**快传技术**

PC版本SDK接口使用文档

|  |  |  |
| --- | --- | --- |
| 文件状态：  [√] 草稿  [ ] 正式修改  [ ] 正在发布 | 文件标识： | 快传SDK接口使用文档 |
| 当前版本： | PC版本1.0.0 |
| 作 者： | 周王钦 |
| 完成日期： | 2017/9/10 |

修订历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 序号 | 版本 | 编写/修订说明 | 修订人 | 修订日期 | 备注 |
| 1 | 1.0.0 | 新建文档 | 周王钦 | 2017/9/10 |  |
| 2 |  |  |  |  |  |

目 录

[1、概述 4](#_Toc493584828)

[2、SDK使用说明 4](#_Toc493584829)

[1 初始化SDK 4](#_Toc493584830)

[2 反初始化SDK 4](#_Toc493584831)

[3 添加事件回调 5](#_Toc493584832)

[4 移除事件回调 5](#_Toc493584833)

[5 注册flyCan 6](#_Toc493584834)

[6 注销flyCan 6](#_Toc493584835)

[7 创建指定会话 6](#_Toc493584836)

[8 监听会话 7](#_Toc493584837)

[9 连接会话 7](#_Toc493584838)

[10 接受会话 7](#_Toc493584839)

[11 拒绝会话 8](#_Toc493584840)

[12 释放会话 8](#_Toc493584841)

[13 发送数据 8](#_Toc493584842)

[14 接收数据 9](#_Toc493584843)

[3.SDK使用流程 9](#_Toc493584844)

[3.1客户端使用流程 9](#_Toc493584845)

[3.2服务端使用流程 9](#_Toc493584846)

[附录1 10](#_Toc493584847)

[获取Sid、Token、appId 10](#_Toc493584848)

# 1、概述

SDK提供用户基本的接口，通过接口调用，可以获得CAN的媒体传输能力。

# 2、SDK使用说明

## 1 初始化SDK

void FlyCanInit(char \*sid, char \*token, char \*appid, char \*sdkid);

功能：SDK初始化

输入参数:

sid：用户主账号ID号，通过flypaas官网注册获得

token：用户主账号秘钥，通过flypaas官网注册获得

Appid: 在flypaas用户开放平台创建产生一个APPid

sdkid: 调用flypaas REST API接口获得，或由flyCAN统一分配的ID号

返回参数:

无

**备注:**

关于账号密钥和sdkid的获取，请参考附录1

## 2 反初始化SDK

void FlyCanUnInit();

功能：SDK反初始化,释放资源

输入参数:

无

返回参数:

无

## 3 添加事件回调

void FlyCanAddCallBack(CallBackFunc cb);

功能：与会话相关的事件触发，通过回调函数cb返回

输入参数:

cb: 回调函数结构体,原型如下:

// 事件回调

typedef void(\*CallBackEvent)(int nEvent, int nCode, void \*pData);

// 接收数据回调

typedef void(\*CallBackRecv)(void\* pHandle, void \*pData, int nLen, int nChannalId);

// 回调函数集合

typedef struct

{

CallBackEvent event\_cb;

CallBackRecv recv\_cb;

} CallBackCB,\*CallBackFunc;

返回参数:

无

CallBackEvent中对应的nEvent类型

#define EVT\_REGISTER\_SUCCUSS 1000 // SDK注册成功

#define EVT\_REGISTER\_FAILURE 1001 // SDK注册失败

#define EVT\_LOGINOUT\_SUCCUSS 1002 // SDK注销成功

#define EVT\_SESSION\_BIND\_SUCCUSS 1003 // Session绑定成功

#define EVT\_SESSION\_BIND\_FAILURE 1004 // Session绑定失败

#define EVT\_SESSION\_LISTEN\_SUCCUSS 1005 // Session监听成功

#define EVT\_SESSION\_LISTEN\_FAILURE 1006 // Session监听失败

#define EVT\_SESSION\_RELEASE\_SUCCUSS 1007 // Session释放成功

#define EVT\_SESSION\_RELEASE\_FAILURE 1008 // Session释放失败

#define EVT\_SESSION\_INCOMING 1009 // 会话连接请求

#define EVT\_SESSION\_ACCEPT 1010 // 会话连接确认

#define EVT\_SESSION\_REJECT 1011 // 会话连接拒绝

## 4 移除事件回调

void FlyCanRemoveCallBack(CallBackFunc cb);

