
                "Data": "{\"brightness\": 28}",
                "Status": 0,
                "CreateTime": 1570786578,
                "UpdateTime": 1570790807
            },
            {
                "TimerId": "f888dd20c70744939a*****1c3a",
                "TimerName": "timer_test",
                "ProductId": "US4*****IK",
                "DeviceName": "411_3",
                "Days": "1000000",
                "TimePoint": "07:00",
                "Repeat": 1,
                "Data": "{\"brightness\": 25}"
            }
        ]
    }
}
```

```
        "Status": 1,  
        "CreateTime": 1570786483,  
        "UpdateTime": 1570786483  
    } ]  
}  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 固件升级

## 上报设备固件版本

最近更新时间：2024-10-03 11:18:11

### 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppReportFirmwareVersion）用于上报设备固件版本。

### 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppReportFirmwareVersion。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
Version	String	是	设备上报的固件版本。

### 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同。定位问题时，需提供该次请求的 RequestId。

### 4. 示例

#### 示例1

##### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b",
  "Action": "AppReportFirmwareVersion",
  "Version": "1.0",
  "ProductId": "PR*****ID",
  "DeviceName": "de****01",
  "AccessToken": "871d2d42*****692afae0"
}
```

### 输出示例：成功

```
{
  "Response": {
    "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。

# 查询设备可升级固件版本

最近更新时间：2024-10-11 10:57:14

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppCheckFirmwareUpdate）用于查询设备可升级固件版本。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppCheckFirmwareUpdate。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
CurrentVersion	String	设备当前固件版本。
DstVersion	String	固件可升级版本。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
```

```
content-type: application/json
{
  "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b",
  "Action": "AppCheckFirmwareUpdate",
  "ProductId": "PR*****ID",
  "DeviceName": "de***01",
  "AccessToken": "871d2d42*****692afae0"
}
```

**输出示例：成功**

```
{
  "Response": {
    "DstVersion": "1.1",
    "CurrentVersion": "1.0",
    "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。
FailedOperation	操作失败。
FailedOperation.DeviceInfoOutdated	设备固件版本错误（设备上报的固件版本与当前固件升级任务不匹配）。

# 查询设备固件下载地址

最近更新时间：2024-10-03 11:18:11

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetDeviceOTAInfo）用于查询设备固件下载地址。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetDeviceOTAInfo。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
FirmwareURL	String	固件下载地址。
TargetVersion	String	目标固件版本。
UploadVersion	String	设备上报的固件版本。

## 4. 示例

## 示例1

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "AccessToken": "871d2d42*****692afae0",
  "Action": "AppGetDeviceOTAInfo",
  "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b",
  "ProductId": "PR*****ID",
  "DeviceName": "de***01",
}
```

### 输出示例: 成功

```
{
  "Response": {
    "FirmwareURL": "https://examplebucket-12500000000.cos.ap-
guangzhou.myqcloud.com/path/to/firmware?sign=***",
    "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b",
    "TargetVersion": "1.1",
    "UploadVersion": "1.0"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。
ResourceNotFound	资源不存在。
ResourceNotFound.DeviceFirmwareTaskNotExist	未查询到固件升级任务。



# 确认固件升级任务

最近更新时间：2024-10-11 10:57:14

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppPublishFirmwareUpdateMessage`）用于用户确认升级后，云端向设备发起固件升级请求。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppPublishFirmwareUpdateMessage</code> 。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
```

```
"RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b",
"Action": "AppPublishFirmwareUpdateMessage",
"ProductId": "PR*****ID",
"DeviceName": "de****01",
"AccessToken": "871d2d42*****692afae0"
}
```

### 输出示例：成功

```
{
  "Response": {
    "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。
ResourceNotFound	资源不存在。
ResourceNotFound.DeviceHasNoFirmware	设备无固件版本。
FailedOperation	操作失败。
FailedOperation.DeviceOffline	设备处于离线状态。

# 上报固件升级任务状态

最近更新时间：2024-10-03 11:18:11

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppReportOTASatus）用于上报设备固件升级状态及进度。

## 2. 输入参数

名称	类型	必选	描述
Access Token	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppReportOTASatus。
ProductId	String	是	产品 ID。
Device Name	String	是	设备名称。
Version	String	是	固件升级的目标版本。
State	String	是	固件升级状态： <ul style="list-style-type: none"><li>downloading：下载中。</li><li>fail：升级失败。</li><li>done：升级成功。</li><li>burning：烧录中。</li><li>updating：更新中。</li></ul>
Result Code	Int	是	错误码： <ul style="list-style-type: none"><li>0：成功。</li><li>-1：下载超时。</li><li>-2：文件不存在。</li><li>-3：签名过期。</li><li>-4：MD5 不匹配。</li></ul>

- -5: 更新固件失败。

Result Msg	String	否	错误附加信息，当 State 取值为 fail 时必填。
Persent	Int	否	升级进度百分比，取值范围 0 ~ 100。

## 3. 输出参数

名称	类型	描述
Request Id	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "AccessToken": "871d2d42*****692afae0",
  "Action": "AppReportOTASstatus",
  "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b",
  "ProductId": "PR*****ID",
  "DeviceName": "de***01",
  "State": "done",
  "ResultCode": 0,
  "ResultMsg": "",
  "Version": "1.1",
  "Persent": 100
}
```

#### 输出示例: 成功

```
{
  "Response": {
    "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b"
  }
}
```

}

## 示例2

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "AccessToken": "871d2d42*****692afae0",
  "Action": "AppReportOTAStatus",
  "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b",
  "ProductId": "PR*****ID",
  "DeviceName": "de***01",
  "State": "downloading",
  "ResultCode": 0,
  "ResultMsg": "",
  "Version": "2.0",
  "Persent": 20
}
```

### 输出示例：成功

```
{
  "Response": {
    "RequestId": "584406e9-bf59-46a3-8d45-39f1891bed7b"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。



# 查询设备固件升级状态

最近更新时间：2024-10-11 16:01:49

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppDescribeFirmwareUpdateStatus`）用于查询设备固件升级状态及进度。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppDescribeFirmwareUpdateStatus</code> 。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
OriVersion	String	升级任务源版本。
DstVersion	String	升级任务目标版本。
ErrMsg	String	错误信息。
Percent	Int64	升级进展百分比。
Status	Int64	升级状态：

- 0: 设备离线。
- 1: 待处理。
- 2: 消息下发成功。
- 3: 下载中。
- 4: 烧录中。
- 5: 失败。
- 6: 升级完成。
- 7: 正在处理中。
- 8: 等待用户确认。
- 20: 下载完成。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
  "Action": "AppDescribeFirmwareUpdateStatus",
  "ProductId": "22*****0",
  "DeviceName": "li*****1",
  "AccessToken": "8b4d16105f*****18edd4e78a3bb8ec"
}
```

### 输出示例: 成功

```
{
  "Response": {
    "DstVersion": "2.0",
    "ErrMsg": "",
    "OriVersion": "1.0",
    "Percent": 0,
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 查看固件信息

最近更新时间：2023-10-24 17:09:51

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppDescribeFirmware`）用于查询设备固件信息。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppDescribeFirmware</code> 。
ProductId	String	是	产品 ID
FirmwareVersion	String	是	固件版本号

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
ProductId	String	产品 ID
Name	String	固件名称。 注意：此字段可能返回 null，表示取不到有效值。
Description	String	固件描述。 注意：此字段可能返回 null，表示取不到有效值。
Md5sum	String	固件 Md5 值。 注意：此字段可能返回 null，表示取不到有效值。
Createtime	Integer	固件上传的秒级时间戳。 注意：此字段可能返回 null，表示取不到有效值。

ProductName	String	产品名称
FwType	String	固件升级模块。 注意：此字段可能返回 null，表示取不到有效值。
Version	String	固件版本号

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
  "Action": "AppDescribeFirmware",
  "ProductId": "22*****0",
  "FirmwareVersion": "t.2",
  "AccessToken": "8b4d16105f*****18edd4e78a3bb8ec"
}
```

### 输出示例：成功

```
{
  "Response": {
    "Createtime": 1599626765,
    "Description": "ttttt",
    "Md5sum": "1a5c386576074d22a604b795e8917e1a",
    "Name": "test",
    "ProductId": "I6KTCZ170U",
    "ProductName": "8bvsn6go_测试OTA升级",
    "RequestId": "01365e8c-f025-40b9-97a4-fa583d6c569e",
    "Version": "t.2",
    "FwType": "mcu"
  }
}
```

## 5. 错误码

错误码	描述
InternalError.InternalRPCError	内部 RPC 错误。
ResourceNotFound.FirmwareNotExist	固件不存在。
ResourceNotFound.StudioProductNotExist	产品不存在。

# 获取设备当前版本信息

最近更新时间：2025-12-29 15:26:14

## 1. 接口描述

接口请求域名：iot.cloud.tencent.com/api/exploreropen/tokenapi。

本接口（AppDescribeDeviceVersion）用于获取设备当前版本。

## 2. 输入参数

名称	类型	必选	描述
Action	string	是	公共参数，本接口取值：AppDescribeDeviceVersion。
RequestId	string	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
AccessToken	string	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
ProductId	string	是	产品 ID。
DeviceName	string	是	设备名称。

## 3. 输出参数

名称	类型	描述
DeviceVersions	array	返回设备版本数组，请参见 <a href="#">DeviceVersions</a> 。
RequestId	string	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
```

```
content-type: application/json
{
    "Action": "AppDescribeDeviceVersion",
    "RequestId": "ksdfsdf",
    "AccessToken": "c1*****847e",
    "ProductId": "DBLAWIULBC",
    "DeviceName": "11"
}
```

### 输出示例：成功

```
{
    "Response": {
        "DeviceVersions": [
            {
                "Version": "2.0",
                "Module": "mcu"
            }
        ],
        "RequestId": "ksdfsdf"
    }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。
ResourceNotFound	资源不存在。
ResourceNotFound.DeviceFirmwareTaskNotExist	未查询到固件升级任务。

# 消息管理

## 获取消息列表

最近更新时间：2023-07-19 10:57:14

### 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetMessages）用于查询消息列表，消息包括设备告警消息、操作通知消息。

### 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时需要提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetMessages。
Category	Int64	是	主类型。 <ul style="list-style-type: none"><li>1: 设备。</li><li>2: 家庭。</li><li>3: 通知。</li></ul>
MsgID	String	否	消息 ID，首次可不传。
MsgTimeStamp	Int64	否	消息的时间戳，首次可不传或传0。
Limit	Int64	是	最大返回条数，最大不超过100。

### 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
Listover	String	用于翻页。

### 4. 示例

## 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
    "Action": "AppGetMessages",
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
    "AccessToken": "8b*****b8ec",
    "Category": 2,
    "MsgID": "",
    "MsgTimestamp": 0,
    "Limit": 5
}
```

## 输出示例：成功

```
{
    "Response": {
        "Data": {
            "Listover": true,
            "Msgs": [
                {
                    "UserID": "1",
                    "FromUserID": "",
                    "MsgID": "65167410520395776",
                    "Category": 1,
                    "MsgType": 100,
                    "MsgTitle": "Device Warning",
                    "MsgContent": "Some message Wed Nov 27 2019 11:52:01
GMT+0000 .",
                    "MsgTimestamp": 1574855521420,
                    "ProductId": "",
                    "DeviceName": "",
                    "DeviceAlias": "",
                    "FamilyId": "",
                    "FamilyName": "",
                    "RelateUserID": "",
                    "CreateAt": "2019-11-27T11:52:01.42Z"
                },
                {
                    "UserID": "1",

```

```
"FromUserID": "1",
"MsgID": "65167395785805824",
"Category": 2,
"MsgType": 200,
"MsgTitle": "成员添加结果通知",
"MsgContent": "“jordan”将“dylan”添加为“深圳的家”的成员，成员
可以查看和控制家里的智能设备。 Wed Nov 27 2019 11:51:57 GMT+0000",
"MsgTimestamp": 1574855517907,
"ProductId": "",
"DeviceName": "",
"DeviceAlias": "",
"FamilyId": "1",
"FamilyName": "test",
"RelateUserID": "2",
"CreateAt": "2019-11-27T11:51:57.907Z"
}, {
"UserID": "1",
"FromUserID": "1",
"MsgID": "65167386239569920",
"Category": 2,
"MsgType": 201,
"MsgTitle": "成员删除结果通知",
"MsgContent": "“jordan”将“dylan”从“深圳的家”中删除，删除后将
不能查看和控制家中的智能设备。 Wed Nov 27 2019 11:51:54 GMT+0000",
"MsgTimestamp": 1574855515631,
"ProductId": "",
"DeviceName": "",
"DeviceAlias": "",
"FamilyId": "1",
"FamilyName": "test",
"RelateUserID": "2",
"CreateAt": "2019-11-27T11:51:55.631Z"
}, {
"UserID": "1",
"FromUserID": "1",
"MsgID": "65167339519217664",
"Category": 3,
"MsgType": 300,
"MsgTitle": "设备分享结果通知",
```

```
"MsgContent": "“dylan”已成功接收“jordan”分享的设备“智能灯”，  
现在可以查看和控制设备了。 Wed Nov 27 2019 11:51:44 GMT+0000",  
    "MsgTimestamp": 1574855504492,  
    "ProductId": "22F9Y6II7O",  
    "DeviceName": "light1",  
    "DeviceAlias": "light1",  
    "FamilyId": "",  
    "FamilyName": "",  
    "RelateUserID": "2",  
    "CreateAt": "2019-11-27T11:51:44.492Z"  
  }]  
,  
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"  
}  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 删除消息

最近更新时间：2023-07-14 16:01:49

## 1. 接口描述

接口请求域名： `iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（`AppDeleteMessage`）用于删除消息。

## 2. 输入参数

名称	类型	必选	描述
<code>AccessToken</code>	<code>String</code>	是	公共参数， <code>AccessToken</code> 用于对一个已经登录的用户鉴权。
<code>RequestId</code>	<code>String</code>	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需提供该次请求的 <code>RequestId</code> 。
<code>Action</code>	<code>String</code>	是	公共参数，本接口取值： <code>AppDeleteMessage</code> 。
<code>MsgID</code>	<code>String</code>	是	消息 ID。

## 3. 输出参数

名称	类型	描述
<code>RequestId</code>	<code>String</code>	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 <code>RequestId</code> 。
<code>Data</code>	<code>String</code>	返回数据。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Action": "AppDeleteMessage",
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
  "AccessToken": "8b*****b8ec",
  "MsgID": 1633328834809856
```

}

### 输出示例：成功

```
{  
  "Response": {  
    "Data": "OK",  
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"  
  }  
}
```

### 输出示例：失败

```
{  
  "Response": {  
    "Error": {  
      "Code": "InvalidParameterValue.InvalidAccessToken",  
      "Message": "Token无效"  
    },  
    "RequestId": ""  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.MsgIDInvalid	消息 ID 非法。

# 手动智能 创建手动智能联动

最近更新时间：2024-11-14 15:47:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（AppCreateScene）用于创建手动智能联动。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppCreateScene。
FamilyId	String	是	手动智能联动所属家庭 ID。
SceneName	String	是	手动智能联动的名称。
SceneIcon	String	是	手动智能联动的背景图片地址。
Actions	Array of Action	是	手动智能联动动作列表。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
SceneId	String	手动智能联动 ID。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Actions": [
    {
      "ActionType": 0,
      "ProductId": "R320****0EU",
      "DeviceName": "df****yY",
      "Data": "{\"brightness\": 25}"
    }
  ],
  "AccessToken": "x****v2",
  "RequestId": "req_1",
  "Action": "AppCreateScene",
  "FamilyId": "f_9b30*****c3d9588fcc1",
  "SceneName": "name",
  "SceneIcon": "icon"
}
```

### 输出示例: 成功

```
{
  "Response": {
    "SceneId": "a_cd61ddbe****47d21d6aeffea",
    "RequestId": "req_1"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

InvalidParameterValue.NoPermission

用户对该设备无权限。

# 获取手动智能联动列表

最近更新时间：2024-11-14 15:47:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（AppGetSceneList）用于获取手动智能联动列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetSceneList。
FamilyId	String	是	家庭 ID。
Offset	Int	是	消息偏移量。
Limit	Int	是	最大返回消息条数，最大值为 50。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
SceneList	Array of <code>Scene</code>	手动智能联动列表。
Total	Int	手动智能联动总数。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
```

```
{  
    "AccessToken": "****v2",  
    "RequestId": "req_1",  
    "Action": "AppGetSceneList",  
    "FamilyId": "f_9b309*****4c3d9588fcc1",  
    "Offset": 0,  
    "Limit": 10  
}
```

## 输出示例：成功

```
{  
    "Response": {  
        "RequestId": "req_1",  
        "SceneList": [  
            {  
                "SceneId": "s_f7feb440d*****28f3a598d00c",  
                "FamilyId": "f_9b309d8*****a11b4c3d9588fcc1",  
                "SceneName": "f****na",  
                "SceneIcon": "",  
                "Actions": [ // Actions数组中的每项的字段，为空则不出现  
                    {  
                        "ActionType": 0,  
                        "ProductId": "R32****0EU",  
                        "DeviceName": "df****yY",  
                        "Data": "{\"brightness\": 25}",  
                        "AliasName": "1****45" // 说明： AliasName为设备别名 。  
                    },  
                    {  
                        "ActionType": 1,  
                        "Data": "100"  
                    }  
                ],  
                "UserId": "1",  
                "CreateTime": 1578557536,  
                "UpdateTime": 1578560598,  
                "Flag": 0, // Flag : 0 为正常的场景，1：场景中的设备异常，不在家庭中  
                "Status": 0 // 场景运行状态 0未运行，1运行中  
            }  
        ]  
    }  
}
```

```
    ],
    "Total": 1
}
}

// IconUrl为具体设备的url
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。
UnsupportedOperation.SceneRunInProgress	场景执行中。

# 删除手动智能联动

最近更新时间：2024-10-11 16:01:50

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（`AppDeleteScene`）用于删除手动智能联动。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppDeleteScene</code> 。
SceneId	String	是	手动智能联动 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "AccessToken": "****v2",
  "RequestId": "req_1",
  "SceneId": "s_527cb52*****b453cb6a46f653",
  "Action": "AppDeleteScene"
}
```

输出示例：成功

```
{ "Response": { "RequestId": "req_1" } }
```

## 5. 错误码

错误码	描述
InternalError	内部错误
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。
UnsupportedOperation.SceneRunInProgress	场景执行中。

# 修改手动智能联动

最近更新时间：2024-11-14 15:47:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（AppModifyScene）用于修改手动智能联动。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppModifyScene。
SceneId	String	是	手动智能联动 ID。
SceneName	String	是	手动智能联动的名称。
SceneIcon	String	是	手动智能联动的背景图片地址。
Actions	Array of Action	是	手动智能联动动作列表。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
```

```
{  
  "Actions": [  
    {  
      "ActionType": 0,  
      "ProductId": "R32*****0EU",  
      "DeviceName": "df*****yY",  
      "Data": "{\"brightness\": 25}"  
    }  
  ],  
  "AccessToken": "****v2",  
  "RequestId": "req_1",  
  "Action": "AppModifyScene",  
  "SceneId": "s_527cb5*****3cb6a46f653",  
  "SceneName": "name",  
  "SceneIcon": "*****icon"  
}
```

### 输出示例: 成功

```
{"Response": {"RequestId": "req_1"}}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。
UnsupportedOperation.SceneRunInProgress	场景执行中。

# 执行手动智能联动

最近更新时间：2024-10-11 16:01:50

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（AppRunScene）用于执行手动智能联动。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppRunScene。
SceneId	String	是	手动智能联动 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "AccessToken": "****v2",
  "RequestId": "req_1",
  "SceneId": "s_527cb5*****53cb6a46f653",
  "Action": "AppRunScene"
}
```

输出示例：成功

```
{ "Response": { "RequestId": "req_1" } }
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限
UnsupportedOperation.SceneRunInProgress	场景执行中。

# 自动智能 创建自动智能联动

最近更新时间：2024-10-11 16:01:50

## 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/appapi](https://iot.cloud.tencent.com/api/exploreropen/appapi)。

本接口（AppCreateAutomation）用于创建自动智能联动。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppCreateAutomation。
FamilyId	String	是	自动智能联动所属家庭 ID。
Icon	String	是	自动智能联动的背景图片地址。
Name	String	是	自动智能联动的名称。
Status	int	是	自动智能联动开关状态。 <ul style="list-style-type: none"><li>0：关闭。</li><li>1：启用。</li></ul>
MatchType	int	是	条件匹配类型。 <ul style="list-style-type: none"><li>0：全部条件满足。</li><li>1：其中一个条件满足。</li></ul>
Conditions	Array of Condition	是	自动智能联动触发条件。
Actions	Array of Action	是	自动智能联动动作列表。

EffectiveBeginTime	String	是	生效开始时间。
EffectiveEndTime	String	是	生效结束时间。
EffectiveDays	String	是	生效日期由0和1组成的7位数字，0表示不执行，1表示执行，第1位为周日，依次表示周一至周六。

### 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
AutomationId	String	自动智能联动 ID。

### 4. 示例

#### 输入示例

```

POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
    "RequestId": "req_1",
    "Action": "AppCreateAutomation",
    "FamilyId": "f_9*****c1",
    "Icon": "https://www.icon.com/",
    "Name": "autoName",
    "Status": 0, //联动的开关 0: 关闭 1: 启用
    "MatchType": 1, //条件匹配类型 0: 条件全部与 1: 条件全部或
    "EffectiveBeginTime": "00:00", // 【两个新增参数，用来表示开始和结束时间】
    "EffectiveEndTime": "23:59",
    "EffectiveDays": "1111111", // 由0和1组成的7位数字，0表示不执行，1表示执行，第1位为周日，依次表示周一至周六
    "Conditions": [
        {
            "CondId": "abc", //条件ID，保证在一个联动下唯一即可
            "CondType": 0, //条件类型 0: 设备属性值条件 1: 定时条件
            "Property": {

```

```
        "ProductId": "N****4",
        "DeviceName": "g***ev",
        "AliasName": "devName",
        "IconUrl": "productIcon",
        "PropertyId": "switch", //设备的属性Id
        "Op": "eq", //条件操作符 eq 等于 ne 不等于 gt 大于 lt 小
        "ge": 大于等于
        "le": 小于等于
        "Value": 1 //比较的值
    }
},
{
    "CondId": "efg",
    "CondType": 1,
    "Timer": {
        "Days": "1000000", // 由0和1组成的7位数字, 0表示不执行, 1表示
        执行, 第1位为周日, 依次表示周一至周六
        "TimePoint": "09:30" // 触发时间, 24小时制, 例如"14:00"
    }
},
],
"Actions": [
{
    "ActionType": 0, // 0: 则为设备动作, 具体参数设置为Data 0 : 设
    备动作 1: 延时 2: 场景 3: 通知
    "ProductId": "R32***0EU",
    "DeviceName": "df***yY",
    "Data": "{\"brightness\": 25}", // ActionType为 0 : 设备动作参
    数 为1: 延时时间单位秒 2: 执行场景Id 3: 通知内容
    "AliasName": "12345",
    "IconUrl": "url"
},
{
    "ActionType": 1, //1: 则为延时动作, 具体延时时间为Data, 值为秒
    "Data": "100"
},
],
"AccessToken": "***v2"
}
```

## 输出示例：成功

```
{  
  "Response": {  
    "AutomationId": "a_cd61dd*****d21d6aeffea",  
    "RequestId": "req_1"  
  }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。
LimitExceeded	数量超过限制。

# 获取自动智能联动列表

最近更新时间：2024-10-11 16:01:50

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（AppGetAutomationList）用于获取自动智能联动列表。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetAutomationList。
FamilyId	String	是	家庭 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
List	Array of AutomationListItem	自动智能联动列表。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppGetAutomationList",
  "FamilyId": "f_9*****c1",
```

```
        "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec"  
    }  
}
```

### 输出示例: 成功

```
{  
    "Response": {  
        "List": [  
            {  
                "AutomationId": "a_cd61*****d21d6aeffea",  
                "Icon": "https://www.****.com/",  
                "Name": "autoName",  
                "Status": 0  
            }  
        ],  
        "RequestId": "req_1"  
    }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限

