**ZYIOT Open Account SDK**

**使用说明**

版本：V2.0 日期：2018-12-27

2018

**专注云技术、共筑云安全**

**关于本文档**

ZYIOT Open Account SDK的使用说明。

**发布说明**

|  |  |  |
| --- | --- | --- |
| **日期** | **版本** | **发布说明** |
| 2018-12-27 | V2.0.0 | 1. 新增网关子设备功能 2. 新增Open Acount支持 |
| 2017-06-22 | V1.0.0 | 首次发布 |

**文档变更通知**

用户可咨询技术支持人员或通过邮箱：[info@zhuyun-it.com](mailto:info@zhuyun-it.com) 获取最新文档。

**目录**

[1 接入必读 3](#_Toc19478)

[2 简介 4](#_Toc15016)

[3 名词解释 5](#_Toc19505)

[4 功能介绍 7](#_Toc644)

[5 配置SDK 8](#_Toc22197)

[5.1 环境准备 8](#_Toc3287)

[5.2 创建应用 8](#_Toc18876)

[5.3 SDK安装 8](#_Toc20191)

[5.4 配置工程 9](#_Toc16118)

[6 主要流程 11](#_Toc5984)

[6.1 初始化SDK 11](#_Toc24195)

[6.2 添加设备、分享设备、编辑被分享权限、删除设备 13](#_Toc30889)

[6.3 获取用户下的设备列表 14](#_Toc31097)

[6.4 获取指定设备的所有属性 15](#_Toc28525)

[6.5 控制设备 15](#_Toc12756)

[6.6 设备固件升级 17](#_Toc5514)

[6.7 关闭IOT Client 18](#_Toc25242)

# 接入必读

Android的demo已经包含了开放平台SDK的jar文件，请将SDK的jar文件拷贝到项目工程的libs文件夹中。

# 简介

本文档用于说明筑云开放平台SDK Android版本接口之间的关系以及接口调用顺序，对开放平台SDK Android版本主要流程都有详细说明和代码示例。主要有功能介绍、安装说明、权限配置和主要流程介绍。

# 名词解释

|  |  |
| --- | --- |
| 名词 | 注解 |
| appId | 筑云提供的对应APPID |
| appSecret | 与APPID对应的secret |
| tenantId | 筑云提供的一个标识ID |
| appToken | 第三方用户自有用户系统鉴权的token |
| keyhash | 设备唯一标识（序列号） |
| devTypeId | 设备类型ID |
| ZOT | 用户与设备相关功能的业务处理模块（处理除FOTA与IOT之外的业务）。SDK接口分为三大模块，分别是ZOT、IOT、FOTA。 |
| IOT Client | SDK中专门用来与设备进行通信的客户端 |
| Event | IOT Client与设备进行通信的消息封装（比如包含属性名、属性值），详见API文档。 |
| FOTA | 设备固件升级模块 |
| 设备属性 | 根据设备类型由厂家规定，包含属性名、属性值类型、属性值。（属性规定，详见厂家设备属性定义文档）。 |
| 属性值类型 | 表示Event封装的属性值的类型，1是string；2是int；4是hex。 |
| hexTypeSize | 通过Event发送语音文件时，分片发送时单片内容的最大数据长度。 |
| endpointType | 端点类型，分为：0：wifi; 1：app; 2：app下挂设备(app直连子设备); 3：网关下挂设备 |
| parentKeyhash | 父设备keyhash，针对app下挂设备和网关下挂设备，为这两种类型向上一级的设备keyhash |
| authority | 用户对于设备的权限（管理员无此字段），JSON格式内容。 |

表 3-1

# 功能介绍

|  |  |
| --- | --- |
| 功能 | 说明 |
| 用户管理 | 支持第三方用户登录，支持查看设备下的用户列表，查看用户下的设备列表 |
| 临时用户管理 | 支持将设备分享给临时用户，可增加临时用户、修改临时用户权限、删除临时用户（临时用户使用设备可以不需要自己的账号，需要被分享时得到的tempUserId） |
| 设备管理 | 支持绑定设备、删除设备、控制设备，还可将设备授权分享给其他用户、可编辑被分享用户的权限、可删除被分享用户 |
| 设备记录 | 获取设备历史记录（设备异常上下线、操控记录，以及授权分享记录） |
| 联动与定时 | 支持设备与设备的联动；支持设备定时执行动作 |
| 触发器 | 支持为自己的设备增加自定义触发器、编辑和删除自定义触发器（减少触发情况，达到更精确触发，得到需要的消息） |