功能：移除事件回调

输入参数：

cb:事件回调函数

返回参数:

无

## 5 注册flyCan

void FlyCanRegister();

功能: 向flyCan注册，结果通过回调异步返回

输入参数:

无

返回参数:

无,成功失败通过回调消息返回

EVT\_REGISTER\_SUCCUSS 成功

EVT\_REGISTER\_FAILURE 失败

## 6 注销flyCan

void FlyCanUnRegister();

功能: SDK发送注销请求,断开和flyCan的连接

输入参数:

无

返回参数:

无,通过消息返回结果

EVT\_LOGINOUT\_SUCCUSS 成功

## 7 创建指定会话

void\* FlyCanCreateSession(int protocol);

功能: 创建一个指定协议会话,返回会话句柄

输入参数：

Protocol：会话协议类型

#define PROTOCOL\_UDP 0

#define PROTOCOL\_RTP 1 (目前暂时还不支持)

输出参数：

会话的句柄handle,成功失败通过事件通知

EVT\_SESSION\_BIND\_SUCCUSS 成功

EVT\_SESSION\_BIND\_FAILURE 失败

## 8 监听会话

void FlyCanListenSession(void \*handle, int nNum);

功能: 服务器端调用，通知flyCan会话可以接收连接请求

输入参数：

FlyCanCreateSession返回的会话句柄handle

nNum:可以监听的客户端数据

输出参数：

无, 成功失败通过事件通知

EVT\_SESSION\_LISTEN\_SUCCUSS 成功

EVT\_SESSION\_LISTEN\_FAILURE 失败

## 9 连接会话

bool FlyCanConnectSession(void \*handle, char \*peersdkid);

功能: 连接会话,通知flyCan连接目的地sdkid

输入参数：

Handle: FlyCanCreateSession返回的会话的句柄handle

Peersdkid: 目的地sdkid

输出参数：

True 发送成功, false 发送失败

**备注：**

目的地方会接收到EVT\_SESSION\_INCOMING连接消息，参数附带了最大可用的通道数

目的地方可以通过FlyCanAcceptSession和flycan\_RejectSession接口来响应此连接请求

接收事件通知EVT\_SESSION\_ACCEPT说明对方同意连接，可以通信

接收事件通知EVT\_SESSION\_REJECT说明对方不同意连接，不可以通信

## 10 接受会话

bool FlyCanAcceptSession(void \*handle);

功能: 接受会话, 目的地接收到EVT\_SESSION\_INCOMING连接请求，调用此接口接收请求

输入参数：

Handle: 接收到连接请求消息 EVT\_SESSION\_INCOMING返回的会话的句柄handle

输出参数：

True 发送成功, false 发送失败

主叫方会接收到EVT\_SESSION\_ACCEPT消息, 参数附带了最大可用的通道数

## 11 拒绝会话

bool FlyCanRejectSession(void \*handle);

功能: 拒绝会话, 目的地收到EVT\_SESSION\_INCOMING连接请求，调用此接口拒绝请求

输入参数：

Handle: 接收到连接请求消息 EVT\_SESSION\_INCOMING返回的会话的句柄handle

输出参数：

True 发送成功, false 发送失败

主叫方会接收到EVT\_SESSION\_REJECT消息

## 12 释放会话

void FlyCanReleaseSession(void \*handle);

功能: 释放会话

输入参数：

Handle:会话句柄

输出参数：

无, 成功失败通过事件通知

EVT\_SESSION\_RELEASE\_SUCCUSS 成功

EVT\_SESSION\_RELEASE\_FAILURE 失败

## 13 发送数据

bool FlyCanSend(void \*handle, void \*pData, int nLen, int nChannalId);

功能: 发送数据

输入参数：

Handle: 会话句柄

pData: 数据包

nLen: 数据包长度

nChannalId: 通道号

输出参数：

True 发送成功, false 发送失败

## 14 接收数据

typedef void(\*CallBackRecv)(void\* pHandle, void \*pData, int nLen, int nChannalId);