# 获取自动智能联动详情

最近更新时间：2024-10-11 16:01:50

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（AppDescribeAutomation）用于获取自动智能联动详情。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppDescribeAutomation。
AutomationId	String	是	自动智能联动 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId
Data	Object of Automation	自动智能联动实体

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppDescribeAutomation",
  "AutomationId": "a_cd61dd*****7d21d6aeffea",
```

```
"AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec"  
}
```

## 输出示例: 成功

```
{  
  "Response": {  
    "Data": {  
      "AutomationId": "a_cd61d*****d6aeffea",  
      "Icon": "https://www.****.com/",  
      "Name": "autoName",  
      "Status": 0,  
      "MatchType": 1,  
      "EffectiveBeginTime": "00:00", // 【两个新增参数, 用来表示开始和结束时  
间】  
      "EffectiveEndTime": "23:59",  
      "EffectiveDays": "1111111", // 由0和1组成的7位数字, 0表示不执行, 1表示执  
行, 第1位为周日, 依次表示周一至周六  
      "Conditions": [  
        {  
          "CondId": "abc",  
          "CondType": 0,  
          "Property": {  
            "ProductId": "NJ****LZT4",  
            "DeviceName": "gw****v",  
            "AliasName": "d****me",  
            "IconUrl": "productIcon",  
            "PropertyId": "switch",  
            "Op": "eq",  
            "Value": 1  
          }  
        },  
        {  
          "CondId": "efg",  
          "CondType": 1,  
          "Timer": {  
            "Days": "1000000",  
            "TimePoint": "09:30"  
          }  
        }  
      ]  
    }  
  }  
}
```

```
        }
    ],
    "Actions": [
        {
            "ActionType": 0,
            "ProductId": "R3*****0EU",
            "DeviceName": "df*****yY",
            "Data": "{\"brightness\": 25}",
            "AliasName": "12345",
            "IconUrl": "url"
        },
        {
            "ActionType": 1,
            "Data": "100"
        }
    ],
    "RequestId": "req_1"
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。

# 删除自动智能联动

最近更新时间：2025-12-19 09:16:00

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（AppDeleteAutomation）用于删除自动智能联动。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppDeleteAutomation。
AutomationId	String	是	自动智能联动 ID。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppDeleteAutomation",
  "AutomationId": "a_cd61d*****1d6aeffea",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec"
}
```

输出示例：成功

```
{ "Response": { "RequestId": "req_1" } }
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。

# 修改自动智能联动

最近更新时间：2024-10-11 16:01:50

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（AppModifyAutomation）用于修改自动智能联动。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppModifyAutomation。
AutomationId	String	是	自动智能联动 ID。
Icon	String	是	新建自动智能联动的背景图片地址。
Name	String	是	自动智能联动的名称。
Status	int	是	自动智能联动开关状态。 <ul style="list-style-type: none"><li>0：关闭。</li><li>1：启用。</li></ul>
MatchType	int	是	条件匹配类型。 <ul style="list-style-type: none"><li>0：全部条件满足。</li><li>1：其中一个条件满足。</li></ul>
Conditions	Array of Condition	是	自动智能联动触发条件。
Actions	Array of Action	是	自动智能联动动作列表。
EffectiveBe	String	是	生效开始时间。

ginTime			
EffectiveEndTime	String	是	生效结束时间。
EffectiveDays	String	是	生效日期 由0和1组成的7位数字，0表示不执行，1表示执行，第1位为周日，依次表示周一至周六。

### 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

### 4. 示例

#### 输入示例

```

POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
  "Action": "AppCreateAutomation",
  "AutomationId": "a_cd61ddbe*****47d21d6aeffea",
  "Icon": "https://www.****.com/",
  "Name": "autoName",
  "Status": 0, //联动的开关 0: 关闭 1: 启用
  "MatchType": 1, //条件匹配类型 0: 条件全部与 1: 条件全部或
  "EffectiveBeginTime": "00:00", // 【两个新增参数，用来表示开始和结束时间】
  "EffectiveEndTime": "23:59",
  "EffectiveDays": "1111111", // 由0和1组成的7位数字，0表示不执行，1表示执行，第1位为周日，依次表示周一至周六
  "Conditions": [
    {
      "CondId": "abc", //条件ID，保证在一个联动下唯一即可
      "CondType": 0, //条件类型 0: 设备属性值条件 1: 定时条件
      "Property": {
        "ProductId": "N****4",
        "DeviceName": "gw**v",
        "AliasName": "devName",
        "IconUrl": "productIcon",
      }
    }
  ]
}

```

```
        "PropertyId": "switch", //设备的属性Id
        "Op": "eq", //条件操作符 eq 等于 ne 不等于 gt 大于 lt 小
        "ge": 大于等于, "le": 小于等于
        "Value": 1 //比较的值
    },
},
{
    "CondId": "efg",
    "CondType": 1,
    "Timer": {
        "Days": "1000000", // 由0和1组成的7位数字, 0表示不执行, 1表示
        执行, 第1位为周日, 依次表示周一至周六
        "TimePoint": "09:30" // 触发时间, 24小时制, 比如"14:00"
    }
},
],
"Actions": [
    {
        "ActionType": 0, // 0: 则为设备动作, 具体参数设置为Data 0 : 设
        备动作 1: 延时 2:场景 3: 通知
        "ProductId": "R3****0EU",
        "DeviceName": "df****yY",
        "Data": "{\"brightness\": 25}", // ActionType为 0 : 设备动作参
        数 为1: 延时时间单位秒 2: 执行场景Id 3:通知内容
        "AliasName": "12345",
        "IconUrl": "url"
    },
    {
        "ActionType": 1, //1: 则为延时动作, 具体延时时间为Data, 值为秒
        "Data": "100"
    }
],
"AccessToken": "***v2"
}
```

## 输出示例: 成功

```
{"Response": {"RequestId": "req_1"}}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限。
LimitExceeded	数量超过限制。

# 修改自动智能联动状态

最近更新时间：2024-10-11 16:01:50

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/appapi`。

本接口（AppModifyAutomationStatus）用于修改自动智能联动状态。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后，获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值： <code>AppModifyAutomationStatus</code> 。
AutomationId	String	是	自动智能联动 ID。
Status	int	是	自动智能联动开关状态。 <ul style="list-style-type: none"><li>0：关闭。</li><li>1：启用。</li></ul>

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "req_1",
```

```
"Action": "AppGetAutomationList",
"AutomationId": "a_cd61dd*****1d6aeffea",
"Status": 1, //联动的开关 0: 关闭 1: 启用
"AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec"
}
```

**输出示例：成功**

```
{ "Response": { "RequestId": "req_1" } }
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InternalError.InternalServerExceptionDB	DB 错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.NoPermission	用户对该设备无权限

# 数据查询

## 获取设备物模型数据

最近更新时间：2024-10-11 16:01:51

### 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetDeviceData）用于获取设备物模型数据。

### 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetDeviceData。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名。

### 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
Data	String	返回数据。

### 4. 示例

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
```

```
{  
    "RequestId": "req_1",  
    "Action": "AppGetDeviceData",  
    "ProductId": "22F9Y****O",  
    "DeviceName": "light1",  
    "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec"  
}
```

## 输出示例: 成功

```
{  
    "Response": {  
        "Data": "{\"brightness\":  
        {\"Value\":8,\"LastUpdate\":1552904784552},\"color\":  
        {\"Value\":1,\"LastUpdate\":1552904784552},\"light_switch\":  
        {\"Value\":0,\"LastUpdate\":1552904784552},\"name\":  
        {\"Value\":\"light1\",\"LastUpdate\":1552904784552}}}",  
        "RequestId": "req_1"  
    }  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

## 6. 错误信息

名称	类型	描述
Code	String	错误码。
Message	String	错误信息

# 获取设备物模型历史数据

最近更新时间：2025-12-29 15:26:14

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口（AppGetDeviceDataHistory）用于获取设备物模型历史数据。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppGetDeviceDataHistory。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名。
FieldName	String	是	查询的属性名称。
Context	String	否	翻页游标，首次查询时，可不带。
MinTime	Int	是	开始时间，毫秒时间戳。
MaxTime	Int	是	结束时间，毫秒时间戳。
Limit	Int	是	单页数据量。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

Context	String	翻页游标。
FieldName	String	查询的属性名称。
Listover	Bool	数据是否查询完。 <ul style="list-style-type: none"><li>True 表示未读取完可继续读取。</li><li>False 表示已全部读取完。</li></ul>
Results	Array of <a href="#">DataHistoryItem</a>	返回数据。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
  "Action": "AppGetDeviceDataHistory",
  "ProductId": "22F9Y6II70",
  "DeviceName": "light1",
  "MinTime": 1560949017000,
  "MaxTime": 1561035417000,
  "Limit": 20,
  "FieldName": "light_switch",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec"
}
```

#### 输出示例：成功

```
{
  "Response": {
    "Context": "abc",
    "FieldName": "brightness",
    "Listover": false,
```

```
"RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
"Results": [
  {
    "Time": "1560949025494",
    "Value": "77"
  }
]
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 获取设备的历史事件

最近更新时间：2024-10-11 16:01:51

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`。

本接口(`AppListEventHistory`)用于获取设备的历史事件。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，用户通过微信号、手机或邮箱账号登录成功后,获取的访问 Token。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	AppListEventHistory。
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
Type	String	否	搜索的事件类型：alert 表示告警，fault 表示故障，info 表示信息，为空则表示查询上述所有类型事件。
StartTime	Int64	否	起始时间（Unix 时间戳，秒级），为0表示当前时间。
EndTime	Int64	否	结束时间（Unix 时间戳，秒级），为0表示当前时间。
Context	String	否	搜索上下文，用作查询游标。
Size	Int64	否	单次获取的历史数据项目的最大数量。
EventId	String	否	事件标识符，可以用来指定查询特定的事件，如果不指定，则查询所有事件。

## 3. 输出参数

名称	类型	描述

RequestID	String	公共参数，唯一请求 ID，与入参相同。
Context	String	搜索上下文，用作查询游标。
Total	Int64	搜索结果数量。
Listover	Bool	搜索结果是否已经结束。
EventHistory	Array	搜集结果集。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "AccessToken": "accesstoken*****acc",
  "RequestId": "026fe48b-0c31-4a7c-aaf6-83ba3e9s4bcf",
  "ProductID": "ABCDE12345",
  "DeviceName": "DEV",
  "StartTime": 0,
  "EndTime": 0,
  "Type": "info",
  "Size": 5,
  "Context": "",
  "Action": "AppListEventHistory"
}
```

#### 输出示例：成功

```
{
  "Response": {
    "ClientToken": "1ee12*****7df703e0d",
    "OutputParams": "",
    "RequestId": "026fe48b-0c31-4a7c-aaf6-83ba3e9s4bcf",
    "Result": [
      {
        "EventID": "00000000000000000000000000000000",
        "EventTime": "2018-01-01T00:00:00Z",
        "EventName": "DeviceEvent",
        "EventContent": "DeviceEvent content"
      }
    ]
  }
}
```

```
"Context":  
"DnF1ZXJ5VGhIbkZld*****SUQ0UWVPZUlMLXI1QnBkbGc=",  
"EventHistory": [ {  
    "TimeStamp": 1553484903,  
    "ProductId": "product1",  
    "DeviceName": "light3",  
    "EventId": "status_report",  
    "Type": "info",  
    "Data": "{\"status\": 0, \"message\": \"test  
message:11:35:03.288277\"}"  
},  
    "Total": 1  
}  
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误
InvalidParameterValue.InvalidAccessToken	Token 无效。

# 长连接通信

## 注册监听

最近更新时间：2025-06-03 14:30:02

### 1. 接口描述

接口请求域名： `wss://iot.cloud.tencent.com/iotstudio_v2_weapp_1`

本接口（`DeviceRegistrationMonitor`）用于向 WS 服务端进行订阅，订阅成功后可以通过 `ws.onmessage` 的监听获取到 WS 服务端实时推送设备上下线状态及属性数据。

### 2. 输入参数

名称	类型	必选	描述
<code>action</code>	<code>String</code>	是	公共参数，本接口取值： <code>ActivePush</code> 。
<code>reqId</code>	<code>String</code>	是	公共参数，标示请求序列，唯一请求 ID，可自行生成。
<code>params</code>	<code>Object of ActionPushParam</code>	是	注册设备的时候传入的参数。

### 3. 输出参数

名称	类型	描述
<code>reqId</code>	<code>String</code>	公共参数，标示请求序列，与入参相同。
<code>error</code>	<code>String</code>	状态码，为空表示正确。
<code>error_message</code>	<code>String</code>	错误详细说明，为空表示正确。
<code>data</code>	<code>Object of ActionPushData</code>	监听之后的服务器回包。

### 4. 示例

#### 示例1

输入示例（发送前需将数据转换为 json 字符串）

```
const ws = new
WebSocket("wss://iot.cloud.tencent.com/iotstudio_v2_weapp_1")

ws.send(JSON.stringify({
  "action": "ActivePush",
  "reqId": "req0001",
  "params": {
    "DeviceIds": ["产品ID/DeviceName", "产品ID/DeviceName"],
    "AccessToken": "f6*****3aad"
  }
}));
```

### 输出示例：成功

```
{
  "error": "",
  "error_message": "",
  "data": {
    "deviceIds": ["产品ID/DeviceName", "产品ID/DeviceName"]
  },
  "reqId": "req0001"
}
```

### 输出示例：失败

```
{
  "error": "",
  "error_message": "",
  "data": {
    "deviceIds": []
  },
  "reqId": "req0001"
}
```

# 心跳

最近更新时间：2025-06-03 14:30:03

## 1. 接口描述

接口请求域名： `wss://iot.cloud.tencent.com/iotstudio_v2_weapp_1`

设备保活，通过该接口更新设备时间戳，保持设备监听状态，同时保持 WebSocket 连接。可以多次调用，每60s 调用一次。

## 2. 输入参数

名称	类型	必选	描述
<code>action</code>	<code>String</code>	是	公共参数，本接口取值：YunApi。
<code>reqId</code>	<code>String</code>	是	公共参数，标示请求序列，唯一请求 ID，可自行生成。
<code>params</code>	<code>Object</code>	是	调用接口传入的详细参数，请参见 <a href="#">YunApiParam</a> 。

## 3. 输出参数

名称	类型	描述
<code>reqId</code>	<code>String</code>	公共参数，表示请求序列，与入参相同。
<code>error</code>	<code>String</code>	状态码,为空表示正确。
<code>error_message</code>	<code>String</code>	错误详细说明，为空表示正确。
<code>data</code>	<code>Object</code>	心跳接口的服务器回包，请参见 <a href="#">HeartBeatData</a> 。

## 4. 示例

输入示例（发送前需将数据转换为 json 字符串）

```
const ws = new
WebSocket("wss://iot.cloud.tencent.com/iotstudio_v2_weapp_1")

ws.send(JSON.stringify({
  "action": "YunApi",
  "reqId": "req0001",
  "params": {
```

```
        "Action": "AppDeviceTraceHeartBeat",
        "ActionParams": {
            "DeviceIds": ["产品ID/DeviceName", "产品ID/DeviceName"],
            "AccessToken": "f6*****3aad",
            "RequestId": "rest-client"
        }
    }
})
```

### 输出示例：成功

```
{
    "error": "",
    "error_message": "",
    "data": {
        "Response": {
            "RequestId": "rest-client"
        }
    },
    "reqId": "req0001"
}
```

### 输出示例：失败（服务器错误）

```
{
    "error": "-1",
    "error_message": "后台连接出错，请联系管理员",
    "data": {},
    "reqId": "req0001"
}
```

### 输出示例：失败（逻辑错误）

```
{
    "error": "",
    "error_message": "",
    "data": {
        "Response": {
            "Error": {

```

```
    "Code":  
    "UnauthorizedOperation.APPNoPermissionToStudioProduct",  
    "Message": "App对操作该产品无权限"  
},  
    "RequestId": "rest-client"  
}  
,  
    "reqId": "req0001"  
}
```

# 设备状态推送

最近更新时间：2024-11-14 15:47:32

## 1. 接口描述

接口请求域名： `wss://iot.cloud.tencent.com/iotstudio_v2_weapp_1`

设备状态推送用于在小程序或 App 实时获取用户绑定设备的上下线状态、设备上报的属性与事件以及设备行为执行结果。需要成功调用注册设备监听接口，通过监听 WebSocket 的 `OnMessage` 获取设备状态与属性值的实时推送，若获取到的 `event.data.push` 为 `true`，则代表该条消息为设备状态变更的主动推送。

## 2. 输入参数

名称	类型	描述
<code>push</code>	<code>Boolean</code>	该条消息为设备状态变更的主动推送。
<code>action</code>	<code>String</code>	固定为 <code>DeviceChange</code> ，代表设备状态变更。
<code>params</code>	<code>String</code>	返回设备信息，详细说明见 <a href="#">消息类型说明</a> 。

## 3. 输出参数

名称	类型	描述
<code>RequestId</code>	<code>String</code>	请求 ID。

## 4. 示例

### 应用端 ws 订阅设备状态变化

#### 请求示例

```
ws.onmessage = (event) => {
  let data;
  try {
    data = JSON.parse(event.data);
  } catch (e) {
    console.log(`onMessage parse event.data error: ${event.data}`);
    return;
  }

  if (data.push) {
```

```
        this.emit('push', data);
    } else if (typeof data.reqId !== 'undefined' &&
this.requestHandlerMap.has(data.reqId)) {
    this.requestHandlerMap.get(data.reqId)(null, data);
}
};
```

### 输入参数示例：

```
{
    "action": "DeviceChange",
    "params": {
        "Time": "2023-07-13T14:19:48+08:00",
        "Type": "Property",
        "SubType": "Report",
        "Topic": "$tThing/up/property/productId/deviceName",
        "Payload": "eyJ*****i0jQwfX0=",
        "Seq": 1689229188406,
        "DeviceId": "productId/deviceName"
    },
    "push": true
}
```

### 输出示例：成功

```
{
    "Response": {
        "RequestId": "up03"
    }
}
```

## 5. params 参数说明

参数名称	参数类型	说明
Time	String	时间，UTC 格式，如：2023-07-13T14:19:48+08:00。
Type	String	消息类型，详见 <a href="#">Type 类型说明</a> 。

SubType	String	消息子类型，详见 <a href="#">SubType 类型说明</a> 。
Topic	String	消息 Topic，可登录 <a href="#">物联网开发平台控制台</a> 进入目标产品开发页，选择设备开发 > Topic 列表来查看。
Payload	String	消息体 base64 编码内容。详见 <a href="#">Payload 说明</a> 。
Seq	Int64	序列号。
DeviceId	String	设备 ID，产品 ID/deviceName 格式。

## Type 类型说明

Type 类型	描述
StatusChange	设备状态变化消息，包括设备上线与下线消息通知。
Shadow	影子消息。
Property	属性消息。
Template	物模型消息。
Event	事件消息。
Action	行为消息。
UserDefined	用户自定义。

## SubType 说明

SubType 类型	描述
Online	设备上线。
Offline	设备下线。
Report	设备上报。
Push	服务端推送到设备。

## Payload 说明

**Payload** 是消息体 base64 编码后字符串。示例如下：

### 消息体：

```
{  
  "method": "control",  
  "clientToken": "12345678",  
  "params": {  
    "switch": 0  
  }  
}
```

### base64编码：

```
"Payload":  
"eyJtZXRob2QiOiJjb250cm9sIiwiY2xpZW50VG9rZW4iOiIxMjM0NTY3OCIsInBhcmFtcyI  
6eyJzd2l0Y2giOjB9fQ=="
```

## Template 消息示例

物模型消息 Topic 为空，Payload 是消息 base64 编码后内容。

```
{  
  "action": "DeviceChange",  
  "params": {  
    "Time": "2023-07-13T15:00:11+08:00",  
    "Type": "Template",  
    "SubType": "Push",  
    "Topic": "",  
    "Payload":  
"eyJtZXRob2QiOiJjb250cm9sIiwiY2xpZW50VG9rZW4iOiIxMjM0NTY3OCIsInBhcmFtcyI  
6eyJzd2l0Y2giOjB9fQ==",  
    "Seq": 1689231611907,  
    "DeviceId": "productId/deviceName"  
  },  
  "push": true  
}
```

## Payload base64 解码消息体

详细说明见 [物模型](#)。

```
{  
  "method": "control",  
  "clientToken": "12345678",  
  "params": {  
    "switch": 0  
  }  
}
```

## Event 消息示例

```
{  
  "Time": "2019-04-11T20:42:36+08:00",  
  "Type": "Event",  
  "SubType": "Report",  
  "Topic": "$thing/up/event/US4CJ11DIK/LIAO",  
  "Payload":  
    "ewogICAgICAgICJtZXRob2QiOiAiZXZlbnRfcG9zdCIsciaGICAgICAgInZlcnPb24iOia  
    iMS4wIiwKICAgICAgICAiZXZlbnRJZCI6ICJsb3dfdm9sdGFnZSIsciaGICAgICAgInBhcmF  
    tcyI6IHsKICAgICAgICAgICAgInZvbHRhZ2UiOiAyLjI3ODA4MQogICAgICAgIH0sCiAgICA  
    gICAgImNsaWVudFRva2VuIjogImNsaWVudFRva2VuLTU1OSJ9",  
  "Seq": 300,  
  "DeviceId": "US4CJ11DIK/LIAO"  
}
```

## Payload base64 解码消息体

```
{  
  "method": "event_post",  
  "version": "1.0",  
  "eventId": "low_voltage",  
  "params": {  
    "voltage": 2.278081  
  },  
  "clientToken": "clientToken-559"  
}
```

## 属性消息示例

```
{  
  "Time": "2019-10-30T10:41:00+08:00",  
  "Type": "Property",  
  "SubType": "Report",  
  "Topic": "$thing/up/property/WTJZX0MF2/device_property001",  
  "Payload":  
    "eyJ0aW1lc3RhbXAiOiAwLCAicGFyYW1zIjogeyJicmlnaHRuZXNzIjogNTl9LCAibWV0aG9  
    kIjogInJlcG9ydCIsICJjbG1bnRUb2tlbiI6ICJ0bTIwMTkxMDMwMTA0MDUzXzEifQ==",  
  "Seq": 6,  
  "DeviceId": "WTJZX0MF2/device_property001"  
}
```

## Payload base64 解码消息体

```
{  
  "method": "report",  
  "params": {  
    "timestamp": 0,  
    "params": {  
      "brightness": 59  
    },  
    "method": "report",  
    "clientToken": "tm20191030104053_1"  
  }  
}
```

## StatusChange 消消息示例

设备上线消息。

```
{  
  "Time": "2019-03-28T15:17:33+08:00",  
  "Type": "StatusChange",  
  "SubType": "Online",  
  "Topic": "",  
  "Payload": "",  
  "Seq": 6  
}
```

设备下线。

```
{  
  "Time": "2019-03-28T15:17:33+08:00",  
  "Type": "StatusChange",  
  "SubType": "Offline",  
  "Topic": "",  
  "Payload": "",  
  "Seq": 7  
}
```

## Action 行为示例

### Action 消息推送示例

```
{  
  "Time": "2021-05-13T21:34:05+08:00",  
  "Type": "Action",  
  "SubType": "Push",  
  "Topic": "",  
  "Payload":  
    "eyJtZXRob2QiOiJhY3Rpb24iLCJjbGllbnRUb2tlbiI6IjEzNjk0NzQ5Mzo6cmVxXzEiLCJ  
    hY3Rpb25JZCI6ImxpZ2h0X2JsaW5rIiwidGltZXN0YW1wIjoxNjIwOTEyODQ1LCJwYXJhbXM  
    iOnsiY29sb3IiOjEsInRpBWUiOjYsInRvdGFsX3RpBWUiOjYwfX0K",  
  "Seq": 1620912845,  
  "DeviceId": "0CP5FIRP28/testble"  
}
```

### Payload base64 解码消息体

```
{  
  "method": "action",  
  "clientToken": "136947493::req_1",  
  "actionId": "light_blink",  
  "timestamp": 1620912845,  
  "params": {  
    "color": 1,  
    "time": 6,  
    "total_time": 60  
}
```