表4-1

# 配置SDK

* 1. 环境准备

支持 Android Studio 1.5 以上 支持 JDK 7.0 以上版本  
 支持 Android 手机系统 4.2 以上版本

* 1. 创建应用

联系筑云开发人员申请tenantID、appID、appSecret。

* 1. SDK安装

1. 解压缩ZYIOTSDK.zip；
2. 拷/libs下jar包到自己工程对应目录（包含下图中5个jar包）；

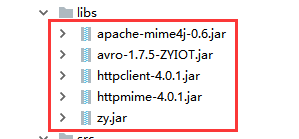


图5-1

1. 将以下依赖内容添加到build.gradle文件 的dependencies内：

implementation **'com.android.support:multidex:1.0.3'**implementation fileTree(**dir**: **'libs'**, **include**: [**'\*.jar'**])  
implementation **'org.lucee:commons-io:2.6.0'**implementation **'commons-codec:commons-codec:1.9'**implementation **'org.apache.commons:commons-compress:1.18'**implementation **'org.apache.httpcomponents:httpcore:4.4.10'**implementation **'org.glassfish:javax.annotation:10.0-b28'**implementation **'org.slf4j:log4j-over-slf4j:1.7.25'**implementation **'ch.qos.logback:logback-classic:1.1.2'**implementation **'ch.qos.logback:logback-core:1.1.2'**implementation **'org.slf4j:slf4j-api:1.7.25'**implementation **'org.codehaus.jackson:jackson-core-asl:1.9.13'**implementation **'org.codehaus.jackson:jackson-mapper-asl:1.9.12'**implementation **'org.bouncycastle:bcprov-jdk15on:1.60'**implementation **'io.netty:netty-all:4.1.6.Final'**

* 1. 配置工程

1. 配置 Android 权限

在 AndroidMainfest.xml 文件中添加：

<**uses-permission android:name="android.permission.INTERNET"** />  
<**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"** />  
<**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"** />  
<**uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE"** />

1. 配置build.gradle

defaultConfig {

……  
 multiDexEnabled **true** *//to deal with too much jar code*}

buildTypes {  
 release {

**……** lintOptions {*//mutil dex* checkReleaseBuilds **true** abortOnError **false** }  
 }  
 debug {

……lintOptions {*//mutil dex* checkReleaseBuilds **false** abortOnError **false** }  
 }  
}

packagingOptions {  
 exclude **'META-INF/DEPENDENCIES'** exclude **'META-INF/ASL2.0'**}

注意：因代码量过多，所以需要引入multiDex功能包，并为其配置build.gradle。

1. 配置 AndroidManifest.xml

添加如下配置，指定启动的Application，用于引入multiDex功能。其中yourSelfApplication替换为自己代码中定义的Application类——自定义Application一定要继承MultiDexApplication。