通过回调函数CallBackRecv 来获取数据

参数:

pHandle: 对应会话的句柄

pData : 数据包

nLen : 数据长度

nChannalId : 数据通道

# 3.SDK使用流程

## 3.1客户端使用流程

1. 调用FlyCanInit初始化SDK

2. 调用FlyCanAddCallBack增加回调函数

3. 调用FlyCanRegister向flyCan注册

4. 上述步骤成功后调用FlyCanCreateSession创建一个会话

5. 调用FlyCanListenSession监听创建的会话（也可以不调用）

6. 调用FlyCanConnectSession发起会话连接

7. 接收到被叫EVT\_SESSION\_ACCEPT消息，握手成功，可以通信

8. 调用FlyCanSend进行发送数据,接收数据通过回调返回

9. 接收到被叫EVT\_SESSION\_REJECT消息，握手不成功，连接断开

10. 通信完成后，调用FlyCanReleaseSession关闭创建的会话

11. 调用FlyCanUnRegister注销释放资源

12． 调用FlyCanUnInit()去初始化

## 3.2服务端使用流程

1. 调用FlyCanInit初始化SDK

2. 调用FlyCanAddCallBack增加回调函数

3. 调用FlyCanRegister向flyCan注册

4. 上述步骤成功后调用FlyCanCreateSession创建一个会话

5. 调用FlyCanListenSession监听创建的会话

6. 接收到EVT\_SESSION\_INCOMING消息，等待响应

7. 调用FlyCanAcceptSession接收连接请求,握手成功，可以通信

8. 调用FlyCanSend进行发送数据,接收数据通过回调返回

9. 调用FlyCanRejectSession拒绝连接请求,握手不成功，连接断开

10. 通信完成后，调用FlyCanReleaseSession关闭创建的会话

11. 调用FlyCanUnRegister注销释放资源

12． 调用FlyCanUnInit()去初始化

# 附录1

## 获取Sid、Token、appId

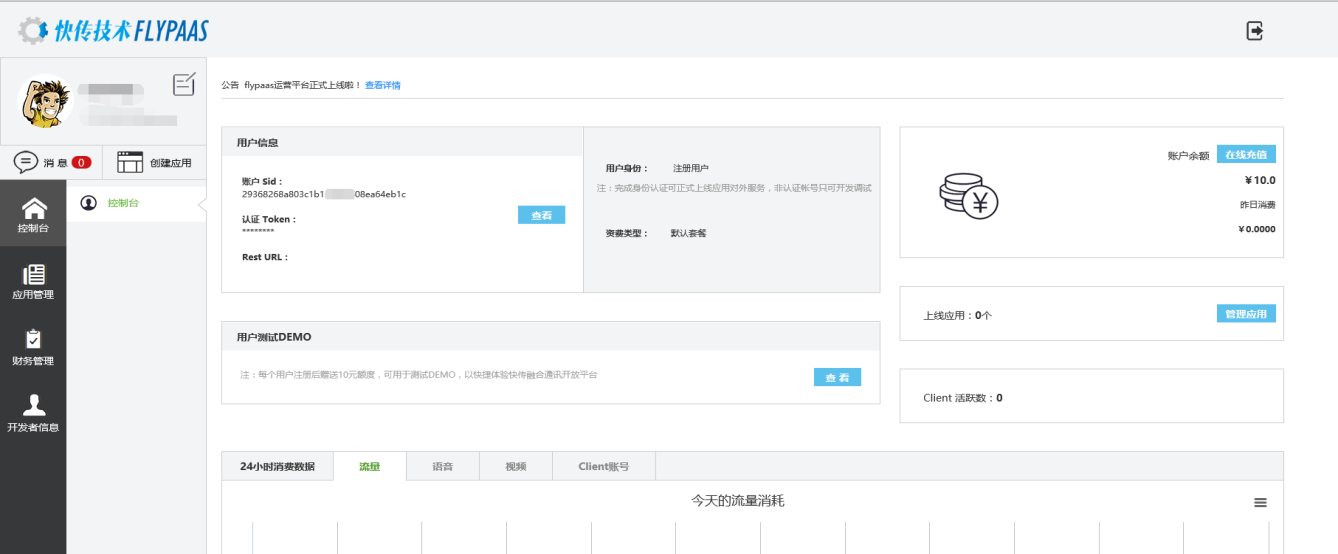
I **登录[www.flypaas.com](http://www.flypaas.com)**