```
    }  
}
```

## Action 消息 report 示例

```
{  
    "Time": "2021-05-13T21:34:05+08:00",  
    "Type": "Action",  
    "SubType": "Report",  
    "Topic": "",  
    "Payload":  
        "eyJtZXRob2QiOiJhY3Rpb25fcmlCJjbG11bnRUb2tlbiI6IjEzNjk0NzQ5Mzo6cmV  
        xXzEiLCJjb2RlIjowLCJzdGF0dXMiOiJzdWNjIiwi cmVzcG9uc2UiOnsicmVzdWx0IjogMX1  
        9",  
        "Seq": 1620912845,  
        "DeviceId": "0CPSFIRP28/testble"  
}
```

## Payload base64 解码消息体

```
{  
    "method": "action_reply",  
    "clientToken": "136947493::req_1",  
    "code": 0,  
    "status": "succ",  
    "response": {  
        "result": 1  
    }  
}
```

# 云存服务

## 拉取云存事件列表

最近更新时间：2024-05-11 14:15:22

### 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)

本接口（`lotVideoDescribeCloudStorageEvents`）用于拉取云存事件列表。

### 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值： <code>lotVideoDescribeCloudStorageEvents</code> 。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需要提供该次请求的 <code>RequestId</code> 。
AccessToken	String	是	公共参数， <code>AccessToken</code> 用于对一个已经登录的用户鉴权。
ProductId	String	是	产品ID。
DeviceName	String	是	设备名称。
StartTime	Integer	否	起始时间（Unix 时间戳，秒级），为0 表示当前时间 - 24h
EndTime	Integer	否	结束时间（Unix 时间戳，秒级），为0 表示当前时间
Context	String	否	请求上下文，用作查询游标
Size	Integer	否	查询数据项目的最大数量，默认为10。假设传 <code>Size=10</code> ，返回的实际事件数量为N，则 $5 \leq N \leq 10$
EventId	String	否	事件标识符，可以用来指定查询特定的事件，如果不指定，则查询所有事件

UserId	String	否	用户ID
ChannelId	Integer	否	通道ID 非NVR设备则不填 NVR设备则必填 默认为无

### 3. 输出参数

名称	类型	描述
RequestId	String	RequestId。
Events	Array of CloudStorageEvent	云存事件列表
Context	String	请求上下文, 用作查询游标
Listover	Boolean	拉取结果是否已经结束
Total	Integer	内部结果数量, 并不等同于事件总数
VideoURL	String	视频播放URL

### 4. 示例

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Action": "AppDescribeTemplateBinding",
  "AccessToken": "bf*****d098",
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
  "ProductId": "PRODUCT_ID",
  "DeviceName": "DEVICENAME"
}
```

#### 输出示例：成功

{

```
"Response": {
    "RequestId": "bd3ad0f6-5b9f-47cd-83db-223356926772",
    "Total": 13,
    "Events": [
        {
            "StartTime": 842084401,
            "EndTime": 842084401,
            "Thumbnail": "",
            "EventId": "id1_data"
        },
        {
            "StartTime": 842084401,
            "EndTime": 842084401,
            "Thumbnail": "",
            "EventId": "id1_data"
        },
        {
            "StartTime": 842084401,
            "EndTime": 842084401,
            "Thumbnail": "",
            "EventId": "id1_data"
        },
        {
            "StartTime": 1614847697,
            "EndTime": 167000000,
            "Thumbnail": "face_dectcion.jpg",
            "EventId": "id1_data"
        },
        {
            "StartTime": 1614847697,
            "EndTime": 167000000,
            "Thumbnail": "face_dectcion.jpg",
            "EventId": "id1_data"
        },
        {
            "StartTime": 1614847697,
            "EndTime": 167000000,
            "Thumbnail": "face_dectcion.jpg",
            "EventId": "id1_data"
        },
        {
            "StartTime": 1614847697,
            "EndTime": 167000000,
            "Thumbnail": "face_dectcion.jpg",
            "EventId": "id1_data"
        }
    ]
}
```

```
{  
    "StartTime": 1614847697,  
    "EndTime": 167000000,  
    "Thumbnail": "face_dection.jpg",  
    "EventId": "id1_data"  
},  
{  
    "StartTime": 1614847697,  
    "EndTime": 167000000,  
    "Thumbnail": "face_dection.jpg",  
    "EventId": "id1_data"  
},  
{  
    "StartTime": 1614847697,  
    "EndTime": 167000000,  
    "Thumbnail": "face_dection.jpg",  
    "EventId": "id1_data"  
},  
{  
    "StartTime": 1614847697,  
    "EndTime": 167000000,  
    "Thumbnail": "face_dection.jpg",  
    "EventId": "id1_data"  
}  
],  
"Context":  
"DnF1ZXJ5VGh1bkZldGNoKAAAAAAABxjx1Fm1DUHBXME5XVE55eUdHMDUwZTV4WWcAAAAAAca  
q9BZFeGQwTXM5X1RvNnFRMFNhRm1MNT1RAAAAAAHGqvYWRXhkME1zOV9UbzZxUTBTYUztTDU  
5UQAAAAABxk0BFnN5Q1hWNUp0UjcyZkZma21icmF3SmcAAAAAAcaq9RZFeGQwTXM5X1RvNnF  
RMFNhRm1MNT1RAAAAAAHGhvUWTz12c3FBNVdUWnFSWXh4d012MGlzdwAAAAABxjxmFm1DUHB  
XME5XVE55eUdHMDUwZTV4WWcAAAAAAcaG9xZPOXZzcUE1V1RacVJZeHh3TXYwaXN3AAAAAAH  
GVbkWTVBJSndzX0VReVNoU3Job2pWUm40UQAAAAABxk0AFnN5Q1hWNUp0UjcyZkZma21icmF  
3SmcAAAAAAcaG--  
BZPOXZzcUE1V1RacVJZeHh3TXYwaXN3AAAAAAHGqvgWRXhkME1zOV9UbzZxUTBTYUztTDU5U  
QAAAAABxkfDFjdNdGYtaTR5UTBPNmFFcHkyUmlUVHcAAAAAAcY88hZOVnkwbjNfNFRqV2YwS  
Wk1LWczc2V3AAAAAAHGhvYWTz12c3FBNVdUWnFSWXh4d012MGlzdwAAAAABxqr3FkV4ZDBNc  
z1fVG82cVEwU2FGbUw1OVEAAAAAAcaG--  
RZPOXZzcUE1V1RacVJZeHh3TXYwaXN3AAAAAAHGR8IWN010Zi1pNH1RME82YUVweTJSaVRUd  
wAAAAABxquSFnJKTjhzbk5mUmZxMVVCcVhpVUNTZFEAAAAAAcY8ZxZpQ1BwVzBOV1ROeX1HR  
zA1MGU1eFlnAAAAAAHGehEWTRRa1JkRDhUek9qem1CSmNFbHhUZwAAAAABxqr5FkV4ZDBNc
```

```
z1fVG82cVEwU2FGbUw1OVEAAAAAAcZNBBZzeUNYVjVKdFI3MmZGZmtpYnJhd0pnAAAAAAHGV
boWTVBJSndzX0VReVNoU3JOb2pWUm40UQAAAAABxnoQFk00UWtSZEQ4VHpPanptQkpjRWx4V
GcAAAAAAcarkxZySk44c250Z1JmcTFVQnFYaVVDU2RRAAAAAHGTQIWc31DWFY1SnRSNzJmR
mZraWJyYXdKZwAAAAABxpZ4Fno4eVNTTD1LUVUyb1VfMFFFcnhramcAAAAAAcY88xZOVnkwb
jNfNFRqV2YwSWk1LWczc2V3AAAAAAHGTQMWC31DWFY1SnRSNzJmRmZraWJyYXdKZwAAAAABx
kfEFjdNdGYtaTR5UTBPNmFFcHkyUmlUVHcAAAAAAcarlBZySk44c250Z1JmcTFVQnFYaVVDU
2RRAAAAAAHgqvoWRXhkME1zOV9UbzZxUTBTYUztTDU5UQAAAAABxpZ6Fno4eVNTTD1LUVUyb
1VfMFFFcnhramcAAAAAAcZHxRY3TXRmLWk0eVEwTzZhRXB5M1JpVFR3AAAAAAHGVbsWTVBJS
ndzX0VReVNoU3JOb2pWUm40UQAAAAABxpZ5Fno4eVNTTD1LUVUyb1VfMFFFcnhramcAAAAAA
cY8aBZpQ1BwVzBOV1ROeX1HRzA1MGU1eFlnAAAAAAHgqvsWRXhkME1zOV9UbzZxUTBTYUztT
DU5UQAAAAABxpZ7Fno4eVNTTD1LUVUyb1VfMFFFcnhramc=",
    "Listover": false,
    "VideoURL": "http://test.iotvideo.tencentcs.com/timeshift/live/test.m3u8"
}
}
```

## 输出示例：失败

```
{
    "Response": {
        "Error": {
            "Code": "InvalidParameterValue.InvalidAccessToken",
            "Message": "Token无效"
        },
        "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"
    }
}
```

## 5. 错误码

错误码	描述
InvalidParameter	无效参数。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TokenInfoSaveError	Token 保存失败。

# 拉取云存事件缩略图

最近更新时间：2024-11-01 09:43:32

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`

本接口（`IoTVideoDescribeCloudStorageThumbnail`）用于拉取云存事件缩略图。

## 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值： <code>IoTVideoDescribeCloudStorageThumbnail</code> 。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需要提供该次请求的 <code>RequestId</code> 。
AccessToken	String	是	公共参数， <code>AccessToken</code> 用于对一个已经登录的用户鉴权。
ProductId	String	是	产品ID。
DeviceName	String	是	设备名称。
Thumbnail	String	是	缩略图文件名

## 3. 输出参数

名称	类型	描述
RequestId	String	<code>RequestId</code> 。
ThumbnailURL	String	缩略图访问地址

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Action": "AppDescribeTemplateBinding",
  "AccessToken": "bf*****d098",
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
  "ProductId": "PRODUCT_ID",
  "DeviceName": "DEVICENAME",
  "Thumbnail": "/100008401725/AQTV2839QJ/sp01_32820237_7/events/1615200613.jpg"
}
```

## 输出示例：成功

```
{
  "Response": {
    "RequestId": "c299d2c9-805f-41a5-853d-db32ba1a16b5",
    "ThumbnailURL": "https://video-cv-1258344699.cos.ap-
guangzhou.myqcloud.com/%2F100008401725%2FAQTV2839QJ%2Fsp01_32820237_5%2F
events%2F1614908216.jpg?q-sign-algorithm=sha1&q-
ak=*****&q-sign-time=1614909378%3B1614912978&q-
key-time=1614909378%3B1614912978&q-
header-list=host&q-url-param-list=&q-
signature=59cfa8d69bb77980ba757cb11de62fc6c4880f5"
  }
}
```

## 输出示例：失败

```
{
  "Response": {
    "Error": {
      "Code": "InvalidParameterValue.InvalidAccessToken",
      "Message": "Token无效"
    },
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"
  }
}
```

## 5. 错误码

错误码	描述
InvalidParameter	无效参数。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TokenInfoSaveError	Token 保存失败。

# 获取具有云存的日期

最近更新时间：2024-05-11 14:15:22

## 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)

本接口（`IoTVideoDescribeCloudStorageDate`）用于获取具有云存的日期。

## 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值： <code>IoTVideoDescribeCloudStorageDate</code> 。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需要提供该次请求的 <code>RequestId</code> 。
AccessToken	String	是	公共参数， <code>AccessToken</code> 用于对一个已经登录的用户鉴权。
ProductId	String	是	产品ID。
DeviceName	String	是	设备名称。
UserId	String	否	用户ID

## 3. 输出参数

名称	类型	描述
RequestId	String	<code>RequestId</code> 。
Data	Array of String	云存日期数组，示例值：["2021-01-05", "2021-01-06"]

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
```

```
content-type: application/json
{
  "Action": "AppDescribeTemplateBinding",
  "AccessToken": "bf*****d098",
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
  "ProductId": "PRODUCT_ID",
  "DeviceName": "DEVICENAME"
}
```

### 输出示例：成功

```
{
  "Response": {
    "RequestId": "77d90d07-e48c-4cf0-bec5-afe5b137f80e",
    "Data": [
      "2021-01-14"
    ]
  }
}
```

### 输出示例：失败

```
{
  "Response": {
    "Error": {
      "Code": "InvalidParameterValue.InvalidAccessToken",
      "Message": "Token无效"
    },
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"
  }
}
```

## 5. 错误码

错误码	描述
InvalidParameter	无效参数。
InvalidParameterValue	参数取值错误。

InvalidParameterValue.TokenInfoSaveE  
rror

Token 保存失败。

# 获取某一天云存时间轴

最近更新时间：2024-05-11 14:15:22

## 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)

本接口（`IoTVideoDescribeCloudStorageTime`）用于获取某一天云存时间轴。

## 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值： <code>IoTVideoDescribeCloudStorageTime</code> 。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需要提供该次请求的 <code>RequestId</code> 。
AccessToken	String	是	公共参数， <code>AccessToken</code> 用于对一个已经登录的用户鉴权。
ProductId	String	是	产品ID。
DeviceName	String	是	设备名称。
Date	String	是	云存日期，例如"2020-01-05"
StartTime	Integer	否	开始时间，unix时间
EndTime	Integer	否	结束时间，unix时间
UserId	String	否	用户ID

## 3. 输出参数

名称	类型	描述
RequestId	String	<code>RequestId</code> 。

Data	CloudStorageTimelineData	返回数据
------	--------------------------	------

## 4. 示例

### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "Action": "AppDescribeTemplateBinding",
  "AccessToken": "bf*****d098",
  "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
  "ProductId": "PRODUCT_ID",
  "DeviceName": "DEVICENAME",
  "Date": "2021-01-14"
}
```

### 输出示例：成功

```
{
  "Response": {
    "RequestId": "ac74ecca-b866-4ab1-87e0-18b92816fb20",
    "Data": [
      {
        "VideoURL": "47b9d579-9088-4cc2-a16e-2b8cf245319b",
        "TimeList": [
          {
            "StartTime": 1610607600,
            "EndTime": 1610608200
          },
          {
            "StartTime": 1610607600,
            "EndTime": 1610608200
          },
          {
            "StartTime": 1610607600,
            "EndTime": 1610608200
          }
        ]
      }
    ]
  }
}
```

```
        "StartTime": 1610607600,  
        "EndTime": 1610608200  
    }  
]  
}  
}  
}
```

### 输出示例：失败

```
{  
    "Response": {  
        "Error": {  
            "Code": "InvalidParameterValue.InvalidAccessToken",  
            "Message": "Token无效"  
        },  
        "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"  
    }  
}
```

## 5. 错误码

错误码	描述
InvalidParameter	无效参数。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TokenInfoSaveError	Token 保存失败。

# 获取视频防盗链播放 URL

最近更新时间：2025-12-19 09:17:03

## 1. 接口描述

接口请求域名：[iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)

本接口（`IoTVideoGenerateSignedVideoURL`）用于获取视频防盗链播放 URL。

## 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值： <code>IoTVideoGenerateSignedVideoURL</code> 。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需要提供该次请求的 <code>RequestId</code> 。
AccessToken	String	是	公共参数， <code>AccessToken</code> 用于对一个已经登录的用户鉴权。
ProductId	String	是	产品ID。
DeviceName	String	是	设备名称。
VideoURL	String	是	视频播放原始URL地址
ExpireTime	Integer	是	播放链接过期时间

## 3. 输出参数

名称	类型	描述
RequestId	String	<code>RequestId</code> 。
SignedVideoURL	String	视频防盗链播放URL

## 4. 示例

## 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
    "Action": "AppDescribeTemplateBinding",
    "AccessToken": "bf*****d098",
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
    "ProductId": "PRODUCT_ID",
    "DeviceName": "DEVICENAME",
    "ExpireTime": 1619331648,
    "VideoURL":
    "http://zylcb.iotvideo.tencentcs.com/timeshift/live/1.m3u8"
}
```

## 输出示例：成功

```
{
    "Response": {
        "SignedVideoURL":
        "http://zylcb.iotvideo.tencentcs.com/timeshift/live/1.m3u8?sign=x",
        "RequestId": "xx"
    }
}
```

## 输出示例：失败

```
{
    "Response": {
        "Error": {
            "Code": "InvalidParameterValue.InvalidAccessToken",
            "Message": "Token无效"
        },
        "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"
    }
}
```

## 5. 错误码

错误码	描述
InvalidParameter	无效参数。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TokenInfoSaveError	Token 保存失败。

# 拉取图片流数据

最近更新时间：2025-12-19 09:18:02

## 1. 接口描述

接口请求域名：`iot.cloud.tencent.com/api/exploreropen/tokenapi`

本接口（`IoTVideoDescribeCloudStorageStreamData`）用于拉取图片流数据。

## 2. 输入参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值： <code>IoTVideoDescribeCloudStorageStreamData</code> 。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 <code>uuid</code> 。定位问题时，需要提供该次请求的 <code>RequestId</code> 。
AccessToken	String	是	公共参数， <code>AccessToken</code> 用于对一个已经登录的用户鉴权。
ProductId	String	是	产品ID。
DeviceName	String	是	设备名称。
StartTime	Integer	是	起始时间

## 3. 输出参数

名称	类型	描述
RequestId	String	<code>RequestId</code> 。
VideoStream	String	图片流视频地址
AudioStream	String	图片流音频地址

## 4. 示例

## 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
    "Action": "AppDescribeTemplateBinding",
    "AccessToken": "bf*****d098",
    "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68",
    "ProductId": "PRODUCT_ID",
    "DeviceName": "DEVICENAME",
    "StartTime": 1619331648
}
```

## 输出示例：成功

```
{
    "Response": {
        "VideoStream": "http://xxxxxxxx",
        "AudioStream": "http://xxxxxxxx",
        "RequestId": "xx"
    }
}
```

## 输出示例：失败

```
{
    "Response": {
        "Error": {
            "Code": "InvalidParameterValue.InvalidAccessToken",
            "Message": "Token无效"
        },
        "RequestId": "f92406b3-5a9a-4fe8-bc43-45e3d794bb68"
    }
}
```

## 5. 错误码

错误码	描述
-----	----

InvalidParameter	无效参数。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.TokenInfoSaveError	Token 保存失败。

# 音乐服务

## 歌单歌曲

最近更新时间：2025-12-29 15:26:14

### 1. 接口描述

接口请求域名： [iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)

本接口（playlist\_song）用于获取歌单歌曲。

### 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppKugouUserCommand。
KGCommand	String	是	子命令，本接口取值：playlist_song。
DeviceId	String	是	设备 Id 由产品 ID 和设备 Name 组成。
KugouParams	Object of <a href="#">KugouParams_playlist_song</a>	是	子命令的参数。

### 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
ErrorCode	Integer	请求返回的错误码，0为没有错误。
ErrorMsg	String	请求返回的错误信息。

g		
Data	Object of <a href="#">Data_playlist_song</a>	返回结果。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "test0191119-012345",
  "Action": "AppKugouUserCommand",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "DeviceId": "LAEG4YJE1A/dev01",
  "KGCommand": "playlist_song",
  "KugouParams": { . . . }
}
```

#### 输出示例：成功

```
{
  "Response": {
    "Data": { . . . },
    "ErrorCode": 0, "ErrorMsg": "",
    "RequestId": "keyi20191119-012345"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。

InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.InternalRPCError	调用超时。
ResourceNotFound	资源不存在。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.AppNotExists	应用未创建或是已删除。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.DeviceNotGateway	不是网关设备。
UnsupportedOperation.NotBindVirtualDevice	无法绑定虚拟设备。
UnsupportedOperation.DeviceNotBind	子设备未绑定到网关设备。
UnsupportedOperation.CannotBindSameFamily	该设备已经绑定请勿重复绑定。
LimitExceeded	超出限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。

# 获取歌曲播放链接

最近更新时间：2025-12-29 15:26:14

## 1. 接口描述

接口请求域名：iot.cloud.tencent.com/api/exploreropen/tokenapi

本接口（song\_url）用于获取歌曲播放链接。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用uuld。定位问题时，需提供该次请求的RequestId。
Action	String	是	公共参数，本接口取值：AppKugouUserCommand。
KGCommand	String	是	子命令，本接口取值：song_url。
DeviceId	String	是	设备 Id 由产品 ID 和设备 Name 组成。
KugouParams	Object of <a href="#">KugouParams_song_url</a>	是	子命令的参数。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的RequestId。
ErrorCode	Integer	请求返回的错误码，0为没有错误。
ErrorMsg	String	请求返回的错误信息。

Data	Object of <a href="#">Data_song_url</a>	返回结果
------	--	------

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
  "RequestId": "test0191119-012345",
  "Action": "AppKugouUserCommand",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "DeviceId": "LAEG4YJE1A/dev01",
  "KGCommand": "song_url",
  "KugouParams": { ... }
}
```

#### 输出示例：成功

```
{
  "Response": {
    "Data": { ... },
    "ErrorCode": 0, "ErrorMsg": "",
    "RequestId": "keyi20191119-012345"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

InvalidParameterValue.InternalRPCError	调用超时。
ResourceNotFound	资源不存在。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.AppNotExists	应用未创建或是已删除。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.DeviceNotGateway	不是网关设备。
UnsupportedOperation.NotBindVirtualDevice	无法绑定虚拟设备。
UnsupportedOperation.DeviceNotBind	子设备未绑定到网关设备。
UnsupportedOperation.CannotBindSameFamily	该设备已经绑定请勿重复绑定。
LimitExceeded	超出限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。

# 每日推荐

最近更新时间：2025-12-29 15:26:14

## 1. 接口描述

接口请求域名： [iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)

本接口（awesome\_everyday）用于获取每日推荐歌单。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppKugouUserCommand。
KGCommand	String	是	子命令，本接口取值：awesome_everyday。
DeviceId	String	是	设备Id 由产品 ID 和设备 Name 组成。
KugouParams	Object of <a href="#">KugouParams_awesome_everyday</a>	是	子命令的参数。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
ErrorCode	Integer	请求返回的错误码，0为没有错误。
ErrorMsg	String	请求返回的错误信息。

Data	Object of <a href="#">Data_awesome_everyday</a>	返回结果。
------	--	-------

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json

{
  "RequestId": "test0191119-012345",
  "Action": "AppKugouUserCommand",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "DeviceId": "LAEG4YJE1A/dev01",
  "KGCommand": "awesome_everyday",
  "KugouParams": { ... }
}
```

#### 输出示例: 成功

```
{
  "Response": {
    "Data": { ... },
    "ErrorCode": 0, "ErrorMsg": "",
    "RequestId": "keyi20191119-012345"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

InvalidParameterValue.InternalRPCError	调用超时。
ResourceNotFound	资源不存在。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.AppNotExists	应用未创建或是已删除。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.DeviceNotGateway	不是网关设备。
UnsupportedOperation.NotBindVirtualDevice	无法绑定虚拟设备。
UnsupportedOperation.DeviceNotBind	子设备未绑定到网关设备。
UnsupportedOperation.CannotBindSameFamily	该设备已经绑定请勿重复绑定。
LimitExceeded	超出限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。

# 批量查询歌曲信息

最近更新时间：2025-12-29 15:26:14

## 1. 接口描述

接口请求域名： [iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)

本接口（song\_infos）用于批量查询歌曲信息。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuid。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppKugouUserCommand。
KGCommand	String	是	子命令，本接口取值：song_infos。
DeviceId	String	是	设备Id 由产品 ID 和设备 Name 组成。
KugouParams	Object of <a href="#">KugouParams_song_infos</a>	是	子命令的参数。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
ErrorCode	Integer	请求返回的错误码，0为没有错误。
ErrorMsg	String	请求返回的错误信息。

Data	Object of <b>Data_song_infos</b>	返回结果。
------	-------------------------------------	-------

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "test0191119-012345",
  "Action": "AppKugouUserCommand",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "DeviceId": "LAEG4YJE1A/dev01",
  "KGCommand": "song_infos",
  "KugouParams": { ... }
}
```