<**application  
 android:name="yourSelfApplication"  
 ……**

**>**

**</application>**

## 

# 主要流程

## 初始化SDK

初始化SDK时筑云后台需要通过第三方平台对用户进行鉴权，具体流程如下图7-1：

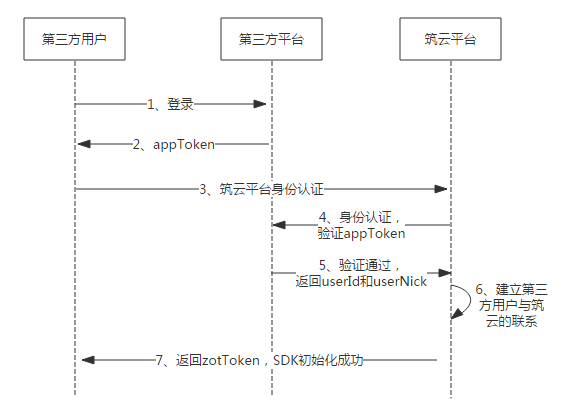


图7-1

1. APP使用第三方api向第三方后台登录。
2. APP登录成功，获取第三方token。
3. APP通过筑云的SDK接口向筑云后台进行身份认证，参数包括筑云提供的的appId，appSecret，以及第三方用户token。
4. 筑云后台通过第三方预留的http url进行身份认证，参数包括第三方的token和MD5（token+app\_secret）。
5. 第三方认证通过，返回userId和userNick(昵称）。
6. 筑云后台创建筑云的userId和第三方userId的对应关系，创建筑云的用户。
7. 返回zotToken，后续的筑云SDK的API访问使用该zotToken。
8. 向第三方平台验证token：