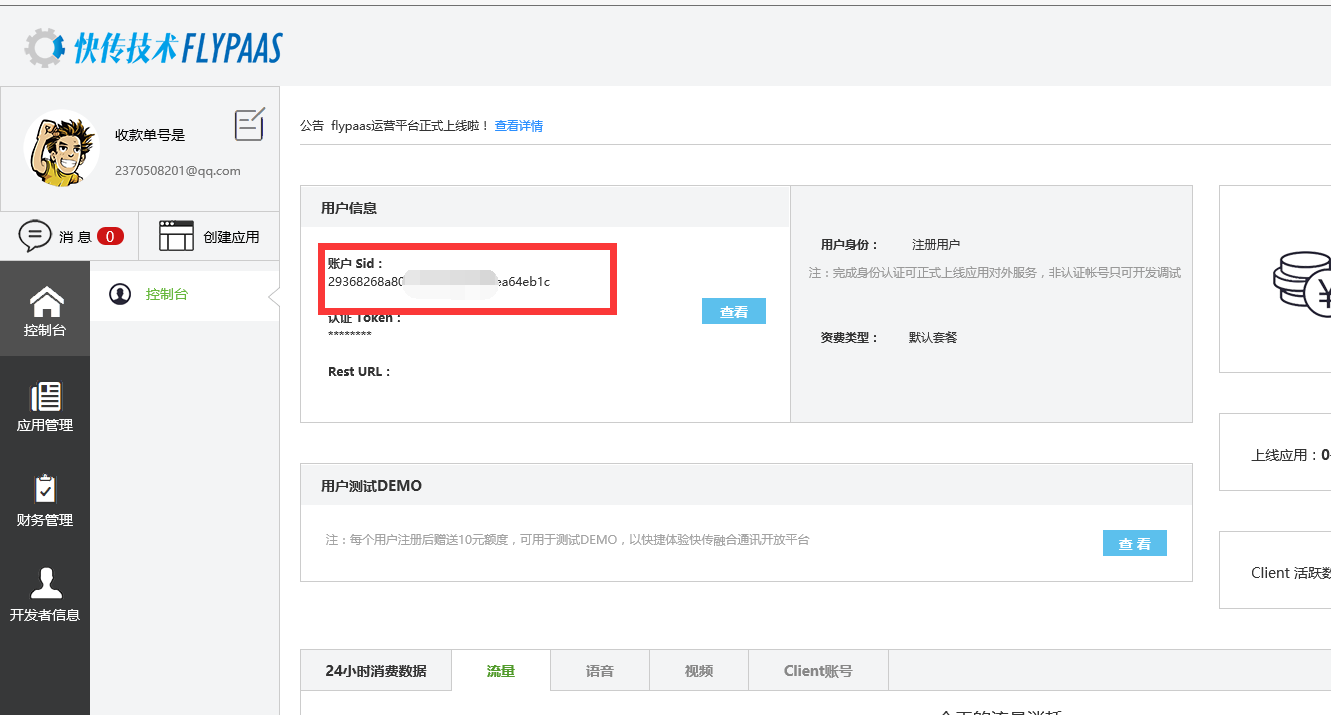
**II 页面右上角选择登录进入登录页面**



**III 输入邮箱账号、密码进行登录并进入用户平台**



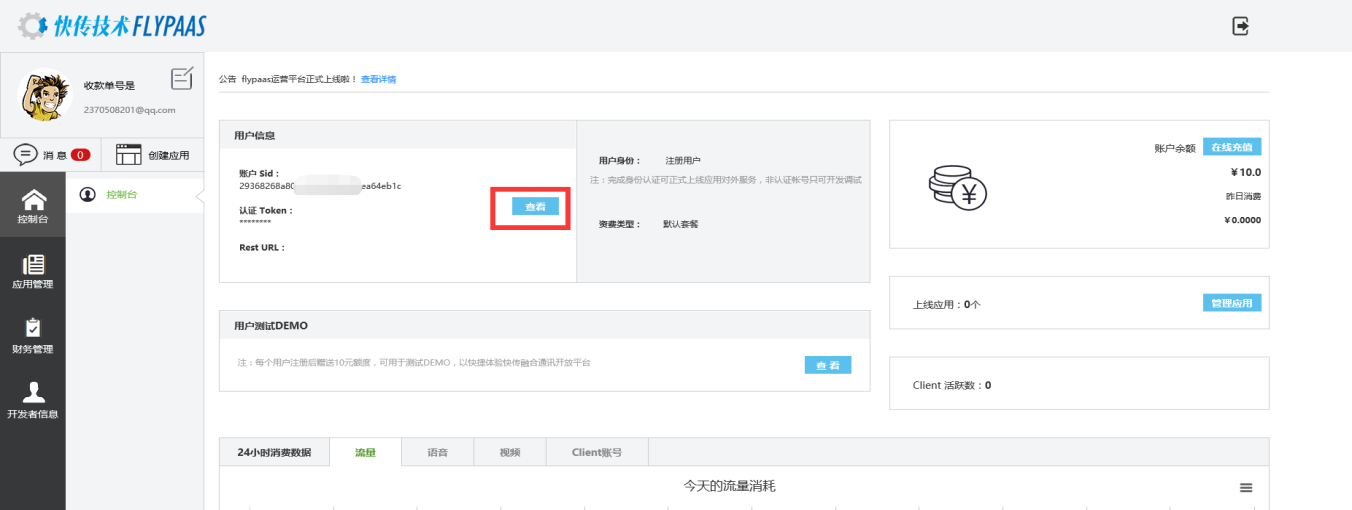
**IV 获取Sid**

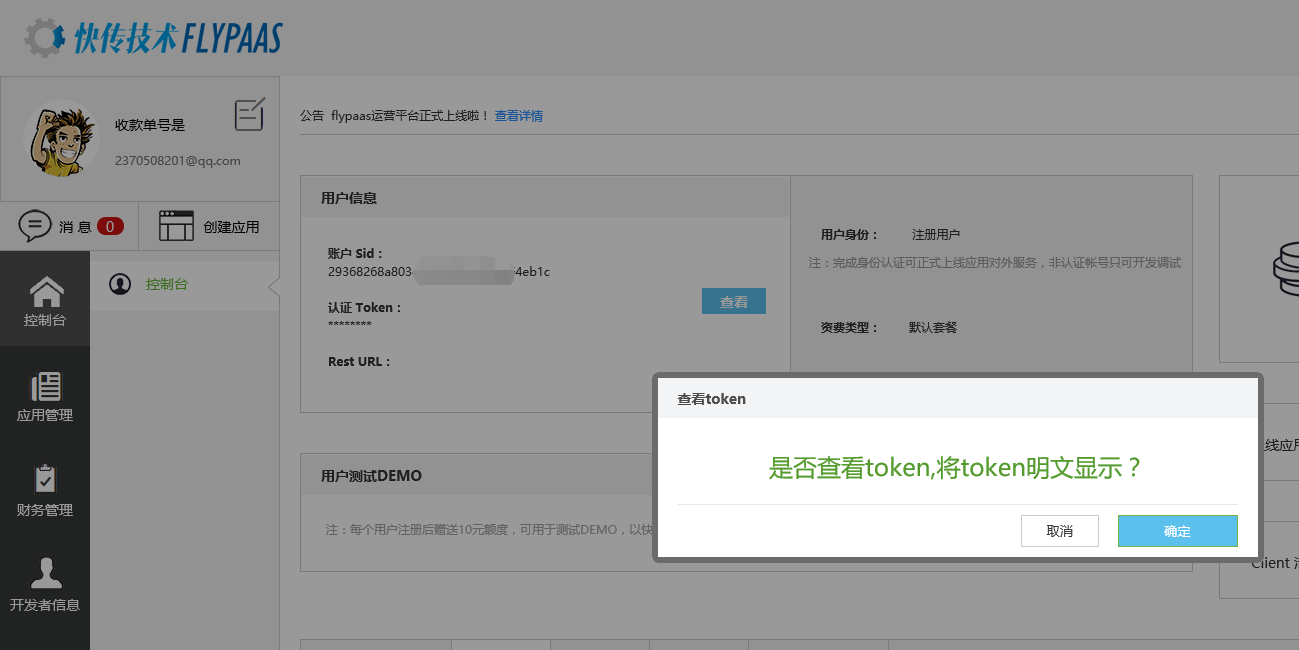


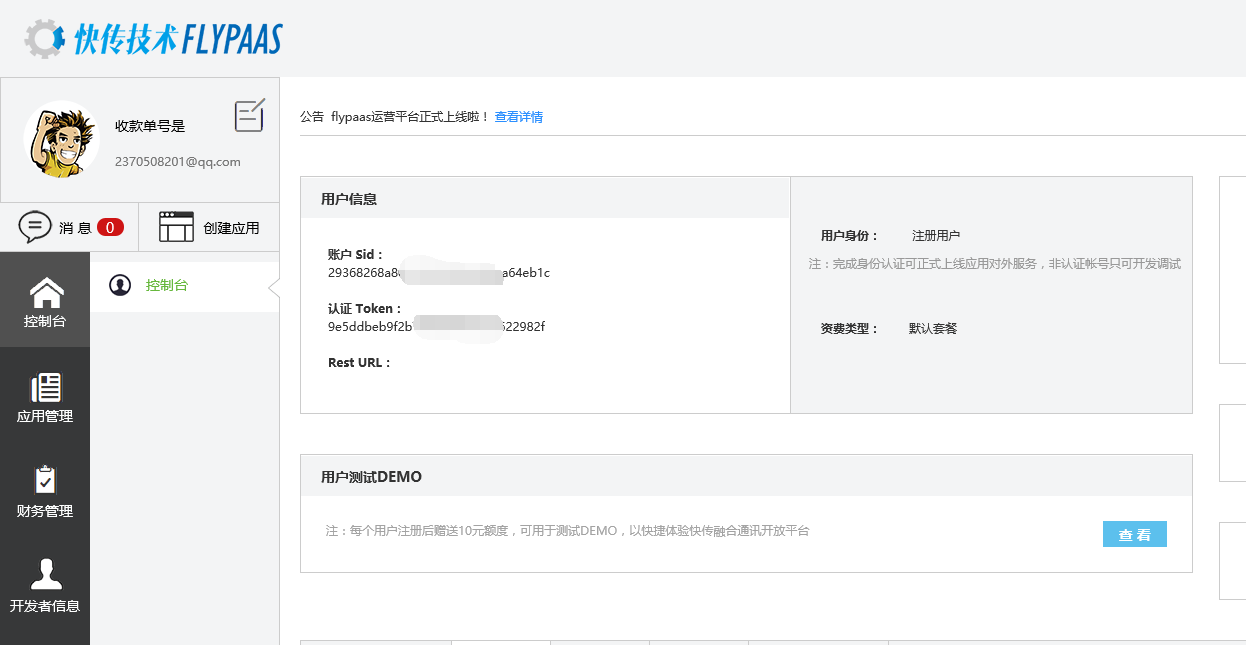
用户首页红色方框部分即为用户Sid

**V 获取Token**

用户首页中，点击红色方框部分，即在其左侧显示Token







**VI 获取appId**

1. 点击 “应用管理”->“应用列表”进入应用列表页面



1. 点击应用对应的“管理”按钮，进入应用信息界面



1. 红色方框部分即为应用所对应的appSid



注意：只有通过了账号认证的用户才能创建应用，只有应用申请上线成功才能使用appSid，

**VII 获取sdkId**

通过HTTPSPOST方式提交请求，flyCAN开放平台收到请求后，为app创建SDKID，返回SDKID的基本信息。

（1）请求

**•请求地址**

/{SoftVersion}/Accounts/{accountSid}/applySDKID

**•请求包头**

详情请查阅统一请求包头，并使用主账号进行验证。

**•请求包体**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 约束 | 说明 |
| appId | String | 必选 | 应用Id |

**•XML请求示例**

POST/2017-03-28/Accounts/fb2a7e7f036ea668bbb657e00bfb6ab8/applySDKID?sig=59B06091736DDBD16F7F727C04CB4542