#### 输出示例：成功

```
{
  "Response": {
    "Data": { ... },
    "ErrorCode": 0, "ErrorMsg": "",
    "RequestId": "keyi20191119-012345"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

InvalidParameterValue.InternalRPCError	调用超时。
ResourceNotFound	资源不存在。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.AppNotExists	应用未创建或是已删除。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.DeviceNotGateway	不是网关设备。
UnsupportedOperation.NotBindVirtualDevice	无法绑定虚拟设备。
UnsupportedOperation.DeviceNotBind	子设备未绑定到网关设备。
UnsupportedOperation.CannotBindSameFamily	该设备已经绑定请勿重复绑定。
LimitExceeded	超出限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。

# 推荐歌单

最近更新时间：2025-12-29 15:26:14

## 1. 接口描述

接口请求域名：iot.cloud.tencent.com/api/exploreropen/tokenapi

本接口（playlist\_awesome）用于获取推荐歌单。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppKugouUserCommand。
KGCommand	String	是	子命令，本接口取值：playlist_awesome。
DeviceId	String	是	设备 Id 由产品 ID 和设备 Name 组成。
KugouParams	Object of <a href="#">KugouParams_playlist_awesome</a>	是	子命令的参数。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
ErrorCode	Integer	请求返回的错误码，0 为没有错误。
ErrorMsg	String	请求返回的错误信息。

Data	Object of <a href="#">data_playlist_awesome</a>	返回结果。
------	--	-------

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "test0191119-012345",
  "Action": "AppKugouUserCommand",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "DeviceId": "LAEG4YJE1A/dev01",
  "KGCommand": "playlist_awesome",
  "KugouParams": { ... }
}
```

#### 输出示例：成功

```
{
  "Response": {
    "Data": { ... },
    "ErrorCode": 0, "ErrorMsg": "",
    "RequestId": "keyi20191119-012345"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

InvalidParameterValue.InternalRPCError	调用超时。
ResourceNotFound	资源不存在。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.AppNotExists	应用未创建或是已删除。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.DeviceNotGateway	不是网关设备。
UnsupportedOperation.NotBindVirtualDevice	无法绑定虚拟设备。
UnsupportedOperation.DeviceNotBind	子设备未绑定到网关设备。
UnsupportedOperation.CannotBindSameFamily	该设备已经绑定请勿重复绑定。
LimitExceeded	超出限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。

# 新歌首发

最近更新时间：2025-12-29 15:26:14

## 1. 接口描述

接口请求域名： [iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)

本接口(awesome\_newsong)用于获取首发新歌。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppKugouUserCommand。
KGCommand	String	是	子命令，本接口取值：awesome_newsong。
DeviceId	String	是	设备Id 由产品 ID 和设备 Name 组成。
KugouParams	Object of <a href="#">KugouParams_awesome_newsong</a>	是	子命令的参数。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
ErrorCode	Integer	请求返回的错误码，0为没有错误。
ErrorMsg	String	请求返回的错误信息。

Data	Object of <b>Data_awesome_newsong</b>	返回结果。
------	--	-------

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "test0191119-012345",
  "Action": "AppKugouUserCommand",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "DeviceId": "LAEG4YJE1A/dev01",
  "KGCommand": "awesome_newsong",
  "KugouParams": { ... }
}
```

#### 输出示例：成功

```
{
  "Response": {
    "Data": { ... },
    "ErrorCode": 0, "ErrorMsg": "",
    "RequestId": "keyi20191119-012345"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。

InvalidParameterValue.InternalRPCError	调用超时。
ResourceNotFound	资源不存在。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.AppNotExists	应用未创建或是已删除。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.DeviceNotGateway	不是网关设备。
UnsupportedOperation.NotBindVirtualDevice	无法绑定虚拟设备。
UnsupportedOperation.DeviceNotBind	子设备未绑定到网关设备。
UnsupportedOperation.CannotBindSameFamily	该设备已经绑定，请勿重复绑定。
LimitExceeded	超出限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。

# 用户登出

最近更新时间：2025-12-29 15:26:14

## 1. 接口描述

接口请求域名： [iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)  
本接口(user\_logout)用于用户登出。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppKugouUserCommand。
KGCommand	String	是	子命令，本接口取值：user_logout。
DeviceId	String	是	设备 Id 由产品 ID 和设备 Name 组成。
KugouParams	Object of <a href="#">KugouParams_user_logout</a>	是	子命令的参数。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
ErrorCode	Integer	请求返回的错误码，0为没有错误。
ErrorMsg	String	请求返回的错误信息。
Data	Object of <a href="#">data_user_logout</a>	返回结果。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "test0191119-012345",
  "Action": "AppKugouUserCommand",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "DeviceId": "LAEG4YJE1A/dev01",
  "KGCommand": "user_logout",
  "KugouParams": { ... }
}
```

#### 输出示例：成功

```
{
  "Response": {
    "Data": { ... },
    "ErrorCode": 0, "ErrorMsg": "",
    "RequestId": "keyi20191119-012345"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.InternalRPCError	调用超时。

ResourceNotFound	资源不存在。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.AppNotExists	应用未创建或是已删除。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.DeviceNotGateway	不是网关设备。
UnsupportedOperation.NotBindVirtualDevice	无法绑定虚拟设备。
UnsupportedOperation.DeviceNotBind	子设备未绑定到网关设备。
UnsupportedOperation.CannotBindSameFamily	该设备已经绑定，请勿重复绑定。
LimitExceeded	超出限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。

# 用户信息

最近更新时间：2025-12-29 15:26:14

## 1. 接口描述

接口请求域名： [iot.cloud.tencent.com/api/exploreropen/tokenapi](https://iot.cloud.tencent.com/api/exploreropen/tokenapi)  
本接口(user\_info)用于获取用户信息。

## 2. 输入参数

名称	类型	必选	描述
AccessToken	String	是	公共参数，AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
Action	String	是	公共参数，本接口取值：AppKugouUserCommand。
KGCommand	String	是	子命令，本接口取值：user_info。
DeviceId	String	是	设备 Id 由产品 ID 和设备 Name 组成。
KugouParams	Object of <a href="#">KugouParams_user_info</a>	是	子命令的参数。

## 3. 输出参数

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。
ErrorCode	Integer	请求返回的错误码，0为没有错误。
ErrorMsg	String	请求返回的错误信息。
Data	Object of <a href="#">Data_user_info</a>	返回结果。

## 4. 示例

### 示例1

#### 输入示例

```
POST https://iot.cloud.tencent.com/api/exploreropen/tokenapi HTTP/1.1
content-type: application/json
{
  "RequestId": "test0191119-012345",
  "Action": "AppKugouUserCommand",
  "AccessToken": "8b4a70dd16105f*****18edd4e78a3bb8ec",
  "DeviceId": "LAEG4YJE1A/dev01",
  "KGCommand": "user_info",
  "KugouParams": {...}
}
```

#### 输出示例：成功

```
{
  "Response": {
    "Data": {...},
    "ErrorCode": 0, "ErrorMsg": "",
    "RequestId": "keyi20191119-012345"
  }
}
```

## 5. 错误码

错误码	描述
InternalError	内部错误。
InvalidParameterValue	参数取值错误。
InvalidParameterValue.InvalidAccessToken	Token 无效。
InvalidParameterValue.InternalRPCError	调用超时。

ResourceNotFound	资源不存在。
ResourceNotFound.DeviceNotExist	设备未创建或是已删除。
ResourceNotFound.StudioProductNotExist	产品尚未创建或已被删除。
ResourceNotFound.AppNotExists	应用未创建或是已删除。
UnsupportedOperation	不支持的操作。
UnsupportedOperation.DeviceNotGateway	不是网关设备。
UnsupportedOperation.NotBindVirtualDevice	无法绑定虚拟设备。
UnsupportedOperation.DeviceNotBind	子设备未绑定到网关设备。
UnsupportedOperation.CannotBindSameFamily	该设备已经绑定请勿重复绑定。
LimitExceeded	超出限制。
LimitExceeded.FamilyDeviceExceedLimit	设备数量超出限制。
UnauthorizedOperation	无权操作。
UnauthorizedOperation.APPNoPermissionToStudioProduct	App 对操作该产品无权限。

# 数据结构

最近更新时间：2025-12-29 15:26:14

## Action

### 自动智能联动动作

被如下接口引用：`AppCreateAutomation`、`AppCreateScene`、`AppDescribeAutomation`、`AppGetSceneList`、`AppModifyAutomation`、`AppModifyScene`。

名称	类型	必选	描述
ActionType	Int	是	<p>动作类型。</p> <ul style="list-style-type: none"><li>0: 设备动作。</li><li>1: 延时。</li><li>2: 场景。</li><li>3: 通知。</li></ul>
ProductId	String	否	产品 ID。
DeviceName	String	否	设备名称。
Data	String	是	<ul style="list-style-type: none"><li>ActionType=0: 设备动作参数。</li><li>ActionType=1: 延时时间单位秒。</li><li>ActionType=2: 执行场景 Id。</li><li>ActionType=3: 通知内容。</li></ul>
AliasName	String	否	设备别名或者场景名称。
IconUrl	String	否	产品或者场景的图标 URL。

## ActionHistories

### 设备动作历史

被如下接口引用：`AppGetDeviceActionHistories`。

名称	类型	描述
DeviceName	String	设备名称。

	g	
ActionId	String	动作 Id。
ActionName	String	动作名称。
ReqTime	int64	请求时间。
RspTime	int64	响应时间。
InputParams	string	输入参数。
OutputParams	string	输出参数。
Calling	string	调用方式。
ClientToken	string	调用 Id。
Status	string	调用状态。

## ActionPushData

监听之后的服务器回包

被如下接口引用：DeviceRegistrationMonitor。

名称	类型	描述
deviceIds	[String]	返回成功注册监听的设备列表。 如果入参的设备 ID 无返回，表示该用户未绑定要注册监听的设备。

## ActionPushParam

注册设备的时候传入的参数

被如下接口引用：DeviceRegistrationMonitor。

名称	类型	必选	描述
AccessToken	String	是	AccessToken 为调用登录接口之后获取的 Token。

DeviceIds	[String]	是	需要注册监听的设备列表，必须是已经绑定到该用户名下的设备。
-----------	----------	---	-------------------------------

## AlarmDetailItem

### 预警详细信息

被如下接口引用：AppGetInsuranceAlarmDetailList。

名称	类型	必选	描述
Name	String	是	预警名称。
Reason	String	是	隐患分析。
Detail	String	是	预警细节。
Suggestion	String	是	处置建议。
AlarmTimeStamp	Int64	是	预警时间。

## AlarmSummaryItem

### 预警摘要信息

被如下接口引用：AppGetInsuranceAlarmSummaryList。

名称	类型	必选	描述
Name	String	是	预警名称。
Reason	String	是	隐患分析。
Suggestion	String	是	处置建议。
AlarmTimeStamp	Int64	是	预警时间。
AlarmCount	Int64	是	总预警数。

## AppDeviceInfo

### App 设备信息

被如下接口引用：AppBindSubDeviceInFamily、AppSecureAddDeviceInFamily、AppSigBindDeviceInFamily。

名称	类型	描述
DeviceId	String	设备 ID。
ProductId	String	产品 ID。
DeviceName	String	设备名。
AliasName	String	设备别名。
CreateTime	String	创建 unix 时间戳（秒级）。
UpdateTime	String	最后更新时间，unix 时间戳（秒级）。
FamilyId	String	家庭 ID。
RoomId	String	房间 ID。
IconUrl	String	图标 Url。

## AppInsuranceDeviceInfo

### 设备信息

被如下接口引用：AppGetUnBindDeviceList。

名称	类型	必选	描述
DeviceId	String	是	设备 ID。
DeviceNo	String	是	设备 SN 或车架号。

## AppServiceInsuranceInfo

### 保单信息

被如下接口引用：AppGetInsuranceList。

名称	类型	必选	描述
ItemId	String	是	保单类别。
ItemType	Int64	是	保单类型。 <ul style="list-style-type: none"><li>0: 测试保单。</li><li>1: 正式保单。</li></ul>
InsuranceNo	String	是	保单号。
Address	String	是	地址。
BeginTime	Int64	是	保单生效时间。
EndTime	Int64	是	保单结束时间。
Score	Int64	是	评分。
AlarmNum	Int64	是	预警条数。
Status	Int64	是	保单状态。 <ul style="list-style-type: none"><li>1: 失效。</li><li>2: 保障中。</li><li>3: 待生效。</li></ul>
ServiceName	String	是	服务名称。
BindDeviceId	String	是	绑定设备 ID。

## Automation

### 自动智能联动实体

被如下接口引用：AppDescribeAutomation。

名称	类型	必选	描述
AutomationId	String	是	自动智能联动 ID。
Icon	String	是	自动智能联动的背景图片地址。

Name	String	是	自动智能联动的名称。
Status	int	是	自动智能联动开关状态。 • 0: 关闭。 • 1: 启用。
MatchType	int	是	条件匹配类型。 • 0: 全部条件满足。 • 1: 其中一个条件满足。
EffectiveBeginTime	String	是	生效开始时间。
EffectiveEndTime	String	是	生效结束时间。
EffectiveDays	String	是	生效日期 由0和1组成的7位数字，0表示不执行，1表示执行，第1位为周日，依次表示周一至周六。
Conditions	Array of <a href="#">Condition</a>	是	自动智能联动触发条件。
Actions	Array of <a href="#">Action</a>	是	自动智能联动动作列表。

## AutomationListItem

### 自动智能联动信息

被如下接口引用：[AppGetAutomationList](#)。

名称	类型	必选	描述
AutomationId	String	是	自动智能联动 ID。
Icon	String	是	自动智能联动的背景图片地址。
Name	String	是	自动智能联动的名称。
Status	int	是	自动智能联动开关状态。 • 0: 关闭。 • 1: 启用。

## Condition

### App 设备信息

被如下接口引用：AppCreateAutomation、AppDescribeAutomation、AppModifyAutomation。

名称	类型	必选	描述
CondId	String	是	条件 ID。
CondType	Int32	是	条件类型。 <ul style="list-style-type: none"><li>0: 设备属性条件。</li><li>1: 定时条件。</li></ul>
Property	Property	否	条件属性。
Timer	Timer	否	条件触发时间。

## Data

### 返回数据

被如下接口引用：device\_activation。

名称	类型	描述
activation_date	string	设备激活日期。

## DataHistoryItem

### 历史数据记录

被如下接口引用：AppGetDeviceDataHistory。

名称	类型	描述
Time	String	毫秒时间戳。
Value	String	数据取值。

## Data\_accompany\_awesome\_personal

### 返回数据

被如下接口引用：accompany\_awesome\_personal。

名称	类型	必选	描述
accompany	Array Of Object of <a href="#">accompany_awesome_personal</a>	否	伴奏个性推荐。

## Data\_accompany\_awesome\_top

### 返回数据

被如下接口引用: [accompany\\_awesome\\_top](#)。

名称	类型	必选	描述
accompany	Array Of Object of <a href="#">accompany_awesome_top</a>	否	伴奏推荐列表。

## Data\_accompany\_category\_info

### 返回数据

被如下接口引用: [accompany\\_category\\_info](#)。

名称	类型	必选	描述
category_id	String	是	分类 id。
category_name	String	是	分类名。
category_im g	String	是	分类图片。

## Data\_accompany\_category\_list

### 返回数据

被如下接口引用: [accompany\\_category\\_list](#)。

名称	类型	必选	描述
groups	Array Of groups_accompany_category_list of <a href="#">groups_accompany_category_list</a>	否	伴奏分组。
recommen ds	Array Of recommends_accompany_category_list of <a href="#">recommends_accompany_category_list</a>	否	推荐分类。

## Data\_accompany\_category\_song

**返回数据**

被如下接口引用: [accompany\\_category\\_song](#)。

名称	类型	必选	描述
accompany	Array Of Object of <a href="#">accompany_category_song</a>	否	伴奏列表。

## Data\_accompany\_favorite\_list

被如下接口引用: [accompany\\_favorite\\_list](#)。

名称	类型	必选	描述
playlist_id	String	否	歌单 id。
total	Integer	否	总数。
accompany	Array Of Object of <a href="#">accompany_favorite_list</a>	否	伴奏列表。

## Data\_accompany\_free\_info

**返回数据**

被如下接口引用: [accompany\\_free\\_info](#)。

名称	类型	必选	描述
accompany	Array Of accompany_accompany_free_info of <a href="#">accompany_accompany_free_info</a>	是	体验信息。

## Data\_accompany\_free\_song\_url

**返回数据**

被如下接口引用: [accompany\\_free\\_song\\_url](#)。

名称	类型	必选	描述
accompany_size	String	是	伴奏大小单位字节。

has_origin al	Inte ger	否	音频是否有原唱音轨。 • 0:无。 • 1:有。
singer_na me	Stri ng	是	歌手名。
playable_ code	Inte ger	是	是否可以播放 枚举值依次对应： • 0: 可以。 • 1: 海外地区不能播放。 • 2: 歌曲无版权不能播放。 • 3: 会员歌曲, 非会员不能播放。 • 4: 付费内容, 须购买才可播放。 • 5: 音乐方案策略, 非会员不能播放。 • 6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情 况) • 9: 未知原因, 无权播放。
accompan y_id	Stri ng	是	伴奏 ID。
accompan y_url	Stri ng	是	伴奏下载链接。
duration	Inte ger	是	伴奏时长单位毫秒。
singer_id	Stri ng	是	歌手 ID。
song_nam e	Stri ng	是	歌曲名。

## Data\_accompany\_free\_top\_list

### 返回数据

被如下接口引用: accompany\_free\_top\_list。

名称	类型	必选	描述
groups	Array Of groups_accompany_free_top_list of <a href="#">groups_accompany_free_top_list</a>	否	分组列表。

## Data\_accompany\_free\_top\_song

### 返回数据

被如下接口引用: accompany\_free\_top\_song。

名称	类型	必选	描述
accompan y	Array Of Object of <a href="#">accompany_free_top_song</a>	否	伴奏列表。

## Data\_accompany\_free\_url

### 返回数据

被如下接口引用: accompany\_free\_url.

名称	类型	必选	描述
playable_code	Inte ger	是	是否可以播放 枚举值依次对应： <ul style="list-style-type: none"><li>0: 可以。</li><li>1: 海外地区不能播放。</li><li>2: 歌曲无版权不能播放。</li><li>3: 会员歌曲, 非会员不能播放。</li><li>4: 付费内容, 须购买才可播放。</li><li>5: 牛方案策略, 非会员不能播放。</li><li>6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。</li><li>9: 未知原因, 无权播放。</li></ul>
accompan y_id	Strin g	是	伴奏 ID。
accompan y_size	Strin g	是	伴奏大小单位字节。
accompan y_url	Strin g	是	伴奏下载链接。
has_origin al	Inte ger	否	音频是否有原唱音轨。 <ul style="list-style-type: none"><li>0: 无。</li><li>1: 有。</li></ul>
duration	Inte ger	是	伴奏时长单位毫秒。

singer_name	String	是	歌手名。
song_name	String	是	歌曲名。

## Data\_accompany\_hq\_url

### 返回数据

被如下接口引用：accompany\_hq\_url。

名称	类型	必选	描述
has_origin al	Integer	否	音频是否有原唱音轨。 <ul style="list-style-type: none"><li>0: 无。</li><li>1: 有。</li></ul>
duration	Integer	是	伴奏时长单位毫秒。
song_name	String	是	歌曲名。
playable_code	Integer	是	是否可以播放 枚举值依次对应： <ul style="list-style-type: none"><li>0: 可以。</li><li>1: 海外地区不能播放。</li><li>2: 歌曲无版权不能播放。</li><li>3: 会员歌曲，非会员不能播放。</li><li>4: 付费内容，须购买才可播放。</li><li>5: 牛方案策略，非会员不能播放。</li><li>6: 固定向版权下架不能播放（针对 APP 有权但设备端无权的情况）。</li><li>9: 未知原因，无权播放。</li></ul>
accompan y_id	String	是	伴奏 ID。
accompan y_size	String	是	伴奏大小单位字节。
singer_na me	String	是	歌手名。
accompan	String	是	伴奏下载链接。

y_url	g		
singer_id	String	是	歌手 ID。

## Data\_accompany\_info

### 返回数据

被如下接口引用：accompany\_info。

名称	类型	必选	描述
accompany	Array Of Object of <a href="#">accompany_accompany_info</a>	否	伴奏列表。

## Data\_accompany\_lyric

### 返回数据

被如下接口引用：accompany\_lyric。

名称	类型	必选	描述
lyric	String	是	base64编码的歌词内容。
offset	Integer	是	歌词偏移量。
format	String	是	歌词格式。

## Data\_accompany\_mv

### 返回数据

被如下接口引用：accompany\_mv。

名称	类型	必选	描述
mv_id	String	是	MV id。
mv_name	String	是	MV 名。
duration	Integer	否	MV 时长，单位毫秒。

offset	Integer	否	MV 与伴奏对齐时间偏移量。
--------	---------	---	----------------

## Data\_accompany\_newsearch\_song

### 返回数据

被如下接口引用: accompany\_newsearch\_song。

名称	类型	必选	描述
accompan y	Array Of Object of <a href="#">accompany_newsearch_song</a>	否	伴奏列表。
author	Array Of Object of <a href="#">author_newsearch_song</a>	否	歌手列表。

## Data\_accompany\_opus\_del

### 返回数据

被如下接口引用: accompany\_opus\_del。

名称	类型	描述
data	object	objects

## Data\_accompany\_opus\_list

### 返回数据

被如下接口引用: accompany\_opus\_list。

名称	类型	必选	描述
total	Integer	否	总数
opuses	Array Of opuses_accompany_opus_list of <a href="#">opuses_accompany_opus_list</a>	否	opuses

## Data\_accompany\_opus\_upload

### 返回数据

被如下接口引用: accompany\_opus\_upload。

名称	类型	必选	描述

opus_id	String	否	作品 id。
---------	--------	---	--------

## Data\_accompany\_opus\_url

### 返回数据

被如下接口引用：accompany\_opus\_url。

名称	类型	必选	描述
url	String	是	作品下载链接。
duration	Integer	是	时长单位秒。
size	Integer	是	伴奏大小单位字节。

## Data\_accompany\_search\_song

### 返回数据

被如下接口引用：accompany\_search\_song。

名称	类型	必选	描述
accompany	Array Of Object of <a href="#">accompany_search_song</a>	否	伴奏搜索。
total	Integer	否	总数。

## Data\_accompany\_search\_tips

被如下接口引用：accompany\_search\_tips。

名称	类型	必选	描述
tips	Array Of String	否	提示。

## Data\_accompany\_singer\_list

### 歌手列表

被如下接口引用：accompany\_singer\_list。

名称	类型	必选	描述

singers	Array Of <a href="#">singers_accompany_singer_list</a> of <a href="#">singers_accompany_singer_list</a>	否	热门歌手。
letters	Array Of <a href="#">letters_accompany_singer_list</a> of <a href="#">letters_accompany_singer_list</a>	否	letters

## Data\_accompany\_singer\_song

### 返回数据

被如下接口引用: accompany\_singer\_song。

名称	类型	必选	描述
total	Integer	否	总数。
accompany	Array Of Object of <a href="#">accompany_singer_song</a>	否	伴奏列表。

## Data\_accompany\_songpitch

### 返回数据

被如下接口引用: accompany\_songpitch。

名称	类型	必选	描述
pitch	String	是	音高。
offset	Integer	是	音高偏移量。
type	Integer	是	来源。 • 0为默认值。 • 1为算法制作。 • 2为人工导入。
md5sum	String	是	md5 签名, 用于完整性校验。

## Data\_accompany\_theme\_list

### 返回数据

被如下接口引用: accompany\_theme\_list。

名称	类型	必选	描述
themes	Array Of themes_accompany_theme_list of themes_accompany_theme_list	否	主题列表。

## Data\_accompany\_theme\_song

被如下接口引用: accompany\_theme\_song。

名称	类型	必选	描述
accompany	Array Of Object of accompany_theme_song	否	伴奏列表。

## Data\_accompany\_top\_list

被如下接口引用: accompany\_top\_list

名称	类型	必选	描述
groups	Array Of groups_accompany_top_list of groups_accompany_top_list	否	分组列表。

## Data\_accompany\_top\_song

返回数据

被如下接口引用: accompany\_top\_song。

名称	类型	必选	描述
accompany	Array Of Object of accompany_top_song	否	伴奏列表。

## Data\_accompany\_url

返回数据

被如下接口引用: accompany\_url。

名称	类型	必选	描述
accompan y_id	String	是	伴奏 ID

duration	Integer	是	伴奏时长单位毫秒
singer_id	String	是	歌手 ID
song_name	String	是	歌曲名
playable_code	Integer	是	是否可以播放 枚举值依次对应： • 0：可以。 • 1：海外地区不能播放。 • 2：歌曲无版权不能播放。 • 3：会员歌曲，非会员不能播放。 • 4：付费内容，须购买才可播放。 • 5：牛方案策略，非会员不能播放。 • 6：因定向版权下架不能播放（针对 APP 有权但设备端无权的情况）。 • 9：未知原因，无权播放。
status	Integer	是	• 0：成功获取到伴奏下载信息。 • 1：当日免费体验次数已经用完。
accompany_size	String	是	伴奏大小单位字节。
accompany_url	String	是	伴奏下载链接。
has_origin al	Integer	否	音频是否有原唱音轨。 • 0：无。 • 1：有。
singer_name	String	是	歌手名。

## Data\_album\_info

### 返回数据

被如下接口引用：album\_info。

名称	类型	描述
album_id	Integer	专辑 id。

album_name	String	专辑名。
album_translate_name	String	专辑译名。
album_img	String	专辑封面。
intro	String	专辑简介。
company	String	唱片公司。
singer_id	String	歌手 id。
singer_name	String	歌手名。
total	Integer	歌曲总数。
publish_time	String	发布时间。
songs	Array of <a href="#">songs_album_info</a>	歌曲列表。

## Data\_asset\_purchased\_albums

### 返回数据

被如下接口引用: `asset_purchased_albums`。

名称	类型	描述
total	Integer	总数。
albums	Array of <a href="#">albums_asset_purchased_albums</a>	专辑信息。

## Data\_asset\_purchased\_songs

### 返回数据

被如下接口引用: `asset_purchased_songs`。

名称	类型	描述
total	Integer	总数。

songs	Array of <a href="#">songs_asset_purchased_songs</a>	歌曲列表。
-------	---	-------

## Data\_awesome\_everyday

### 返回数据

被如下接口引用：awesome\_everyday。

名称	类型	描述
songs	array of <a href="#">songs_awesome_everyday</a>	歌曲列表。

## Data\_awesome\_newsong

### 返回数据

被如下接口引用：awesome\_newsong。

名称	类型	描述
total	Integer	总数。
songs	Array of <a href="#">songs_awesome_newsong</a>	歌曲列表。

## Data\_awesome\_recommend

### 返回数据

被如下接口引用：awesome\_recommend。

名称	类型	描述
songs	Array of <a href="#">songs_awesome_recommend</a>	歌曲列表。

## Data\_child\_home

### 返回数据

被如下接口引用：child\_home。

名称	类型	必选	描述

groups	Array Of groups_child_home of <a href="#">groups_child_home</a>	否	资源分组。
--------	--	---	-------

## Data\_child\_resource

### 返回数据

被如下接口引用: `child_resource`。

名称	类型	必选	描述
type	Integer	否	大分类。 <ul style="list-style-type: none"><li>3-歌单。</li><li>4-专辑。</li><li>7-小分类。</li></ul>
total	Integer	否	总数。
lists	Array Of lists_child_resource of <a href="#">lists_child_resource</a>	否	资源分组。

## Data\_favorite\_oper\_playlist

### 返回数据

被如下接口引用: `favorite_oper_playlist`。

名称	类型	描述
playlist_ex tra_id	String	歌单附加 id, 移除收藏时使用。
playlist_id	String	操作的歌单 id。

## Data\_favorite\_oper\_song

### 返回数据

被如下接口引用: `favorite_oper_song`。

名称	类型	描述
song_extr a_id	String	歌曲附加 id, 移除收藏时使用。
song_id	String	操作的歌曲 id。

## Data\_favorite\_other\_list

### 返回数据

被如下接口引用：favorite\_other\_list。

名称	类型	描述
playlists	Array of <a href="#">playlists_favorite_other_list</a>	歌单。

## Data\_favorite\_self\_list

### 返回数据

被如下接口引用：favorite\_self\_list。

名称	类型	描述
playlists	Array of <a href="#">playlists_favorite_self_list</a>	歌单。

## Data\_favorite\_song

### 返回数据

被如下接口引用：favorite\_song。

名称	类型	描述
total	Integer	总数。
songs	Array of <a href="#">songs_favorite_song</a>	歌曲列表。

## Data\_give\_common\_record

### 返回数据

被如下接口引用：give\_common\_record。

名称	类型	描述
status	Integer	-
times	Integer	剩余领取次数。

collection_time	String	上一次领取时间。
-----------------	--------	----------

## Data\_give\_common\_scancodeLogin

### 返回数据

被如下接口引用: `give_common_scancodeLogin`。

名称	类型	描述
status	Integer	-
order_no	String	订单号(同 sp 内唯一)。
partner_no	String	厂商订单号(同 sp 内唯一)。
create_time	String	订单创建时间。
vip_end_time	String	vip 有效期终止时间。
collection_time	String	上一次领取时间。

## Data\_live\_program\_list

### 返回数据

被如下接口引用: `live_program_list`。

名称	类型	必选	描述
programs	Array Of <code>programs_live_program_list</code> of <code>programs_live_program_list</code>	否	节目分组。

## Data\_live\_room\_status

### 返回数据

被如下接口引用: `live_room_status`。

名称	类型	必选	描述
status	Integer	是	房间内节目状态。 • 0: 停播。

			<ul style="list-style-type: none"> <li>• 1: 开播。</li> </ul>
horizontal	Integer	是	<p>横屏流状态。</p> <ul style="list-style-type: none"> <li>• 0: 停播。</li> <li>• 1: 开播。</li> </ul>
vertical	Integer	是	<p>竖屏流状态。</p> <ul style="list-style-type: none"> <li>• 0: 停播。</li> <li>• 1: 开播。</li> </ul>

## Data\_live\_room\_url

### 返回数据

被如下接口引用：live\_room\_url。

名称	类型	必选	描述
vertical	Array Of vertical_live_room_url of <a href="#">vertical_live_room_url</a>	否	竖屏流信息。
status	String	是	<p>房间内节目状态。</p> <ul style="list-style-type: none"> <li>• 0: 停播。</li> <li>• 1: 开播。</li> </ul>
room_id	Integer	否	房间 id。
expire	Integer	否	链接过期时间。
horizontal	Array Of horizontal_live_room_url of <a href="#">horizontal_live_room_url</a>	否	横屏流信息。

## Data\_longaudio\_free\_home

### 返回数据

被如下接口引用：longaudio\_free\_home。

名称	类型	必选	描述
groups	Array Of groups_longaudio_free_home of <a href="#">groups_longaudio_free_home</a>	否	资源分组。

## Data\_longaudio\_resource

### 返回数据

被如下接口引用：longaudio\_resource。

名称	类型	必选	描述
is_end	String	否	是否存在下一页。 <b>特别提醒：</b> 返回结果个数可能少于请求的size，建议以瀑布流的方式设计UI交互。
lists	Array Of lists_longaudio_resource of <a href="#">lists_longaudio_resource</a>	否	资源列表。

## Data\_minipauth

### 返回数据

被如下接口引用：wechat\_minipauth。

名称	类型	描述
token	string	用户令牌。
expire	integer	令牌有效期 unix 时间戳，单位秒。
userid	string	用户 id。

## Data\_monitor\_delay

返回值为空。

被如下接口引用：monitor\_delay。

## Data\_monitor\_listen

返回值为空。

被如下接口引用：monitor\_listen。

## Data\_mv\_category\_list

### 返回数据

被如下接口引用：mv\_category\_list。

名称	类型	必选	描述
categories	Array Of categories_mv_category_list of <a href="#">categories_mv_category_list</a>	否	分类列表。

## Data\_mv\_category\_mv

## 返回数据

被如下接口引用: [mv\\_category\\_mv](#)

名称	类型	必选	描述
mvs	Array Of mvs_mv_category_mv of <a href="#">mvs_mv_category_mv</a>	否	MV 列表。

## Data\_mv\_info

### 返回数据

被如下接口引用: [mv\\_info](#)。

名称	类型	必选	描述
song_id	String	是	歌曲 ID。
singer_id	String	是	歌手 ID。
singer_name	String	是	歌手名称。
author_name	String	否	MV 制作人名称.
intro	String	否	简介。
tags	<a href="#">tags_mv_info</a>	否	MV 信息。
mv_id	String	是	MV ID。
mv_img	String	是	MV 图片。
mv_name	String	是	MV 名称。
accompany_id	String	否	伴奏 ID。

## Data\_mv\_infos

### 返回数据

被如下接口引用: [mv\\_infos](#)。

名称	类型	必选	描述
mvInfos	Array Of mvInfos_mv_infos of <a href="#">mv_infos</a>	否	MV 信息。

## mvInfos\_mv\_infos

## Data\_mv\_live\_tme

### 返回数据

被如下接口引用: mv\_live\_tme。

名称	类型	必选	描述
mvs	Array Of mvs_mv_live_tme of <a href="#">mvs_mv_live_tme</a>	否	MV 列表。

## Data\_mv\_search

### 返回数据

被如下接口引用: mv\_search。

名称	类型	必选	描述
mvs	Array Of mvs_mv_search of <a href="#">mvs_mv_search</a>	否	MV 列表。
total	Integer	否	总数。

## Data\_mv\_trial

### 返回数据

被如下接口引用: mv\_trial。

名称	类型	必选	描述
mv_size_id	Integer	否	流畅版 MV 大小。
mv_url_id	String	否	流畅版 MV 下载链接。
mv_size_sd	Integer	否	标清版 MV 大小。
mv_url_sd	String	否	标清版 MV 下载链接。
mv_size_qhd	Integer	否	高清版 MV 大小。
mv_url_qhd	String	否	高清版 MV 下载链接。
mv_size_hd	Integer	否	超清版 MV 大小。

mv_url_hd	String	否	超清版 MV 下载链接。
mv_size_hd	Integer	否	蓝光版 MV 大小。
mv_url_fhd	String	否	蓝光版MV 下载链接。
duration	Integer	是	MV 时长单位毫秒。
mv_id	String	是	MV ID。
mv_name	String	是	MV 名。
singer_name	String	是	歌手名。

## Data\_mv\_url

### 返回数据

被如下接口引用: mv\_url

名称	类型	必选	描述
mv_size_hd	Integer	否	超清版 MV 大小。
mv_url_hd	String	否	超清版 MV 下载链接。
mv_size_fhd	Integer	否	蓝光版 MV 大小。
mv_url_fhd	String	否	蓝光版MV 下载链接。
mv_size_qhd	Integer	否	高清版 MV 大小。
mv_url_qhd	String	否	高清版 MV 下载链接。
mv_size_sd	Integer	否	标清版 MV 大小。
mv_url_sd	String	否	标清版 MV 下载链接。
mv_size_ld	Integer	否	流畅版 MV 大小。
mv_url_ld	String	否	流畅版 MV 下载链接。
duration	Integer	是	MV 时长单位毫秒。
mv_id	String	是	MV ID。

mv_name	String	是	MV 名。
singer_name	String	是	歌手名。

## Data\_playlist\_awesome

### 返回数据

被如下接口引用：playlist\_awesome

名称	类型	描述
playlists	Array of <a href="#">playlists_playlist_awesome</a>	歌单。

## Data\_playlist\_song

### 返回数据

被如下接口引用：playlist\_song。

名称	类型	描述
total	Integer	总数。
songs	Array of <a href="#">songs_playlist_song</a>	歌曲列表。

## Data\_playlist\_top

### 返回数据

被如下接口引用：playlist\_top

名称	类型	描述
groups	Array of <a href="#">groups_playlist_top</a>	分组列表。