Accept:application/xml

Content-Type:application/xml;charset=utf-8

Authorization:ZTAzYmM5MTA2YzZlZDBlYWViZmNlOGMzNjhmZGNkNDg6MjAxNDA2MjMxNzE2MzI=

<?xmlversion="1.0"encoding="UTF-8"standalone="yes"?>

<SDK>

<appId>e462aba25bc6498fa5ada7eefe1401b7</appId>

</SDK>

**•JSON请求示例**

POST/2017-03-28/Accounts/fb2a7e7f036ea668bbb657e00bfb6ab8/applySDKID?sig=59B06091736DDBD16F7F727C04CB4542

Accept:application/json

Content-Type:application/json;charset=utf-8

Authorization:ZTAzYmM5MTA2YzZlZDBlYWViZmNlOGMzNjhmZGNkNDg6MjAxNDA2MjMxNzMwMzg=

{

"SDK":{

"appId":"cac7d7ae1d3040b886b2766d69ce11b3"

}

}

（2）响应

**•响应包体**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 约束 | 说明 |
| respCode | String |  | 请求状态码，取值000000（成功） |
| SDKID | String |  | 申请获得的SDKID |
| appId | String |  | SDKID所属应用ID |
| createDate | String |  | SDKID创建日期 |

**•XML响应示例**

<?xmlversion="1.0"encoding="UTF-8"standalone="yes"?>

<resp>

<respCode>000000</respCode>

<SDK>

<SDKID>62395051197520</SDKID>

<appId>cac7d7ae1d3040b886b2766d69ce11b3</appId>

<createDate>2017-08-2814:39:51</createDate>

</SDK>

</resp>

**•JSON响应示例**

{

"resp":{

"respCode":"000000",

"SDK":{

"appId":"547eab8def2d45d0ab5c7c710f6b9893",

"createDate":"2017-08-2814:39:51",

"SDKID":"62395051197520"

}

}

}

**VIII 删除sdkId**

通过HTTPSPOST方式提交请求，flyCAN开放平台收到请求后，删除应用所对应的SDKID后，返回结果信息。

（1）请求

**•请求地址**

/{SoftVersion}/Accounts/{accountSid}/dropSDKID

**•请求包头**

详情请查阅统一请求包头，并使用主账号进行验证。

**•请求包体**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 约束 | 说明 |
| appId | String | 必选 | 应用Id |
| SDKID | String | 必选 | 应用对应的SDKID |

**•XML请求示例**

POST/2017-03-28/Accounts/fb2a7e7f036ea668bbb657e00bfb6ab8/dropSDKID?sig=59B06091736DDBD16F7F727C04CB4542

Accept:application/xml

Content-Type:application/xml;charset=utf-8

Authorization:ZTAzYmM5MTA2YzZlZDBlYWViZmNlOGMzNjhmZGNkNDg6MjAxNDA2MjMxNzE2MzI=

<?xmlversion="1.0"encoding="UTF-8"standalone="yes"?>

<SDK>

<appId>e462aba25bc6498fa5ada7eefe1401b7</appId>

<SDKID>62395051197520</SDKID>

</SDK>

**•JSON请求示例**

POST/2017-03-28/Accounts/fb2a7e7f036ea668bbb657e00bfb6ab8/dropSDKID?sig=59B06091736DDBD16F7F727C04CB4542

Accept:application/json

Content-Type:application/json;charset=utf-8

Authorization:ZTAzYmM5MTA2YzZlZDBlYWViZmNlOGMzNjhmZGNkNDg6MjAxNDA2MjMxNzMwMzg=

{

"SDK":{

"appId":"547eab8def2d45d0ab5c7c710f6b9893",

"SDKID":"62395051197521"

}

}

（2）响应

**•响应包体**

|  |  |  |  |
| --- | --- | --- | --- |
| 属性 | 类型 | 约束 | 说明 |
| respCode | String | 必选 | 请求状态码，取值000000（成功） |

**•XML响应示例**

<?xmlversion="1.0"encoding="UTF-8"standalone="yes"?>

<resp>

<respCode>000000</respCode>

</resp>

**•JSON响应示例**

{

"resp":{

"respCode":"000000"

}

}