## Data\_radio\_list

### 返回数据

被如下接口引用：radio\_list。

名称	类型	描述

groups	Array of <a href="#">groups_radio_list</a>	分组列表。
--------	--	-------

## Data\_radio\_song

返回数据

被如下接口引用：radio\_song。

名称	类型	描述
total	Integer	总数。
songs	Array of <a href="#">songs_radio_song</a>	歌曲列表。

## Data\_rigion\_home

返回数据

被如下接口引用：rigion\_home。

名称	类型	必选	描述
groups	Array Of groups_rigion_home of <a href="#">groups_rigion_home</a>	否	资源分组。

## Data\_rigion\_oldman\_index

返回数据

被如下接口引用：rigion\_oldman\_index。

名称	类型	必选	描述
groups	Array Of groups_rigion_oldman_index of <a href="#">groups_rigion_oldman_index</a>	否	资源分组。

## Data\_search\_song

返回数据

被如下接口引用：search\_song。

名称	类型	必选	描述
total	Integer	是	总数。
keyword	String	是	关键词。

songs	Array Of songs_search_song of <a href="#">songs_search_song</a>	是	歌曲列表。
-------	--	---	-------

## Data\_search\_tips

返回数据

被如下接口引用: `search_tips`。

名称	类型	必选	描述
tips	Array Of String	是	tips

## Data\_search\_voice

返回数据

被如下接口引用: `search_voice`。

名称	类型	必选	描述
tts_text	String	是	用于语音提示的文本。
songs	<a href="#">songs_search_voice</a>	是	语音搜索歌曲列表。

## Data\_singer\_album

返回数据

被如下接口引用: `singer_album`。

名称	类型	描述
total	Integer	歌曲总数。
albums	Array of <a href="#">albums_singer_album</a>	专辑列表。

## Data\_singer\_info

返回数据

被如下接口引用: `singer_info`。

名称	类型	描述
singer_id	String	歌手 id。

singer_name	String	歌手名。
area	String	歌手地区。
translate_name	String	歌手译名。
song_count	Integer	歌手歌曲数。
singer_img	String	歌手相片。

## Data\_singer\_list

返回数据

被如下接口引用：singer\_list。

名称	类型	描述
singers	Array of <a href="#">singers_singer_list</a>	歌手

## Data\_singer\_song

返回数据

被如下接口引用：singer\_song。

名称	类型	描述
total	Integer	歌曲总数。
songs	Array of <a href="#">songs_singer_song</a>	歌曲列表。

## Data\_song\_accompany

返回数据

被如下接口引用：song\_accompany。

名称	类型	描述
accompany	Array of <a href="#">accompany_song_accompany</a>	伴奏。

## Data\_song\_infos

返回数据

被如下接口引用: [song\\_infos](#)。

名称	类型	描述
songs	Array of <a href="#">songs_song_infos</a>	歌曲列表。

## Data\_song\_lyric

返回数据

被如下接口引用: [song\\_lyric](#)。

名称	类型	描述
lyric	string	歌词。

## Data\_song\_tolisten

返回数据

被如下接口引用: [song\\_tolisten](#)。

名称	类型	描述
songs	Array of <a href="#">songs_song_tolisten</a>	歌曲列表。

## Data\_song\_url

返回数据

被如下接口引用: [song\\_url](#)。

名称	类型	描述
song_id	String	歌曲 id。
song_name	String	歌曲名。
singer_name	String	歌手名。
song_url	String	歌曲下载链接。

song_size	String	歌曲大小。
song_url_hq	String	歌曲(hq)下载链接。
song_size_hq	String	歌曲(hq)大小。
song_url_sq	String	歌曲(sq)下载链接。
song_size_sq	String	歌曲(sq)大小。
duration	Integer	歌曲时长, 单位毫秒。
playable_code	Integer	是否可以播放 枚举值依次对应: <ul style="list-style-type: none"> <li>• 0: 可以。</li> <li>• 1: 海外地区不能播放。</li> <li>• 2: 歌曲无版权不能播放。</li> <li>• 3: 会员歌曲, 非会员不能播放。</li> <li>• 4: 付费内容, 须购买才可播放。</li> <li>• 5: 牛方案策略, 非会员不能播放。</li> <li>• 6: 因定向版权下架不能播放 (针对 APP有权但设备端无权的情况)。</li> <li>• 9: 未知原因, 无权播放。</li> </ul>
try_url	String	歌曲试听链接。
try_size	Integer	歌曲试听大小, 单位字节。
try_begin	Integer	歌曲试听开始, 单位毫秒。
try_end	Integer	歌曲试听结束, 单位毫秒。

## Data\_top\_list

返回数据

被如下接口引用: `top_list`。

名称	类型	描述
groups	Array of <code>groups_top_list</code>	分组。

## Data\_top\_song

返回数据

被如下接口引用：top\_song。

名称	类型	描述
total	Integer	总数。
songs	songs_top_song	歌曲列表。

## Data\_user\_info

返回数据

被如下接口引用：user\_info。

名称	类型	描述
userid	String	用户 id。
nick_name	String	用户昵称。
img	String	用户头像。
is_vip	Integer	是否vip。 • 0: 否。 • 1: 是。
vip_end_time	String	vip 有效期终止时间。
car_vip_end_time	String	车机会员有效期终止时间。
svip_end_time	String	豪 V 有效期终止时间。

## Data\_user\_logout

返回数据为空。

被如下接口引用：user\_logout。

## Data\_user\_qrcode\_auth

## 返回数据

被如下接口引用: `user_qrcode_auth`。

名称	类型	描述
userid	String	用户 ID。
token	String	用户令牌。
expire	Integer	令牌有效期 unix 时间戳, 单位秒。

## **Data\_user\_qrcode\_get**

### 返回数据

被如下接口引用: `user_qrcode_get`。

名称	类型	描述
qrcode	String	登录二维码 url, 自行渲染。
ticket	String	获取二维码授权信息的凭证。

## **Data\_vip\_activity\_partner\_give**

### 返回数据

被如下接口引用: `vip_activity_partner_give`

名称	类型	必选	描述
create_time	String	否	TME 订单创建时间。
vip_end_time	String	否	vip 有效期终止时间。
status	Integer	是	-
order_no	String	否	TME 订单号(同 sp 内唯一)。
partner_no	String	否	厂商订单号(同 sp 内唯一)。

## **Data\_vip\_ksing\_info**

### 返回数据

被如下接口引用: `vip_ksing_info`。

名称	类型	必选	描述

userid	String	是	用户 ID。
nick_name	String	是	用户昵称。
img	String	是	用户头像。
is_vip	Integer	是	是否vip。 • 0: 否。 • 1: 是。
vip_end_time	String	是	vip 有效期终止时间。

## Data\_vip\_ksing\_partner\_give

返回数据

被如下接口引用: vip\_ksing\_partner\_give

名称	类型	必选	描述
order_no	String	是	TME 订单号(同 sp 内唯一)。
partner_no	String	是	厂商订单号(同 sp 内唯一)。
create_time	String	是	TME 订单创建时间。
vip_end_time	String	是	vip 有效期终止时间。

## Data\_vip\_svip\_common\_give

返回数据

被如下接口引用: vip\_svip\_common\_give

名称	类型	必选	描述
order_no	String	是	TME 订单号(同 sp 内唯一)。
partner_no	String	是	厂商订单号(同 sp 内唯一)。
create_time	String	是	TME 订单创建时间。

## Data\_vip\_tme\_partner\_give

## 返回数据

被如下接口引用: `vip_tme_partner_give`

名称	类型	必选	描述
order_no	String	是	TME 订单号(同 sp 内唯一)。
partner_no	String	是	厂商订单号(同 sp 内唯一)。
create_time	String	是	TME 订单创建时间。
vip_end_time	String	是	vip 有效期终止时间。

## Data\_wechat\_appinfocode

### 返回数据

被如下接口引用: `wechat_appinfocode`

名称	类型	描述
ticket	String	获取第三方 App 信息凭证。
req_name	String	拉起的小程序的请求名。
req_path	String	拉起的小程序的请求路径。

## Data\_wechat\_openlink

### 返回数据

被如下接口引用: `wechat_openlink`

名称	类型	描述
ticket	String	获取扫码授权结果的凭证。
openlink	String	微信小程序 openlink。
expire	Integer	令牌有效期 unix 时间戳, 单位秒。

## Data\_wechat\_qrcodeauth

### 返回数据

被如下接口引用: `wechat_qrcodeauth`

名称	类型	描述
token	String	用户令牌。
userid	String	用户 id。
expire	Integer	令牌有效期 unix 时间戳，单位秒。

## Data\_wechat\_qrcodeget

返回数据

被如下接口引用：wechat\_qrcodeget

名称	类型	描述
ticket	String	获取扫码授权结果的凭证。
qrcode	String	微信小程序二维码图像二进制数据 base64 编码。

## Device

设备信息

被如下接口引用：AppGetDeviceStatuses

名称	类型	描述
DeviceName	String	设备名。
Online	Int	在线状态。
LoginTime	Int64	登录时间。
LastDataTime	Int64	最近登录时间。

## DeviceFenceEvent

设备地理围栏告警事件

被如下接口引用：AppGetFenceEventList

名称	类型	必选	描述
EventType	String	是	设备地理围栏名称。

CreateTime	String	否	创建 unix 时间戳（秒级）。
------------	--------	---	------------------

## DeviceItem

### 设备状态

被如下接口引用：AppGetInsuranceDeviceStatus

名称	类型	必选	描述
ProductName	String	是	设备展示名称。
OnlineStatus	Int	是	设备状态： <ul style="list-style-type: none"><li>0：离线。</li><li>1：在线。</li><li>2：未知。</li></ul>

## DeviceList

### 设备列表

被如下接口引用：AppGetFamilyDeviceList、AppGetFamilySubDeviceList、AppGetGatewayBindDeviceList

名称	类型	描述
ProductId	String	设备 ID。
DeviceName	String	设备名称。
DeviceId	String	设备 Id。
AliasName	String	设备别名。
UserId	String	用户 Id。
RoomId	String	房间 ID。
FamilyId	String	家庭 ID。
IconUrl	String	图标 Url。

DeviceType	Int32	设备类型： • 0: 普通。 • 1: 网关设备。 • 2: 子设备。
CreateTime	Int64	创建 unix 时间戳 (秒级)。
UpdateTime	Int64	最后更新的 unix 时间戳 (秒级)。

## ErrorMessage

错误信息

名称	类型	描述
Code	String	错误码。
Message	String	错误信息。

## EventHistory

搜集结果集

名称	类型	描述
TimeStamp	Int64	事件的时间戳。
ProductId	String	事件的产品 ID。
DeviceName	String	事件的设备名称。
EventId	String	事件的标识符 ID。
Type	String	事件的类型。
Data	string	事件的数据。

## FamilyList

被如下接口引用：AppGetFamilyList

名称	类型	描述

FamilyId	String	家庭 ID。
Name	String	名称。
Role	Int	角色。
CreateTime	Int	创建时间。
UpdateTime	Int	最后更新时间。

## Fence

设备地理围栏信息

被如下接口引用：AppGetFenceList

名称	类型	必选	描述
FenceId	Int64	是	设备地理围栏 ID。
FenceName	String	是	设备地理围栏名称。
FenceArea	String	是	设备地理围栏区域范围。
FenceDesc	String	是	设备地理围栏描述。
AlertCondition	String	是	触发条件。
FenceEnable	Boolean	是	启用状态。
Method	String	是	通知方式。
CreateTime	String	否	创建 unix 时间戳（秒级）。
UpdateTime	String	否	最后更新时间， unix 时间戳（秒级）。

## HeartBeatData

心跳接口的服务器回包

被如下接口引用：HeartBeat

名称	类型	描述
RequestId	String	公共参数，唯一请求 ID，与入参相同，定位问题时，需提供该次请求的 RequestId。

## HeartBeatParam

设备心跳接口入参

被如下接口引用：HeartBeat

名称	类型	描述
DeviceIds	[String]	设备列表。

## IoUser

用户信息

被如下接口引用：AppFindUser

名称	类型	描述
UserId	String	用户 Id。
NickName	String	昵称。
Avatar	String	头像。
CountryCode	String	手机号国家码。
PhoneNumber	String	手机号码。
Email	String	邮箱。

## IoUserDetail

用户信息

被如下接口引用：App GetUser

名称	类型	描述

UserId	String	用户 Id。
NickName	String	昵称。
Avatar	String	头像。
CountryCode	String	手机号国家码。
PhoneNumber	String	手机号码。
Email	String	邮箱。
HasPassword	Int	是否已设置密码。 ● 0: 未设置。 ● 1: 已设置。
HasWxOpenID	Int	是否已绑定微信。 ● 0: 未绑定。 ● 1: 已绑定。

## KugouParams

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：device\_activation。

## KugouParams\_accompany\_awesome\_personal

请求参数

被如下接口引用：accompany\_awesome\_personal

名称	类型	必选	描述
size	Integer	是	每页?项。
page	Integer	是	第?页。

## KugouParams\_accompany\_awesome\_top

请求参数

被如下接口引用：accompany\_awesome\_top。

名称	类型	必选	描述

size	Integer	是	每页?项。
page	Integer	是	第?页。
type	Integer	是	排行榜类型。 <ul style="list-style-type: none"><li>1: 热门榜。</li><li>2: 飙升榜。</li></ul>

## KugouParams\_accompany\_category\_info

### 请求参数

被如下接口引用: accompany\_category\_info。

名称	类型	必选	描述
category_id	String	是	伴奏分类 id。

## KugouParams\_accompany\_category\_list

[object Object]

被如下接口引用: accompany\_category\_list。

## KugouParams\_accompany\_category\_song

### 请求参数

被如下接口引用: accompany\_category\_song。

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。
category_id	String	是	伴奏分类 id。

## KugouParams\_accompany\_favorite\_list

### 请求参数

被如下接口引用: accompany\_favorite\_list。

名称	类型	必选	描述
page	Integer	是	第?页。

size	Integer	是	每页?项。
------	---------	---	-------

## KugouParams\_accompany\_free\_info

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：accompany\_free\_info。

## KugouParams\_accompany\_free\_song\_url

请求参数

被如下接口引用：accompany\_free\_song\_url

名称	类型	必选	描述
accompany_id	String	是	伴奏 id。

## KugouParams\_accompany\_free\_top\_list

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：accompany\_free\_top\_list。

## KugouParams\_accompany\_free\_top\_song

请求参数

被如下接口引用：accompany\_free\_top\_song

名称	类型	必选	描述
group_id	String	是	分组 id。
top_id	String	是	榜单、电台 id。

## KugouParams\_accompany\_free\_url

请求参数

被如下接口引用：accompany\_free\_url。

名称	类型	必选	描述
accompany_id	String	是	伴奏 id。
free_token	String	否	免费令牌。

## KugouParams\_accompany\_hq\_url

### 请求参数

被如下接口引用：accompany\_hq\_url。

名称	类型	必选	描述
accompa ny_id	String	是	伴奏 id。

## KugouParams\_accompany\_info

### 请求参数

被如下接口引用：accompany\_info。

名称	类型	必选	描述
accompa ny_id	String	是	伴奏 id。

## KugouParams\_accompany\_lyric

### 请求参数

被如下接口引用：accompany\_lyric。

名称	类型	必选	描述
accompa ny_id	String	是	伴奏 id。

## KugouParams\_accompany\_mv

### 请求参数

被如下接口引用：accompany\_mv。

名称	类型	必选	描述
accompa ny_id	String	是	伴奏 id。

## KugouParams\_accompany\_newsearch\_song

### 请求参数

被如下接口引用：accompany\_newsearch\_song

名称	类型	必选	描述
keyword	String	是	关键词。

## KugouParams\_accompany\_opus\_del

请求参数

被如下接口引用: accompany\_opus\_del

名称	类型	必选	描述
opus_id	String	是	作品 id。

## KugouParams\_accompany\_opus\_list

请求参数

被如下接口引用: accompany\_opus\_list

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。

## KugouParams\_accompany\_opus\_upload

请求参数

被如下接口引用: accompany\_opus\_upload

名称	类型	必选	描述
opus_has h	String	是	作品 hash。
opus_des c	String	是	作品描述。
krc_id	String	是	歌词 id。
grade	String	是	等级。
opus_name	String	是	作品名。
accompan y_id	String	是	伴奏 id。

duration	Integer	是	作品播放时长，单位秒。
score	String	是	总分。
average_score	String	是	平均分。

## KugouParams\_accompany\_opus\_url

### 请求参数

被如下接口引用：accompany\_opus\_url

名称	类型	必选	描述
opus_id	String	是	作品 id。

## KugouParams\_accompany\_search\_song

### 请求参数

被如下接口引用：accompany\_search\_song

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。
keyword	String	是	关键词。

## KugouParams\_accompany\_search\_tips

### 请求参数

被如下接口引用：accompany\_search\_tips

名称	类型	必选	描述
keyword	String	是	关键词。

## KugouParams\_accompany\_singer\_list

### 请求参数

被如下接口引用：accompany\_singer\_list

名称	类型	必选	描述

type	Integer	是	<p>歌手类型。</p> <ul style="list-style-type: none"> <li>● 1: 全部。</li> <li>● 2: 男歌手。</li> <li>● 3: 女歌手。</li> <li>● 4: 组合。</li> </ul>
area	Integer	是	<p>歌手区域。</p> <ul style="list-style-type: none"> <li>● 1: 华语。</li> <li>● 2: 欧美。</li> <li>● 3: 日韩。</li> </ul>

## KugouParams\_accompany\_singer\_song

请求参数

被如下接口引用: accompany\_singer\_song

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。
singer_id	String	是	歌手 id。

## KugouParams\_accompany\_songpitch

请求参数

被如下接口引用: accompany\_songpitch

名称	类型	必选	描述
accompa ny_id	String	是	伴奏 id。

## KugouParams\_accompany\_theme\_list

请求参数

被如下接口引用: accompany\_theme\_list

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。

## KugouParams\_accompany\_theme\_song

### 请求参数

被如下接口引用: accompany\_theme\_song

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。
theme_id	String	是	主题 id。

## KugouParams\_accompany\_top\_list

请求参数这里为空, 传递空对象 {} 即可。

被如下接口引用: accompany\_top\_list。

## KugouParams\_accompany\_top\_song

### 请求参数

被如下接口引用: accompany\_top\_song

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。
group_id	String	是	分组 id。
top_id	String	否	榜单 id。

## KugouParams\_accompany\_url

### 请求参数

被如下接口引用: accompany\_url

名称	类型	必选	描述
accompany_id	String	是	伴奏 id。

## KugouParams\_album\_info

### 请求参数

被如下接口引用：album\_info

名称	类型	描述
page	Integer	第?页。
size	Integer	每页?项。
album_id	String	专辑 id。

## KugouParams\_asset\_purchased\_albums

### 请求参数

被如下接口引用：asset\_purchased\_albums

名称	类型	描述
page	Integer	第?页。
size	Integer	每页?项。

## KugouParams\_asset\_purchased\_songs

### 请求参数

被如下接口引用：asset\_purchased\_songs

名称	类型	描述
page	Integer	第?页。
size	Integer	每页?项。

## KugouParams\_awesome\_everyday

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：awesome\_everyday。

## KugouParams\_awesome\_newsong

### 请求参数

被如下接口引用：awesome\_newsong

名称	类型	描述

page	Integer	第?页。
size	Integer	每页?项。
top_id	Int	<p>榜单 id。</p> <ul style="list-style-type: none"> <li>• 1: 华语。</li> <li>• 2: 欧美。</li> <li>• 3: 韩语。</li> <li>• 4: 日语。</li> </ul>

## KugouParams\_awesone\_recommend

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：awesome\_recommend。

## KugouParams\_child\_home

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：child\_home。

## KugouParams\_child\_resource

请求参数

被如下接口引用：child\_resource

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。
resource_type	Integer	是	大分类 7-小分类
resource_id	String	是	资源 id。

## KugouParams\_favorite\_oper\_playlist

请求参数

被如下接口引用：favorite\_oper\_playlist

名称	类型	描述
cmd	Integer	<ul style="list-style-type: none"> <li>• 1: 收藏歌单。</li> </ul>

		<ul style="list-style-type: none"> <li>● 2: 取消收藏。</li> </ul>
playlist_id	String	歌单 id。
playlist_extra_id	String	<p>歌单附加 id。</p> <ul style="list-style-type: none"> <li>● 收藏时传入歌单 id。</li> <li>● 取消收藏时传入获取歌单列表(或收藏歌单)时返回的同名字段。</li> </ul>

## KugouParams\_favorite\_oper\_song

请求参数

被如下接口引用: favorite\_oper\_song

名称	类型	描述
cmd	Integer	<ul style="list-style-type: none"> <li>● 1: 收藏歌单。</li> <li>● 2: 取消收藏。</li> </ul>
playlist_id	String	歌单 id。
song_id	String	歌曲 id。
song_extra_id	String	<p>歌曲附加 id。</p> <ul style="list-style-type: none"> <li>● 收藏时传入歌曲 id。</li> <li>● 取消收藏时传入获取歌单歌曲列表时返回的同名字段。</li> </ul>

## KugouParams\_favorite\_other\_list

请求参数这里为空, 传递空对象 {} 即可。

被如下接口引用: favorite\_other\_list。

## KugouParams\_favorite\_self\_list

请求参数这里为空, 传递空对象 {} 即可。

被如下接口引用: favorite\_self\_list。

## KugouParams\_favorite\_song

请求参数

被如下接口引用: favorite\_song

名称	类型	描述

playlist_id	String	歌单 id。
page	Integer	第?页。
size	Integer	每页?项。
type	String	<ul style="list-style-type: none"> <li>默认: other。</li> <li>self: 自建歌单。</li> <li>other: 收藏的歌单。</li> </ul>

## KugouParams\_give\_common\_record

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：give\_common\_record。

## KugouParams\_give\_common\_scancodelogin

请求参数

被如下接口引用：give\_common\_scancodelogin

名称	类型	描述
partner_no	String	厂商订单号(同 sp 内唯一)。

## KugouParams\_live\_program\_list

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：live\_program\_list。

## KugouParams\_live\_room\_status

请求参数

被如下接口引用：live\_room\_status

名称	类型	必选	描述
room_id	String	是	房间 id。

## KugouParams\_live\_room\_url

请求参数

被如下接口引用：live\_room\_url

名称	类型	必选	描述
room_id	String	是	房间 id。

## KugouParams\_longaudio\_free\_home

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：longaudio\_free\_home。

## KugouParams\_longaudio\_resource

请求参数

被如下接口引用：longaudio\_resource

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。
tag_id	String	是	分类 tag_id。

## KugouParams\_monitor\_delay

请求参数

被如下接口引用：monitor\_delay

名称	类型	必选	描述
pid	String	是	牛方案 2.0 pid，产品接入时申请获得。
device_id	String	是	厂商设备唯一标示，由 ascii 可见字符组成。
client_ver	String	是	客户端版本号，厂商自行填写自家版本号，用于辅助监控统计。
platform	String	是	客户端平台版本信息，用于辅助监控统计。
timestamp	Integer	是	unix 时间戳，单位秒。
nonce	String	是	随机字串，由 ascii 可见字符组成。

data	Array Of data_monitor_delay of <a href="#">data_monitor_delay</a>	是	数据。
------	---	---	-----

## KugouParams\_monitor\_listen

### 请求参数

被如下接口引用: [monitor\\_listen](#)

名称	类型	必选	描述
pid	String	是	牛方案 2.0 pid, 产品接入时申请获得。
device_id	String	是	厂商设备唯一标示, 由 ascii 可见字符组成。
timestamp	Integer	是	unix 时间戳, 单位秒。
nonce	String	是	随机字串, 由 ascii 可见字符组成。
data	Array Of data_monitor_listen of <a href="#">data_monitor_listen</a>	是	数据。

## KugouParams\_mv\_category\_list

请求参数这里为空, 传递空对象 {} 即可。

被如下接口引用: [mv\\_category\\_list](#)。

## KugouParams\_mv\_category\_mv

### 请求参数

被如下接口引用: [mv\\_category\\_mv](#)

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。
category_id	String	是	分类 id。
sort	Integer	是	排序。

- 1: 发布时间。
- 2: 标签创建时间。
- 3: 播放量 (倒序)。
- 4: 算法。
- 5: mv 飙升值。

## KugouParams\_mv\_info

请求参数

被如下接口引用: mv\_info

名称	类型	必选	描述
mv_id	String	是	mv_id

## KugouParams\_mv\_infos

请求参数

被如下接口引用: mv\_infos

名称	类型	必选	描述
mvs_id	Array Of String	是	mvid 列表。

## KugouParams\_mv\_live\_tme

请求参数

被如下接口引用: mv\_live\_tme

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。
live_type	Integer	否	1-所有视频, 2-完整演唱会视频。

## KugouParams\_mv\_search

请求参数

被如下接口引用: mv\_search

名称	类型	必选	描述
userid	String	是	TME 用户 id。
page	Integer	是	第?页。
size	Integer	是	每页?项。
keyword	String	是	关键词。

## KugouParams\_mv\_trial

请求参数

被如下接口引用：mv\_trial

名称	类型	必选	描述
mv_id	String	是	mv_id

## KugouParams\_mv\_url

请求参数

被如下接口引用：mv\_url

名称	类型	必选	描述
mv_id	String	是	mv_id

## KugouParams\_playlist\_awesome

请求参数

被如下接口引用：playlist\_awesome

名称	类型	描述
page	Integer	第?页。
size	Integer	每页?项。

## KugouParams\_playlist\_song

请求参数

被如下接口引用：playlist\_song

名称	类型	描述

playlist_id	String	歌单分类 id。
page	Integer	第?页。
size	Integer	每页?项。

## KugouParams\_playlist\_top

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：playlist\_top。

## KugouParams\_radio\_list

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：radio\_list。

## KugouParams\_radio\_song

请求参数

被如下接口引用：radio\_song

名称	类型	描述
radio_id	String	电台 id。

## KugouParams\_rigion\_home

请求参数

被如下接口引用：rigion\_home

名称	类型	必选	描述
id	String	是	专区 id。

## KugouParams\_rigion\_oldman\_index

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：rigion\_oldman\_index。

## KugouParams\_search\_song

请求参数

被如下接口引用：search\_song

名称	类型	必选	描述
page	Integer	是	第?页。
size	Integer	是	每页?项。
keyword	String	是	关键词。

## KugouParams\_search\_tips

请求参数

被如下接口引用: `search_tips`

名称	类型	必选	描述
keyword	String	是	关键词。

## KugouParams\_search\_voice

请求参数

被如下接口引用: `search_voice`

名称	类型	必选	描述
query	String	是	用户语音在处理后的原始文本。
slots	Array Of slots_search_voice of <a href="#">slots_search_voice</a>	是	语音搜索列表。

## KugouParams\_singer\_album

请求参数

被如下接口引用: `singer_album`

名称	类型	描述
singer_id	String	歌手 id。
page	Integer	第?页。
size	Integer	每页?项。
sort	Integer	排序。 ● 1: 最热。

- 2: 最新。

## KugouParams\_singer\_info

请求参数

被如下接口引用: singer\_info

名称	类型	描述
singer_id	String	歌手 id。

## KugouParams\_singer\_list

请求参数

被如下接口引用: singer\_list

名称	类型	描述
page	Integer	第?页。
size	Integer	每页?项。
area	Integer	歌手地区。 <ul style="list-style-type: none"><li>• 0: 全部。</li><li>• 1: 内地。</li><li>• 2: 欧美。</li><li>• 3: 日本。</li><li>• 4: 韩国。</li><li>• 5: 港台。</li><li>• 6: 其它。</li></ul>
type	Integer	歌手类型。 <ul style="list-style-type: none"><li>• 0: 全部。</li><li>• 1: 男。</li><li>• 2: 女。</li><li>• 3: 组合。</li></ul>

## KugouParams\_singer\_song

请求参数

被如下接口引用: singer\_song

名称	类型	描述
singer_id	String	歌手 id。
page	Integer	第?页。
size	Integer	每页?项。

## KugouParams\_song\_accompany

请求参数

被如下接口引用: song\_accompany

名称	类型	描述
song_id	String	歌曲 id。

## KugouParams\_song\_infos

请求参数

被如下接口引用: song\_infos

名称	类型	描述
songs_id	Array of <a href="#">songs_id_song_infos</a>	歌曲 id 列表。

## KugouParams\_song\_lyric

请求参数

被如下接口引用: song\_lyric

名称	类型	描述
song_id	String	歌曲 id。

## KugouParams\_song\_tolisten

请求参数

被如下接口引用: song\_tolisten

名称	类型	描述
song_id	String	歌曲 id。

## KugouParams\_song\_url

请求参数

被如下接口引用: song\_url

名称	类型	描述
song_id	String	歌曲 id。

## KugouParams\_top\_list

请求参数这里为空, 传递空对象 {} 即可。

被如下接口引用: top\_list。

## KugouParams\_top\_song

请求参数

被如下接口引用: top\_song

名称	类型	描述
page	Integer	第?页。
size	Integer	每页?项。
top_id	String	榜单 id。

## KugouParams\_user\_info

请求参数这里为空, 传递空对象 {} 即可。

被如下接口引用: user\_info。

## KugouParams\_user\_logout

请求参数这里为空, 传递空对象 {} 即可。

被如下接口引用: user\_logout。

## KugouParams\_user\_qrcode\_auth

请求参数

被如下接口引用: user\_qrcode\_auth

名称	类型	描述
ticket	String	获取二维码授权信息的凭证。

## KugouParams\_user\_qrcode\_get

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：user\_qrcode\_get。

## KugouParams\_vip\_activity\_partner\_give

请求参数

被如下接口引用：vip\_activity\_partner\_give

名称	类型	必选	描述
partner_no	String	是	厂商订单号(同 sp 内唯一)。
month	Integer	是	购买月份数。
day	Integer	是	购买天数，天数和月数二选一，不可同时为0。

## KugouParams\_vip\_ksing\_info

请求参数这里为空，传递空对象 {} 即可。

被如下接口引用：vip\_ksing\_info。

## KugouParams\_vip\_ksing\_partner\_give

请求参数

被如下接口引用：vip\_ksing\_partner\_give

名称	类型	必选	描述
day	Integer	是	1. 购买天数，天数和月数二选一，不可同时为0。 2. 天数为1为酷狗线上测试充值渠道(请提前联系 TME 导入配额)，酷我Q音目前暂时不支持。
month	Integer	是	购买月份数。
partner_no	String	是	厂商订单号(同 sp 内唯一)。

## KugouParams\_vip\_svip\_common\_give

请求参数

被如下接口引用：vip\_svip\_common\_give

名称	类型	必选	描述

partner_no	String	是	厂商订单号(同 sp 内唯一)。
pay_no	String	是	用户支付订单号。
day	Integer	是	1. 购买天数, 天数和月数二选一, 不可同时为0。 2. 天数为1为酷狗线上测试充值渠道(请提前联系 TME 导入配额), 酷我Q音目前暂时不支持。
month	Integer	是	购买月份数。

## KugouParams\_vip\_tme\_partner\_give

请求参数

被如下接口引用: `vip_tme_partner_give`

名称	类型	必选	描述
userid	String	是	TME 用户 id。
token	String	否	TME 用户 token, 非首次充值支持免 token ( KW 和 QM 传过期或有效 token, KG 传有效 token 或空 )。
partner_no	String	是	厂商订单号(同 sp 内唯一)。
day	Integer	是	1. 购买天数, 天数和月数二选一, 不可同时为0。 2. 天数为1为酷狗线上测试充值渠道(请提前联系 TME 导入配额), 酷我Q音目前暂时不支持。
month	Integer	是	购买月份数。

## KugouParams\_wechat\_appinfocode

请求参数

被如下接口引用: `wechat_appinfocode`

名称	类型	描述
package	String	包名需事先向 TME 注册。

## KugouParams\_wechat\_minipauth

请求参数

被如下接口引用: `wechat_minipauth`

名称	类型	描述

auth_code	string	获取用户信息凭证。
-----------	--------	-----------

## KugouParams\_wechat\_openlink

### 请求参数

被如下接口引用: wechat\_openlink

名称	类型	描述
package	String	包名需事先向TME注册对应手机 APP 的包名（如没有对应手机 APP 可暂时预留一个）。

## KugouParams\_wechat\_qrcodeauth

### 请求参数

被如下接口引用: wechat\_qrcodeauth

名称	类型	描述
ticket	String	获取扫码授权结果的凭证。

## KugouParams\_wechat\_qrcodeget

### 请求参数

被如下接口引用: wechat\_qrcodeget

名称	类型	描述
package	String	包名需事先向 TME 注册。

## Location

### 经纬度

被如下接口引用: AppGetDeviceLocation、AppGetDeviceLocationHistory

名称	类型	描述
Longitude	float32	经度
Latitude	float32	纬度

## MemberList

被如下接口引用: AppGetFamilyMemberList

名称	类型	描述
UserId	String	用户 Id。
NickName	String	用户昵称。
Avatar	String	头像。
Role	Int	角色类型。 <ul style="list-style-type: none"><li>0是普通成员。</li><li>1是管理员。</li></ul>

## MessageAttachment

### 消息加信息

名称	类型	描述
ShareToken	String	分享设备或邀请成员加入家庭时，用于绑定的凭证。

## Msgs

### 消息体

名称	类型	描述
UserId	String	接受者用户 Id。
FromUserId	String	发送者用户 Id。
MsgId	String	消息 Id。
Category	Int	主类型（1: 设备, 2: 家庭, 3: 通知）。
MsgType	Int64	类型。 <ul style="list-style-type: none"><li>101: 设备告警提醒。</li><li>200: 成员添加结果通知。</li><li>201: 成员删除结果通知。</li><li>202: 社区成员添加结果通知。</li><li>203: 社区成员删除结果通知。</li><li>204: 邀请成员加入。</li><li>300: 设备分享结果通知。</li></ul>

- 301: 设备分享邀请。

MsgTitle	String	消息标题。
MsgContent	String	消息内容。
MsgTimestamp	Int64	消息时间戳。
ProductId	String	产品 ID。
DeviceName	String	设备名。
DeviceAlias	String	设备别名。
FamilyId	String	家庭 ID。
FamilyName	String	家庭名称。
RelateUserId	String	关联用户，在家庭消息中，一般指被添加或删除的用户。
CreateAt	String	创建时间，RFC3339 格式。
Attachments	Object of <a href="#">MessageAttachment</a>	消息附加信息。

## Position

### 位置

被如下接口引用：[AppGetDeviceLocation](#)、[AppGetDeviceLocationHistory](#)

名称	类型	描述
Location	<a href="#">Location</a>	经纬度。
CreateTime	int64	创建时间。

## ProductList

### 关联产品列表

名称	类型	描述
ProductId	String	产品 ID。
Name	String	产品名称。
Description	String	产品描述。
UIConfig	String	交互配置信息 json 字符串，包括产品图片、配网引导等。

## Products

### 产品列表

名称	类型	描述
ProductId	String	产品 ID。
Name	String	产品名称。
Description	String	产品描述。
DataTemplate	String	产品数据模板。
NetType	String	产品网络连接类型。
CategoryId	Int32	产品分类 ID。
ProductType	Int32	产品类型。
UpdateTime	Int64	最后更新的 unix 时间戳（秒级）。

## ProductsConfigRsp

### 产品界面配置返回值

被如下接口引用：AppGetProductsConfig

名称	类型	描述
ProductId	String	产品ID
Config	String	产品配置

## Property

条件属性被如下接口引用：AppCreateAutomation、AppDescribeAutomation、AppModifyAutomation

名称	类型	必选	描述
ProductId	String	是	产品 ID。
DeviceName	String	是	设备名称。
AliasName	String	否	设备别名。
IconUrl	String	否	产品图标 URL。
PropertyId	String	否	产品属性 ID。
Op	String	是	条件操作符。 <ul style="list-style-type: none"><li>• eq: 等于。</li><li>• ne: 不等于。</li><li>• gt: 大于。</li><li>• lt: 小于。</li><li>• ge: 大于等于。</li><li>• le: 小于等于。</li></ul>
Value	Any	是	比较的值。

## RiskItem

扣分项结构体被如下接口引用：AppGetInsuranceEvaluation

名称	类型	必选	描述
Name	String	是	扣分项名称。
Reason	String	是	隐患分析。
Suggestion	String	是	处置建议。

## Roomlist

房间列表信息被如下接口引用：AppGetRoomList

名称	类型	描述
RoomId	String	房间的所属 Id。
RoomName	String	房间名称。
DeviceNum	Int	房间拥有的设备数。
CreateTime	Int	创建时间, unix 时间戳(秒级)。
UpdateTime	Int	最后更新时间, unix 时间戳(秒级)。

## Scene

场景实体被如下接口引用: AppGetSceneList

名称	类型	必选	描述
SceneId	String	是	手动智能联动 ID。
FamilyId	String	是	手动智能联动所属家庭 ID。
SceneName	String	是	手动智能联动的名称。
SceneIcon	String	否	手动智能联动的背景图片地址。
Actions	Array of <a href="#">Action</a>	是	手动智能联动动作列表。
UserId	String	是	用户 ID。
CreateTime	int	否	创建时间戳。
UpdateTime	int	否	更新时间戳。
Flag	int	否	设备状态。 • 0: 正常。 • 1: 设备异常。
Status	int	否	运行状态。

- 0: 未运行。
- 1: 运行中。

## ShareDeviceInfo

响应结果被如下接口引用: AppDescribeShareDeviceToken

名称	类型	描述
FromUserId	String	分享用户 Id。
FromUserNick	String	分享用户昵称。
UserId	String	被分享用户 Id。
UserNick	String	被分享用户昵称。
ProductId	String	产品 ID。
DeviceName	String	设备名。
AliasName	String	设备别名。
IconUrl	String	图标 URL。
BindTime	Int64	绑定的 unix 时间戳 (秒级)。
ExpireTime	Int64	有效期 (秒级)。
CreateTime	Int64	创建的 unix 时间戳(秒级)。

## ShareDevices

分享设备信息被如下接口引用: AppDescribeShareDevice、AppListUserShareDevices

名称	类型	描述
ProductId	String	产品 ID。
DeviceName	String	设备名称。

DeviceId	String	设备 ID，是产品 ID/设备名称拼接而来。
IconUrl	String	图标的 url。
AliasName	String	设备别名、备注。
CreateTime	String	设备创建时间，这里是 RFC3339 的格式。

## ShareUserInfo

分享用户信息被如下接口引用：AppListShareDeviceUsers

名称	类型	描述
UserId	String	被分享用户 Id。
NickName	String	昵称。
CountryCode	String	国家码。
PhoneNumber	String	手机号。
BindTime	Int64	绑定时间，秒级 UNIX 时间戳。

## Timer

条件触发时间被如下接口引用：AppCreateAutomation、AppDescribeAutomation、AppModifyAutomation

名称	类型	必选	描述
Days	String	是	定时器开启时间，每一位——0:关闭，1:开启，从左至右依次表示：周日、周一、周二、周三、周四、周五、周六。
TimePoint	String	是	定时器开启时间点。

## TimerList

定时器列表信息被如下接口引用：AppGetTimerList

名称	类型	描述

TimerId	String	定时器 ID。
TimerName	String	定时器名称。
ProductId	String	定时器所属的产品 ID。
DeviceName	String	定时器对应的设备名称。
Days	String	定时器开启时间，每一位——0是关闭，1是开启，从左至右依次表示：周日、周一、周二、周三、周四、周五、周六。
TimePoint	String	定时器开启时间点。
Repeat	Int	是否循环。 • 0: 不需要。 • 1: 需要。
Data	String	定时器启动时下发的状态。
Status	String	开启状态。0表示关闭，1表示开启。
CreateTime	Int64	创建时间，unix 时间戳(秒级)。
UpdateTime	Int64	最后一次更新时间，unix 时间戳(秒级)。

## TokenResultInfo

用户信息被如下接口引用：[AppGetToken](#)、[AppGetTokenByWeiXin](#)

名称	类型	描述
ExpireAt	Int64	截止时间，Unix 秒级时间戳。
Token	String	开发平台返回的 AccessToken，通过该 Token 进行登录后的接口请求。
CancelAccountTime	Int64	仅当用户提交注销申请（AppUserCancelAccount）后再次登录时返回（仅申请首次登录返回），此字段用来返回注销申请提交时间，提醒用户注销流程已取消，Unix 秒级时间戳。

## UserSetting

用户设置被如下接口引用：[App GetUserSetting](#)

名称	类型	必选	描述
EnableWechatPush	Int	否	微信推送。
EnableDeviceMessagePush	Int	否	设备信息推送。
EnableFamilyMessagePush	Int	否	家庭信息推送。
EnableNotifyMessagePush	Int	否	通知信息推送。
AllowEditions	Int	否	小程序白名单版本。0表示仅限家庭版，1表示家庭版和社区版都允许。
UsingEdition	Int	否	小程序当前使用版本。0表示家庭版，1表示社区版。
TemperatureUnit	String	否	温度单位：C 表示摄氏度，F 表示华氏度。
Region	String	否	时区设置，有效时区参见： AppGetGlobalConfig 中的 RegionListEN 或 RegionListCN。
NoDisturbTime	String	否	消息免打扰时段，为空则表示未开启免打扰，开启时则用：HH:MM–HH:MM 表示免打扰时段，例如：21:00–08:00。
NoDisturbDeviceList	Array of String	否	免打扰设备 ID 列表。
NoDisturbFamilyList	Array of String	否	免打扰家庭列表。

## YunApiParam

设备心跳接口被如下接口引用：HeartBeat

名称	类型	必选	描述
AccessToken	String	是	AccessToken 为调用登录接口之后获取 Token。
ActionParams	Object of HeartBeatPara	是	设备心跳接口入参。

m			
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。

## accompany\_accompany\_free\_info

体验信息被如下接口引用：accompany\_free\_info

名称	类型	必选	描述
accompan y_id	String	否	已经体验的伴奏 id。

## accompany\_accompany\_info

伴奏列表被如下接口引用：accompany\_info

名称	类型	必选	描述
has_origin al	Integer	否	音频是否有原唱音轨。 <ul style="list-style-type: none"><li>• 0: 无。</li><li>• 1: 有。</li></ul>
is_hq	Integer	是	是否显示 HQ 标识。 <ul style="list-style-type: none"><li>• 0: 不显示。</li><li>• 1: 显示。</li></ul>
mv_id	String	是	MVID
hash_key	String	是	hashkey
song_id	String	是	歌曲 id。
accompan y_id	String	是	伴奏 id。
song_nam e	String	是	伴奏名。
singer_id	String	是	歌手 id。
singer_na me	String	是	歌手名。

duration	Integer	是	伴奏时长单位毫秒。
album_img	String	是	专辑图片。
has_origin al	Integer	是	音频是否有原唱音轨。 <ul style="list-style-type: none"><li>• 0: 无。</li><li>• 1: 有。</li></ul>
is_ktv	Integer	是	是否有 KTV。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>
krc_id	String	是	歌词 ID。
original_mv_id	String	是	原版本 MV ID。

## accompany\_awesome\_personal

伴奏个性推荐

被如下接口引用: accompany\_awesome\_personal

名称	类型	必选	描述
is_hq	Integer	是	是否显示 HQ 标识。 <ul style="list-style-type: none"><li>• 0: 不显示。</li><li>• 1: 显示。</li></ul>
accompan y_id	String	是	伴奏 id。
song_nam e	String	是	伴奏名。
singer_id	String	是	歌手 id。
singer_na me	String	是	歌手名。

duration	Integer	是	伴奏时长单位毫秒。
album_img	String	是	专辑图片。
has_origin al	Integer	是	音频是否有原唱音轨。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>
is_ktv	Integer	是	是否有 KTV。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>

## accompany\_awesome\_top

伴奏推荐列表被如下接口引用: `accompany_awesome_top`

名称	类型	必选	描述
weight	Integer	是	热度值/飙升值。
is_hq	Integer	是	是否显示HQ标识。 <ul style="list-style-type: none"><li>• 0: 不显示。</li><li>• 1: 显示。</li></ul>
accompan y_id	String	是	伴奏 id。
song_nam e	String	是	伴奏名。
singer_id	String	是	歌手 id。
singer_na me	String	是	歌手名。
duration	Integer	是	伴奏时长单位毫秒。
album_img	String	是	专辑图片。

has_origin al	Integer	是	音频是否有原唱音轨。 • 0: 没有。 • 1: 有。
is_ktv	Integer	是	是否有 KTV。 • 0: 没有。 • 1: 有。
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 • 0: 没有。 • 1: 有。

## accompany\_category\_song

伴奏列表被如下接口引用: accompany\_category\_song

名称	类型	必选	描述
is_hq	Integer	是	是否显示 HQ 标识。 • 0: 不显示。 • 1: 显示。
accompan y_id	String	是	伴奏 id。
song_nam e	String	是	伴奏名。
singer_id	String	是	歌手 id。
singer_na me	String	是	歌手名。
duration	Integer	是	伴奏时长单位毫秒。
album_img	String	是	专辑图片。
has_origin al	Integer	是	音频是否有原唱音轨。 • 0: 没有。 • 1: 有。
is_ktv	Integer	是	是否有 KTV。 • 0: 没有。

			<ul style="list-style-type: none"><li>• 1: 有。</li></ul>
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>

## accompany\_favorite\_list

伴奏列表被如下接口引用：accompany\_favorite\_list

名称	类型	必选	描述
is_hq	Integer	是	是否显示 HQ 标识。 <ul style="list-style-type: none"><li>• 0: 不显示。</li><li>• 1: 显示。</li></ul>
has_origin al	Integer	否	音频是否有原唱音轨。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>
singer_na me	String	是	歌手名。
accompan y_id	String	是	伴奏 id。
song_nam e	String	是	伴奏名字。
duration	Integer	是	伴奏时长单位毫秒。
singer_id	String	是	歌手 id。
album_img	String	是	专辑图片。
is_ktv	Integer	是	是否有 KTV。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 <ul style="list-style-type: none"><li>• 0: 没有。</li></ul>

- 1: 有。

## accompany\_free\_top\_song

伴奏列表被如下接口引用：accompany\_free\_top\_song

名称	类型	必选	描述
is_hq	Integer	是	是否显示 HQ 标识。 <ul style="list-style-type: none"><li>• 0: 不显示。</li><li>• 1: 显示。</li></ul>
has_origin al	Integer	否	音频是否有原唱音轨。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>
singer_na me	String	是	歌手名。
free_token	String	是	免费令牌。
accompan y_id	String	是	伴奏 id。
song_nam e	String	是	伴奏名字。
duration	Integer	是	伴奏时长单位毫秒。
singer_id	String	是	歌手 id。
album_img	String	是	专辑图片。
is_ktv	Integer	是	是否有 KTV。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 <ul style="list-style-type: none"><li>• 0: 没有。</li><li>• 1: 有。</li></ul>

## accompany\_newsearch\_song

伴奏列表被如下接口引用：accompany\_newsearch\_song

名称	类型	必选	描述
song_hot	Integer	是	热度。
accompan y_id	String	是	伴奏 id。
song_nam e	String	是	伴奏名。
singer_id	String	是	歌手 id。
singer_na me	String	是	歌手名。
duration	Integer	是	伴奏时长单位毫秒。
album_img	String	是	专辑图片。
is_ktv	Integer	是	是否有 KTV。 • 0: 没有。 • 1: 有。
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 • 0: 没有。 • 1: 有。

## accompany\_search\_song

伴奏搜索被如下接口引用: accompany\_search\_song

名称	类型	必选	描述
song_hot	Integer	是	热度。
accompan y_id	String	是	伴奏 id。
song_nam e	String	是	伴奏名。
singer_id	String	是	歌手 id。
singer_na me	String	是	歌手名。

duration	Integer	是	伴奏时长单位毫秒。
album_im g	String	是	专辑图片。
has_origin al	Integer	是	音频是否有原唱音轨。 • 0: 无。 • 1: 有。
is_ktv	Integer	是	是否有 KTV。 • 0: 没有。 • 1: 有。
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 • 0: 没有。 • 1: 有。

## accompany\_singer\_song

伴奏列表被如下接口引用：accompany\_singer\_song

名称	类型	必选	描述
is_hq	Integer	是	是否显示 HQ 标识。 • 0: 不显示。 • 1: 显示。
has_origin al	Integer	否	音频是否有原唱音轨。 • 0: 无。 • 1: 有。
singer_na me	String	是	歌手名。
accompan y_id	String	是	伴奏 id。
song_na me	String	是	伴奏名字。
duration	Integer	是	伴奏时长单位毫秒。
singer_id	String	是	歌手 id。

album_im g	String	是	专辑图片。
is_ktv	Integer	是	是否有 KTV。 • 0: 没有。 • 1: 有。
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 • 0: 没有。 • 1: 有。

## accompany\_song\_accompany

伴奏

被如下接口引用: song\_accompany

名称	类型	描述
accompan y_id	String	伴奏 id。
song_nam e	String	伴奏名。
singer_id	String	歌手 id。
singer_na me	String	歌手名。
duration	Integer	伴奏时长单位毫秒。
album_im g	String	专辑图片。
has_origin al	Integer	音频是否有原唱音轨。 • 0: 无。 • 1: 有。
is_ktv	Integer	是否有 KTV。 • 0: 没有。 • 1: 有。
bit_rate	Integer	比特率。

has_pitch	Integer	是否有音高。 • 0: 没有。 • 1: 有。
-----------	---------	-------------------------------

## accompany\_theme\_song

伴奏列表

被如下接口引用: accompany\_theme\_song

名称	类型	必选	描述
is_hq	Integer	是	是否显示 HQ 标识。 • 0: 不显示。 • 1: 显示。
has_origin al	Integer	否	音频是否有原唱音轨。 • 0: 无。 • 1: 有。
singer_na me	String	是	歌手名。
accompan y_id	String	是	伴奏 id。
song_nam e	String	是	伴奏名字。
duration	Integer	是	伴奏时长单位毫秒。
singer_id	String	是	歌手 id。
album_im g	String	是	专辑图片。
is_ktv	Integer	是	是否有 KTV。 • 0: 没有。 • 1: 有。
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 • 0: 没有。 • 1: 有。

## accompany\_top\_song

### 伴奏列表

被如下接口引用: accompany\_top\_song

名称	类型	必选	描述
is_hq	Integer	是	是否显示 HQ 标识。 <ul style="list-style-type: none"><li>0: 不显示。</li><li>1: 显示。</li></ul>
has_origin al	Integer	否	音频是否有原唱音轨。 <ul style="list-style-type: none"><li>0: 无。</li><li>1: 有。</li></ul>
singer_na me	String	是	歌手名。
accompan y_id	String	是	伴奏 id。
song_nam e	String	是	伴奏名字。
duration	Integer	是	伴奏时长单位毫秒。
singer_id	String	是	歌手 id。
album_im g	String	是	专辑图片。
is_ktv	Integer	是	是否有 KTV。 <ul style="list-style-type: none"><li>0: 没有。</li><li>1: 有。</li></ul>
bit_rate	Integer	是	比特率。
has_pitch	Integer	是	是否有音高。 <ul style="list-style-type: none"><li>0: 没有。</li><li>1: 有。</li></ul>

## albums\_asset\_purchased\_albums

### 专辑信息

被如下接口引用: asset\_purchased\_albums

名称	类型	描述
album_id	String	专辑 id。
album_name	String	专辑名。
album_translate_name	String	专辑译名。
album_img	String	专辑封面。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。
album_img_medium	String	专辑封面500px左右。
intro	String	专辑简介。
company	String	唱片公司。
singer_id	String	歌手 id。
singer_name	String	歌手名。
publish_time	String	发布时间.

## albums\_singer\_album

### 专辑信息

被如下接口引用: singer\_album

名称	类型	描述
album_id	String	专辑 id。
album_name	String	专辑名。
album_translate_name	String	专辑译名。
album_img	String	专辑封面。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。
album_img_medium	String	专辑封面500px左右。
intro	String	专辑简介。

company	String	唱片公司。
singer_id	String	歌手 id。
singer_name	String	歌手名。
publish_time	String	发布时间。

## appapiCommon

### AppApi 公共参数

名称	类型	必选	描述
Action	String	是	公共参数，本接口取值：AppCreateEmailUser。
RequestId	String	是	公共参数，唯一请求 ID，可自行生成，推荐使用 uuld。定位问题时，需提供该次请求的 RequestId。
AppKey	String	是	公共参数，应用 AppKey，用于标识对应的 App。
Signature	String	是	公共参数，请求签名，需用户自行生成，用于校验请求的合法性。
Timestamp	Int64	是	公共参数，请求的 Unix 时间戳（秒级）。
Nonce	Int	是	公共参数，随机正整数，与 Timestamp 联合起来，防止重放攻击。

## author\_newsearch\_song

### 歌手列表

被如下接口引用：accompany\_newsearch\_song

名称	类型	必选	描述
singer_id	String	是	歌手 id。
singer_name	String	是	歌手名。
singer_img	String	是	歌手图片。

## categories\_accompany\_category\_list

### 分类

被如下接口引用：accompany\_category\_list

名称	类型	必选	描述
category_name	String	是	分类名。
category_id	String	是	分类 id。

## categories\_mv\_category\_list

### 分类列表

被如下接口引用: mv\_category\_list

名称	类型	必选	描述
category_id	String	是	分类 id。
category_name	String	是	分类名。

## data\_monitor\_delay

### 数据

被如下接口引用: monitor\_delay

名称	类型	必选	描述
api	String	是	api path, 必须与 api 实际 path 保持一致。
userid	String	是	TME 用户 id。
client_ip	String	是	接口使用者 ip, 如设备直接调用 TME api 则上报设备 ip, 反之则厂商服务器 ip。
server_ip	String	是	接口实际调用时的服务器 ip。
ecode	Integer	是	业务错误码 ( error_code )。
reqtime	Integer	是	请求时间时间戳单位秒。
rspsize	Integer	是	响应包大小, 单位字节。
sp	String	是	服务提供商对应枚举值分别为酷狗, 酷我, Q音。
apn	String	是	网络类型。
etype	Integer	是	错误类型。0: 调用成功。1: 网络错误。2: http 错误码。3: 业务错误码。

reqsize	Integer	是	请求包大小, 单位字节。
retry	Integer	是	重试次数 (本次是第几次重试)。
delay	Integer	是	接口调用时延, 单位 ms。

## data\_monitor\_listen

### 数据

被如下接口引用: monitor\_listen

名称	类型	必选	描述
play_quality	String	否	歌曲/长音频播放品质 d, 取值: SQ、HQ、standard; 视频播放品质: FHD、HD、LD、QHD、SD。
duration	Integer	是	歌曲时长 (单位 ms)。
play_time	Integer	是	歌曲实际播放时长 (单位 ms)。
play_type	Integer	是	<ul style="list-style-type: none"> <li>1: 在线。</li> <li>2: 缓存。</li> <li>3: 下载。</li> </ul>
client_ip	String	是	接口实际使用者 ip, 即硬件终端 ip。
media_type	Integer	否	<ul style="list-style-type: none"> <li>1: 听歌。</li> <li>2: 视频MV。</li> <li>3: 伴奏K歌业务。</li> <li>4: 有声音频。</li> <li>5: 直播。</li> </ul>
lvt	String	是	用户播放时间 format: 2019-09-19 00:00:00。
try_play	Integer	否	0: 可完整播放。1: vip 试听类型。2: 免登录试听。 说明: 可完整播放 (该用户有完播权限), vip 试听类型 (非 vip 用户试听 vip 歌曲片段), 免登录试听 (用户未登录, 免费试听权限)。
sp	String	是	服务提供商对应枚举值分别为酷狗, 酷我, Q音。
userid	String	是	TME 用户 id。
song_id	String	是	根据多媒体类型, 赋值相应的 ID: 歌曲 ID、伴奏 ID、MV 的 ID。

api	String	是	api path, 必须与 api 实际 path 保持一致。
source_id	String	是	歌曲 id 来源, 歌曲来源为歌单上传歌单 id, 歌曲来源为榜单, 上传榜单 id。如果都不是, 传空。

## groups\_accompany\_category\_list

伴奏分组

被如下接口引用: accompany\_category\_list

名称	类型	必选	描述
group_name	String	否	分组名。
categories	Array Of categories_accompany_category_list of <a href="#">categories_accompany_category_list</a>	否	分类。

## groups\_accompany\_free\_top\_list

分组列表

被如下接口引用: accompany\_free\_top\_list

名称	类型	必选	描述
group_id	String	是	分组 id。
group_name	String	是	分组名。
tops	Array Of tops_accompany_free_top_list of <a href="#">tops_accompany_free_top_list</a>	否	榜单列表。

## groups\_accompany\_top\_list

分组列表

被如下接口引用: accompany\_top\_list

名称	类型	必选	描述
group_id	String	是	分组 id。
group_name	String	是	分组名。

tops	Array Of tops_accompany_top_list of <a href="#">tops_accompany_top_list</a>	否	tops
------	--	---	------

## groups\_child\_home

资源分组

被如下接口引用: [child\\_home](#)

名称	类型	必选	描述
name	String	否	资源名称。
infos	Array Of infos_child_home of <a href="#">infos_child_home</a>	否	资源列表。

## groups\_longaudio\_free\_home

资源分组

被如下接口引用: [longaudio\\_free\\_home](#)

名称	类型	必选	描述
name	String	否	资源名称。
tag_id	String	否	资源 tag_id, 值为-1 即没有更多的资源。
infos	Array Of infos_longaudio_free_home of <a href="#">infos_longaudio_free_home</a>	否	资源列表。

## groups\_playlist\_top

分组列表

被如下接口引用: [playlist\\_top](#)

名称	类型	描述
category_id	String	分类 id 推荐歌单传入同名字段获取相应歌单。
category_name	String	分类名。
category_icon	String	分类封面。

## groups\_radio\_list

分组列表

被如下接口引用: radio\_list

名称	类型	描述
group_id	String	分组 id。
group_name	String	分组名。
radios	Array of <a href="#">radios_radio_list</a>	电台列表。

## groups\_rigion\_home

资源分组

被如下接口引用: rigion\_home

名称	类型	必选	描述
name	String	否	资源名称。
infos	Array Of <a href="#">infos_rigion_home</a> of <a href="#">infos_rigion_home</a>	否	资源列表。

## groups\_rigion\_oldman\_index

资源分组

被如下接口引用: rigion\_oldman\_index

名称	类型	必选	描述
id	Integer	否	专区 id。
name	String	否	专区名称。
pic	String	否	专区封面。
summary	String	否	描述。

## groups\_top\_list

分组

被如下接口引用: top\_list

名称	类型	描述
group_id	String	分组 id。
group_name	String	分组名。
tops	Array of <a href="#">tops_top_list</a>	榜单。

## horizontal\_live\_room\_url

横屏流信息

被如下接口引用：[live\\_room\\_url](#)

名称	类型	必选	描述
rate	String	否	清晰度标识 (1: 流畅。2: 标清。3: 高清。4: 超清。5: 1080P。)
hls	Array Of String	否	hls 流信息。
httpflv	Array Of String	否	httpflv 流信息。
httpsflv	Array Of String	否	httpsflv 流信息。
rtmp	Array Of String	否	rtmp 流信息。
httpshls	Array Of String	否	httpshls 流信息。
streamName	String	否	流名称。

## infos\_child\_home

资源列表

被如下接口引用：[child\\_home](#)

名称	类型	必选	描述
resource_t	String	是	大分类。

type			3: 歌单。4: 专辑。7: 小分类。
resource_id	String	是	资源 id。
name	String	是	资源名称。
pic	String	是	资源图。

## infos\_longaudio\_free\_home

### 资源列表

被如下接口引用: longaudio\_free\_home

名称	类型	必选	描述
resource_id	Integer	是	资源 id。
name	String	是	资源名称。
pic	String	是	资源封面。

## infos\_rigion\_home

### 资源列表

被如下接口引用: rigion\_home

名称	类型	必选	描述
resource_type	String	是	大分类。 2: 单曲。3: 歌单。103: 歌单合集。4: 专辑。 104: 专辑合集。5: 歌手。6: 排行榜。7: 分类。 8: MV。108: MV 合集。9: 电台。109: 电台合集。
resource_id	String	是	资源 id。
name	String	是	资源名称。
pic	String	是	资源封面。

## letters\_accompany\_singer\_list

### letters

被如下接口引用: accompany\_singer\_list

名称	类型	必选	描述
letter	String	是	歌手首字母。
singers	Array Of singers_accompany_singer_list of <a href="#">singers_accompany_singer_list</a>	是	歌手列表。

## lists\_child\_resource

资源分组

被如下接口引用: `child_resource`

名称	类型	必选	描述
id	String	是	资源 id。
name	String	是	资源名称。
pic	String	是	资源图片。

## lists\_longaudio\_resource

资源列表

被如下接口引用: `longaudio_resource`

名称	类型	必选	描述
resource_id	Integer	是	资源 id。
name	String	是	资源名称。
pic	String	是	资源封面。

## mvInfos\_mv\_infos

mv 信息

被如下接口引用: `mv_infos`

名称	类型	必选	描述
author_name	String	否	MV 制作人名称。
tags	<a href="#">tags_mv_infos</a>	否	mv 信息。

mv_id	String	是	MV id
mv_name	String	是	MV 名称。
singer_id	String	是	歌手 id。
singer_name	String	是	歌手名称。
accompany_id	String	否	伴奏 id。
intro	String	否	简介。
song_id	String	是	歌曲 id。
mv_img	String	是	MV 图片。

## mvs\_mv\_category\_mv

MV 列表

被如下接口引用：mv\_category\_mv

名称	类型	必选	描述
mv_img	String	是	MV 图片。
mv_name	String	是	MV 名称。
singer_id	String	是	歌手 id。
singer_name	String	是	歌手名称。
song_id	String	是	歌曲 id。
mv_id	String	是	MV id

## mvs\_mv\_live\_tme

MV 列表

被如下接口引用：mv\_live\_tme

名称	类型	必选	描述
mv_id	String	是	MV id

mv_img	String	是	MV 图片。
mv_name	String	是	MV 名称。
singer_id	String	是	歌手 id。
singer_name	String	是	歌手名称。
song_id	String	是	歌曲 id。

## mvs\_mv\_search

mv 列表

被如下接口引用: mv\_search

名称	类型	必选	描述
singer_name	String	是	歌手名称。
has_acc	Integer	是	是否有伴奏。 ● 0: 没有。 ● 1: 有。
mv_id	String	是	MV id
mv_img	String	是	MV 图片。
mv_name	String	是	MV 名称。
song_id	String	是	歌曲 id。

## opuses\_accompany\_opus\_list

opuses

被如下接口引用: accompany\_opus\_list

名称	类型	必选	描述
opus_name	String	是	作品名称。
score	Integer	否	总分数。
grade	String	否	等级。

listen_num	Integer	是	试听人数。
album_img	String	是	专辑封面。
accompan y_id	String	是	伴奏 id。
opus_id	String	是	作品 id。
average_s core	Integer	否	平均分。
is_private	Integer	是	是否公开。 • 0: 公开作品。 • 1: 私密作品。
duration	Integer	是	作品播放时间, 单位秒。

## playlists\_favorite\_other\_list

歌单

被如下接口引用: favorite\_other\_list

名称	类型	描述
playlist_id	String	歌单 id。
playlist_na me	String	歌单名称。
pic	String	歌单封面。
playlist_ext ra_id	String	歌单附加 id。

## playlists\_favorite\_self\_list

歌单

被如下接口引用: favorite\_self\_list

名称	类型	描述
playlist_id	String	歌单 id。
playlist_na me	String	歌单名称。

pic	String	歌单封面。
update_time	String	歌单更新时间。
create_time	String	歌单创建时间。
total	Integer	歌单歌曲数量。

## playlists\_playlist\_awesome

歌单

被如下接口引用：playlist\_awesome

名称	类型	描述
playlist_id	String	歌单 id。
playlist_name	String	歌单名称。
pic	String	歌单封面。
update_time	String	歌单更新时间。
create_time	String	歌单创建时间。
intro	string	歌单简介。

## programs\_live\_program\_list

节目分组

被如下接口引用：live\_program\_list

名称	类型	必选	描述
room_id	Integer	是	房间 id。
status	String	是	节目状态。0是未开播，1是开播。
price	String	否	价格。
name	String	是	节目名称。

startTime	String	是	开始时间。
endTime	String	是	结束时间。
intro	String	是	节目描述。
pic	String	是	封面图片。
pay_type	Integer	否	是否付费演出会。0: 不是。1: 是。

## **radios\_radio\_list**

电台列表

被如下接口引用: `radio_list`

名称	类型	描述
radio_id	String	电台 id。
radio_name	String	电台名。
radio_img	String	电台封面。

## **recommends\_accompany\_category\_list**

推荐分类

被如下接口引用: `accompany_category_list`

名称	类型	必选	描述
category_id	String	是	分类 id。
category_name	String	是	分类名。
category_img	String	是	分类图片。

## **singers\_accompany\_singer\_list**

热门歌手

被如下接口引用: `accompany_singer_list`

名称	类型	必选	描述

singer_id	String	是	歌手 id。
singer_name	String	是	歌手名。
singer_img	String	是	歌手图片。

## singers\_accompany\_singer\_list\_1

歌手列表

名称	类型	必选	描述
singer_name	String	是	歌手名。
singer_img	String	是	歌手图片。
singer_id	String	是	歌手 id。

## singers\_singer\_list

歌手

被如下接口引用: singer\_list

名称	类型	描述
singer_id	String	歌手 id。
singer_name	String	歌手名。
area	String	歌手地区。
translate_name	String	歌手译名。
singer_img	String	歌手相片。

## slots\_search\_voice

语音搜索列表

被如下接口引用: search\_voice

名称	类型	必选	描述

name	String	是	-
value	String	是	词槽参数值。

## songs\_album\_info

歌曲列表

被如下接口引用: album\_info

名称	类型	描述
singer_img	String	歌手头像。
album_img	String	专辑封面。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。
album_img_medium	String	专辑封面500px左右。
is_vip_song	Integer	是否vip歌曲。0: 否。1: 是。
mv_id	String	mv id
has_accompany	Integer	是否有伴奏。0: 否。1: 是。
song_id	String	歌曲 id。
song_name	String	歌曲名。
singer_id	String	歌手 id。
singer_name	String	歌手名。
album_id	String	专辑 id。
album_name	String	专辑名。

playable_code	Integer	是否可以播放。枚举值依次对应： <ul style="list-style-type: none"> <li>0: 可以。</li> <li>1: 海外地区不能播放。</li> <li>2: 歌曲无版权不能播放。</li> <li>3: 会员歌曲, 非会员不能播放。</li> <li>4: 付费内容, 须购买才可播放。</li> <li>5: 牛方案策略, 非会员不能播放。</li> <li>6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。</li> <li>9: 未知原因, 无权播放。</li> </ul>
tryPlayable	Integer	是否有试听片段。 <ul style="list-style-type: none"> <li>0: 否。</li> <li>1: 有。</li> </ul>
language	String	语言。

## songs\_asset\_purchased\_songs

### 歌曲列表

被如下接口引用: asset\_purchased\_songs

名称	类型	描述
singer_img	String	歌手头像。
album_img	String	专辑封面。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。
album_img_medium	String	专辑封面500px左右。
is_vip_song	Integer	是否 vip 歌曲。 <ul style="list-style-type: none"> <li>0: 否。</li> <li>1: 是。</li> </ul>
mv_id	String	mv id

has_accompany	Integer	是否有伴奏。 ● 0: 否。 ● 1: 是。
song_id	String	歌曲 id。
song_name	String	歌曲名。
singer_id	String	歌手 id。
singer_name	String	歌手名。
album_id	String	专辑 id。
album_name	String	专辑名。
playable_code	Integer	是否可以播放。枚举值依次对应： ● 0: 可以。 ● 1: 海外地区不能播放。 ● 2: 歌曲无版权不能播放。 ● 3: 会员歌曲, 非会员不能播放。 ● 4: 付费内容, 须购买才可播放。 ● 5: 牛方案策略, 非会员不能播放。 ● 6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。 ● 9: 未知原因, 无权播放。
try_playable	Integer	是否有试听片段。 ● 0: 否。 ● 1: 是。
language	String	语言。

## songs\_awesome\_everyday

### 歌曲列表

被如下接口引用: awesome\_everyday

名称	类型	描述
singer_img	String	歌手头像。

album_img	String	专辑封面。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。
album_img_medium	String	专辑封面500px左右。
is_vip_son g	Integer	是否 vip 歌曲。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 是。</li></ul>
mv_id	String	mv id
has_accom pany	Integer	是否有伴奏。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 是。</li></ul>
song_id	String	歌曲 id。
song_nam e	String	歌曲名。
singer_id	String	歌手 id。
singer_na me	String	歌手名。
album_id	String	专辑 id。
album_na me	String	专辑名。
playable_c ode	Integer	是否可以播放。枚举值依次对应： <ul style="list-style-type: none"><li>0: 可以。</li><li>1: 海外地区不能播放。</li><li>2: 歌曲无版权不能播放。</li><li>3: 会员歌曲, 非会员不能播放。</li><li>4: 付费内容, 须购买才可播放。</li><li>5: 牛方案策略, 非会员不能播放。</li><li>6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情 况)。</li></ul>

		● 9: 未知原因, 无权播放。
try_playable	Integer	是否有试听片段。 ● 0: 否。 ● 1: 有。
language	String	语言。

## songs\_awesome\_newsong

### 歌曲列表

被如下接口引用: awesome\_newsong

名称	类型	描述
singer_img	String	歌手头像。
album_img	String	专辑封面。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。
album_img_medium	String	专辑封面500px左右。
is_vip_song	Integer	是否 vip 歌曲。 ● 0: 否。 ● 1: 是。
mv_id	String	mv id
has_accompany	Integer	是否有伴奏。 ● 0: 否。 ● 1: 是。
song_id	String	歌曲 id。
song_name	String	歌曲名。
singer_id	String	歌手 id。

singer_name	String	歌手名。
album_id	String	专辑 id。
album_name	String	专辑名。
playable_code	Integer	是否可以播放。枚举值依次对应： <ul style="list-style-type: none"> <li>● 0: 可以。</li> <li>● 1: 海外地区不能播放。</li> <li>● 2: 歌曲无版权不能播放。</li> <li>● 3: 会员歌曲, 非会员不能播放。</li> <li>● 4: 付费内容, 须购买才可播放。</li> <li>● 5: 牛方案策略, 非会员不能播放。</li> <li>● 6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。</li> <li>● 9: 未知原因, 无权播放。</li> </ul>
tryPlayable	Integer	是否有试听片段。 <ul style="list-style-type: none"> <li>● 0: 否。</li> <li>● 1: 有。</li> </ul>
language	String	语言。

## **songs\_awesome\_recommend**

歌曲列表

被如下接口引用: awesome\_recommend

名称	类型	描述
singer_img	String	歌手头像。
album_img	String	专辑封面。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。
album_img_medium	String	专辑封面500px左右。

is_vip_son g	Integer	是否 vip 歌曲。 ● 0: 否。 ● 1: 是。
mv_id	String	mv id
has_accom pany	Integer	是否有伴奏。 ● 0: 否。 ● 1: 是。
song_id	String	歌曲 id。
song_nam e	String	歌曲名。
singer_id	String	歌手 id。
singer_na me	String	歌手名。
album_id	String	专辑 id。
album_na me	String	专辑名。
playable_c ode	Integer	是否可以播放。枚举值依次对应： ● 0: 可以。 ● 1: 海外地区不能播放。 ● 2: 歌曲无版权不能播放。 ● 3: 会员歌曲, 非会员不能播放。 ● 4: 付费内容, 须购买才可播放。 ● 5: 牛方案策略, 非会员不能播放。 ● 6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情 况)。 ● 9: 未知原因, 无权播放。
try_playabl e	Integer	是否有试听片段。 ● 0: 否。 ● 1: 有。
language	String	语言。

## songs\_favorite\_song

## 歌曲列表

被如下接口引用: `favorite_song`

名称	类型	描述
<code>song_id</code>	<code>String</code>	歌曲 id。
<code>song_name</code>	<code>String</code>	歌曲名。
<code>singer_id</code>	<code>String</code>	歌手 id。
<code>singer_name</code>	<code>String</code>	歌手名。
<code>album_id</code>	<code>String</code>	专辑 id。
<code>album_name</code>	<code>String</code>	专辑名。
<code>song_extra_id</code>	<code>String</code>	歌曲附加 id, 仅获取自建歌单歌曲列表时返回 (取消收藏时使用)。
<code>playable_code</code>	<code>Integer</code>	是否可以播放。枚举值依次对应: <ul style="list-style-type: none"> <li>● 0: 可以。</li> <li>● 1: 海外地区不能播放。</li> <li>● 2: 歌曲无版权不能播放。</li> <li>● 3: 会员歌曲, 非会员不能播放。</li> <li>● 4: 付费内容, 须购买才可播放。</li> <li>● 5: 牛方案策略, 非会员不能播放。</li> <li>● 6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。</li> <li>● 9: 未知原因, 无权播放。</li> </ul>
<code>is_vip_song</code>	<code>Integer</code>	是否 vip 歌曲。 <ul style="list-style-type: none"> <li>● 0: 否。</li> <li>● 1: 是。</li> </ul>

**`songs_id_song_infos`**

## 歌曲id列表

被如下接口引用: `song_infos`

名称	类型	必选	描述

Values	Array of String	是	一个或者多个过滤值。
--------	-----------------	---	------------

## songs\_playlist\_song

### 歌曲列表

被如下接口引用：playlist\_song

名称	类型	描述
playable_code	Integer	是否可以播放。枚举值依次对应： <ul style="list-style-type: none"><li>0: 可以。</li><li>1: 海外地区不能播放。</li><li>2: 歌曲无版权不能播放。</li><li>3: 会员歌曲, 非会员不能播放。</li><li>4: 付费内容, 须购买才可播放。</li><li>5: 牛方案策略, 非会员不能播放。</li><li>6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。</li><li>9: 未知原因, 无权播放。</li></ul>
song_id	String	歌曲 id。
song_name	String	歌曲名。
singer_id	String	歌手 id。
singer_name	String	歌手名。
album_id	String	专辑 id。
album_name	String	专辑名。
is_vip_song	Integer	是否 vip 歌曲。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 是。</li></ul>

## songs\_radio\_song

### 歌曲列表

被如下接口引用：radio\_song

名称	类型	描述
singer_im g	String	歌手头像。
album_im g	String	专辑封面。
album_im g_mini	String	专辑封面100px左右。
album_im g_small	String	专辑封面300px左右。
album_im g_medium	String	专辑封面500px左右。
is_vip_son g	Integer	是否 vip 歌曲。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 是。</li></ul>
mv_id	String	mv id
has_acco mpany	Integer	是否有伴奏。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 是。</li></ul>
song_id	String	歌曲 id。
song_nam e	String	歌曲名。
singer_id	String	歌手 id。
singer_na me	String	歌手名。
album_id	String	专辑 id。
album_na me	String	专辑名。
playable_c ode	Integer	是否可以播放。枚举值依次对应： <ul style="list-style-type: none"><li>0: 可以。</li><li>1: 海外地区不能播放。</li><li>2: 歌曲无版权不能播放。</li><li>3: 会员歌曲，非会员不能播放。</li></ul>

		<ul style="list-style-type: none"><li>4: 付费内容, 须购买才可播放。</li><li>5: 牛方案策略, 非会员不能播放。</li><li>6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。</li><li>9: 未知原因, 无权播放。</li></ul>
language	String	语言。

## songs\_search\_song

### 歌曲列表

被如下接口引用: `search_song`

名称	类型	必选	描述
singer_name	String	是	歌手名。
album_id	String	是	歌曲所属专辑 id。
album_name	String	是	歌曲所属专辑名。
song_id	String	是	歌曲 id。
song_name	String	是	歌曲名。
singer_id	String	是	歌手 id。
playable_code	Integer	是	是否可以播放。枚举值依次对应: <ul style="list-style-type: none"><li>0: 可以。</li><li>1: 海外地区不能播放。</li><li>2: 歌曲无版权不能播放。</li><li>3: 会员歌曲, 非会员不能播放。</li><li>4: 付费内容, 须购买才可播放。</li><li>5: 牛方案策略, 非会员不能播放。</li><li>6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。</li><li>9: 未知原因, 无权播放。</li></ul>
has_accompany	Integer	是	是否有伴奏。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 是。</li></ul>

singer_im g	String	是	歌手头像。
album_im g_mini	String	是	专辑封面100px左右。
language	String	否	语言。
is_vip_so ng	Integer	是	是否 vip 歌曲。 • 0: 否。 • 1: 是。
album_im g	String	是	专辑封面。
album_im g_small	String	是	专辑封面300px左右。
album_im g_medium	String	是	专辑封面500px左右。
mv_id	String	是	mv id

## songs\_search\_voice

语音搜索歌曲列表

被如下接口引用: `search_voice`

名称	类型	必选	描述
singer_na me	String	否	歌手名。
duration	Integer	是	歌曲时长, 单位 ms。
song_id	String	是	歌曲 id。
song_nam e	String	是	歌曲名。

## songs\_singer\_song

歌曲列表

被如下接口引用: `singer_song`

名称	类型	描述

singer_img	String	歌手头像。
album_img	String	专辑封面。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。
album_img_medium	String	专辑封面500px左右。
is_vip_son g	Integer	是否 vip 歌曲。 • 0: 否。 • 1: 是。
mv_id	String	mv id
has_acco mpany	Integer	是否有伴奏。 • 0: 否。 • 1: 是。
song_id	String	歌曲 id。
song_nam e	String	歌曲名。
singer_id	String	歌手 id。
singer_na me	String	歌手名。
album_id	String	专辑 id。
album_na me	String	专辑名。
playable_c ode	Integer	是否可以播放。枚举值依次对应： • 0: 可以。 • 1: 海外地区不能播放。 • 2: 歌曲无版权不能播放。 • 3: 会员歌曲, 非会员不能播放。 • 4: 付费内容, 须购买才可播放。 • 5: 牛方案策略, 非会员不能播放。

		<ul style="list-style-type: none"><li>6: 固定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。</li><li>9: 未知原因, 无权播放。</li></ul>
try_playable	Integer	是否有试听片段。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 有。</li></ul>
language	String	语言。

## songs\_song\_infos

### 歌曲列表

被如下接口引用: song\_infos

名称	类型	描述
singer_img	String	歌手头像。
album_img	String	专辑封面。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。
album_img_medium	String	专辑封面500px左右。
is_vip_song	Integer	是否 vip 歌曲。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 是。</li></ul>
mv_id	String	mv id
has_accompany	Integer	是否有伴奏。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 是。</li></ul>
song_id	String	歌曲 id。
song_name	String	歌曲名。
singer_id	String	歌手 id。

singer_name	String	歌手名。
album_id	String	专辑 id。
album_name	String	专辑名。
playable_code	Integer	<p>是否可以播放。枚举值依次对应：</p> <ul style="list-style-type: none"> <li>0: 可以。</li> <li>1: 海外地区不能播放。</li> <li>2: 歌曲无版权不能播放。</li> <li>3: 会员歌曲, 非会员不能播放。</li> <li>4: 付费内容, 须购买才可播放。</li> <li>5: 牛方案策略, 非会员不能播放。</li> <li>6: 固定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。</li> <li>9: 未知原因, 无权播放。</li> </ul>
tryPlayable	Integer	<p>是否有试听片段。</p> <ul style="list-style-type: none"> <li>0: 否。</li> <li>1: 有。</li> </ul>
topic_url	String	歌曲专题页面(购买链接)。
language	String	语言。

## songs\_song\_tolisten

### 歌曲列表

被如下接口引用: song\_tolisten

名称	类型	描述
singer_img	String	歌手头像。
album_img	String	专辑封面。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。

album_img_medium	String	专辑封面500px左右。
is_vip_song	Integer	是否 vip 歌曲。0: 否。1: 是。
mv_id	String	mv id
has_accompany	Integer	是否有伴奏。0: 否。1: 是。
song_id	String	歌曲 id。
song_name	String	歌曲名。
singer_id	String	歌手 id。
singer_name	String	歌手名。
album_id	String	专辑 id。
album_name	String	专辑名。
playable_code	Integer	<p>是否可以播放。枚举值依次对应：</p> <ul style="list-style-type: none"> <li>0: 可以。</li> <li>1: 海外地区不能播放。</li> <li>2: 歌曲无版权不能播放。</li> <li>3: 会员歌曲，非会员不能播放。</li> <li>4: 付费内容，须购买才可播放。</li> <li>5: 牛方案策略，非会员不能播放。</li> </ul> <p>6: 因定向版权下架不能播放（针对 APP 有权但设备端无权的情况）。</p> <p>9: 未知原因，无权播放。</p>
tryPlayable	Integer	<p>是否有试听片段。</p> <ul style="list-style-type: none"> <li>0: 否。</li> <li>1: 有。</li> </ul>
topic_url	String	歌曲专题页面(购买链接)。
language	String	语言。

## songs\_top\_song

歌曲列表

被如下接口引用：top\_song

名称	类型	描述
singer_name	String	歌手名。
album_id	String	歌曲所属专辑 id。
album_name	String	歌曲所属专辑名。
song_id	String	歌曲 id。
song_name	String	歌曲名。
singer_id	String	歌手 id。
playable_code	Integer	是否可以播放。枚举值依次对应： <ul style="list-style-type: none"><li>0: 可以。</li><li>1: 海外地区不能播放。</li><li>2: 歌曲无版权不能播放。</li><li>3: 会员歌曲, 非会员不能播放。</li><li>4: 付费内容, 须购买才可播放。</li><li>5: 牛方案策略, 非会员不能播放。</li><li>6: 因定向版权下架不能播放 (针对 APP 有权但设备端无权的情况)。</li><li>9: 未知原因, 无权播放。</li></ul>
is_vip_son	Integer	是否 vip 歌曲。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 是。</li></ul>
mv_id	String	mv id
has_accom	Integer	是否有伴奏。 <ul style="list-style-type: none"><li>0: 否。</li><li>1: 是。</li></ul>
singer_img	String	歌手头像。
album_img_mini	String	专辑封面100px左右。
album_img_small	String	专辑封面300px左右。

language	String	语言。
album_img	String	专辑封面。
album_img_medium	String	专辑封面500px左右。

## tags\_mv\_info

mv 信息

被如下接口引用：mv\_info

名称	类型	必选	描述
tag_id	String	否	标签 id。
parent_id	String	否	标签 parentId。
tag_name	String	否	标签的名称。

## tags\_mv\_infos

mv 信息

被如下接口引用：mv\_infos

名称	类型	必选	描述
tag_id	String	否	标签 id。
parent_id	String	否	标签 parentId。
tag_name	String	否	标签的名称。

## themes\_accompany\_theme\_list

主题列表

被如下接口引用：accompany\_theme\_list

名称	类型	必选	描述
theme_thumbnail	String	是	主题缩略图。
theme_id	String	是	主题 id。

theme_name	String	是	主题名。
theme_img	String	是	主题图片。

## tokenapiCommon

tokenApi的公共参数

名称	类型	必选	描述
Action	String	是	公共参数, 本接口取值: AppLogoutUser。
AccessToken	String	是	公共参数, AccessToken 用于对一个已经登录的用户鉴权。
RequestId	String	是	公共参数, 唯一请求 ID, 可自行生成, 推荐使用 uuld。定位问题时, 需提供该次请求的 RequestId。

## tops\_accompany\_free\_top\_list

榜单列表

被如下接口引用: accompany\_free\_top\_list

名称	类型	必选	描述
top_id	String	是	榜单 id。
top_name	String	是	榜单名。
header_url	String	是	榜单图片。

## tops\_accompany\_top\_list

tops

被如下接口引用: accompany\_top\_list

名称	类型	必选	描述
top_id	String	是	榜单 id。
top_name	String	是	榜单名。

## tops\_top\_list

## 榜单

被如下接口引用: `top_list`

名称	类型	描述
<code>top_id</code>	<code>String</code>	榜单 id。
<code>top_name</code>	<code>String</code>	榜单名。
<code>header_url</code>	<code>String</code>	榜单图片。

## `vertical_live_room_url`

### 竖屏流信息

被如下接口引用: `live_room_url`

名称	类型	必选	描述
<code>rtmp</code>	<code>Array Of String</code>	否	<code>rtmp</code> 流信息。
<code>httpshls</code>	<code>Array Of String</code>	否	<code>httpshls</code> 流信息。
<code>streamName</code>	<code>String</code>	否	流名称。
<code>rate</code>	<code>String</code>	否	清晰度标识 (1: 流畅。2: 标清。3: 高清。4: 超清。5: 1080P。)
<code>hls</code>	<code>Array Of String</code>	否	<code>hls</code> 流信息。
<code>httpflv</code>	<code>Array Of String</code>	否	<code>httpflv</code> 流信息。
<code>httpsflv</code>	<code>Array Of String</code>	否	<code>httpsflv</code> 流信息。

## `WechatTemplate`

### 第三方微信模板

被如下接口引用: `AppDescribeTemplateBinding`

名称	类型	必选	描述

ModelId	String	否	产品型号ID。
TemplatelId	String	否	微信模板ID
Title	String	否	微信模板标题
Status	Int	否	订阅状态: 0未订阅; 1已订阅; 2不接收

## WechatDeviceInfo

微信设备票据

被如下接口引用: APPGetWechatDeviceTicket

名称	类型	必选	描述
DeviceId	String	否	设备ID。
WXIoTDeviceInfo.SN	String	否	设备SN。
WXIoTDeviceInfo.SNTicket	String	否	设备票据。
WXIoTDeviceInfo.ModelId	String	否	产品型号ID。

## DeviceVersions

设备版本信息

被如下接口引用: AppGetFamilyList

名称	类型	描述
Version	String	版本号
Module	String	模块类型