第三方平台在zot中预设url：假设为<https://www.anlin.auth>， 则筑云后台会

向该url以get方式发送<https://www.anlin.auth?appToken=xxx&sign=xxx>

的请求。appToken：app上传的第三方的token sign:MD5（appToken+appSecret），为32位的MD5返回结果为json格式：

{

"code": "200",//200是成功，其它是失败

"msg": "成功",//错误提示

"description": "",//错误详细描述

"data": {//结果数据

"userId":"xxxx", //必填

"userNick":"XXXX" //必填

}

}

代码示例，如图7-2：



图7-2 SDK 初始化

## 添加设备、分享设备、编辑被分享权限、删除设备

支持添加绑定设备、删除设备、分享设备、编辑设备用户权限等。

代码示例：

添加设备：



图7-3

删除设备：

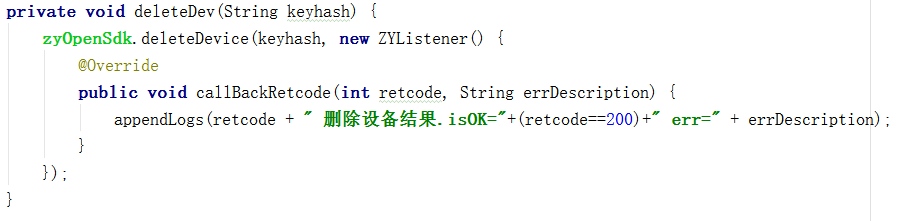


图7-4

分享设备、编辑分享的用户权限，以及删除被分享的用户：



图7-5-1

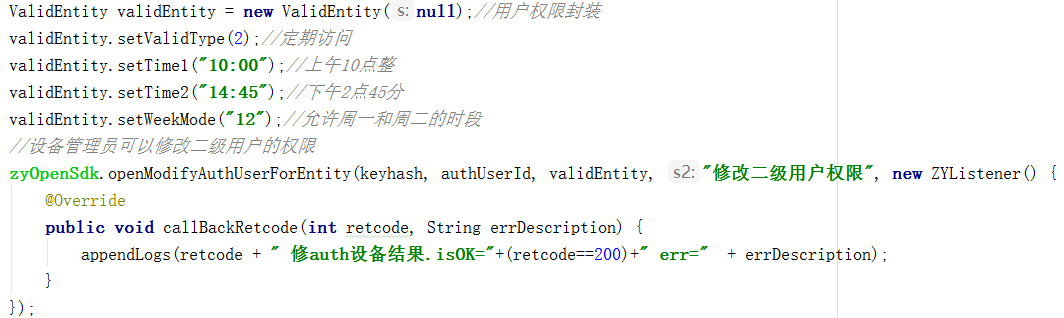


图7-5-2

## 获取用户下的设备列表

可以获取用户拥有的设备列表，代码示例：

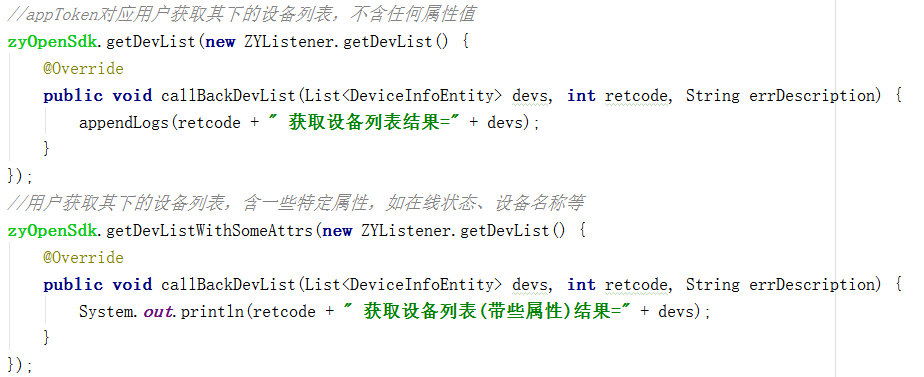


图7-6 获取设备列表

## 获取指定设备的所有属性

可以获取指定设备的所有属性值，代码示例：

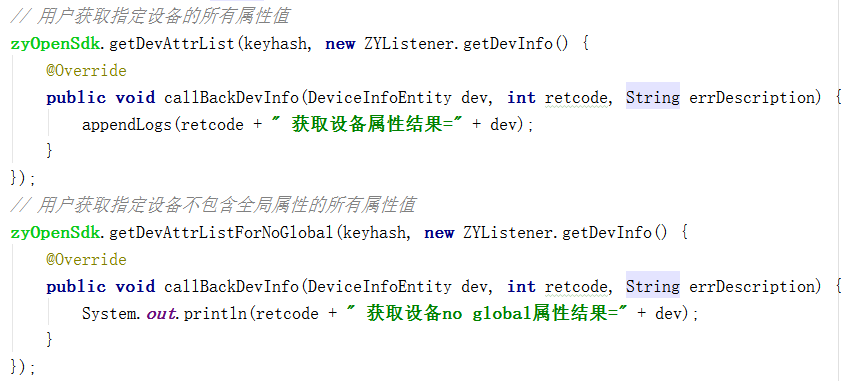


图7-7

## 控制设备

设置并实现指定Listener代理，可以得到不同的回调结果。如实现ZYEventResponseListener：发送Event后根据实现的onEventSyncResponseWithStatus回调接口查看发送结果，另外也可以通过实现onEventForAPP接口实时得到设备的状态变化，如设备上下线时的Event通知。代码示例如下：

设置回调代理并实现：

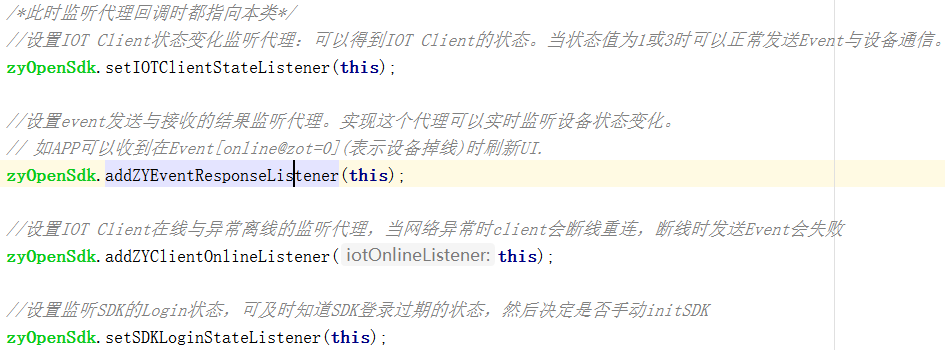


图7-8设置IOT相关代理

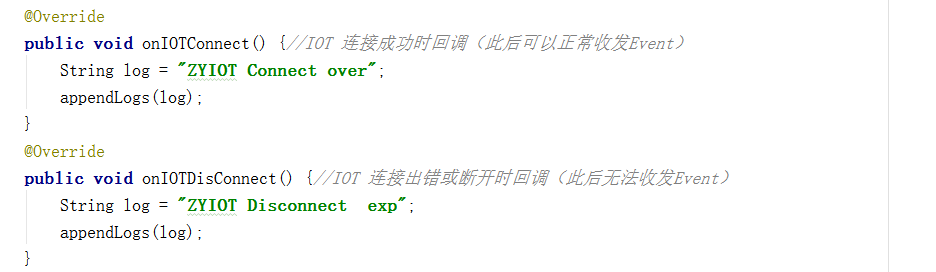


图7-9-1 IOTOnlineListener代理实现监听IOT Client状态变化

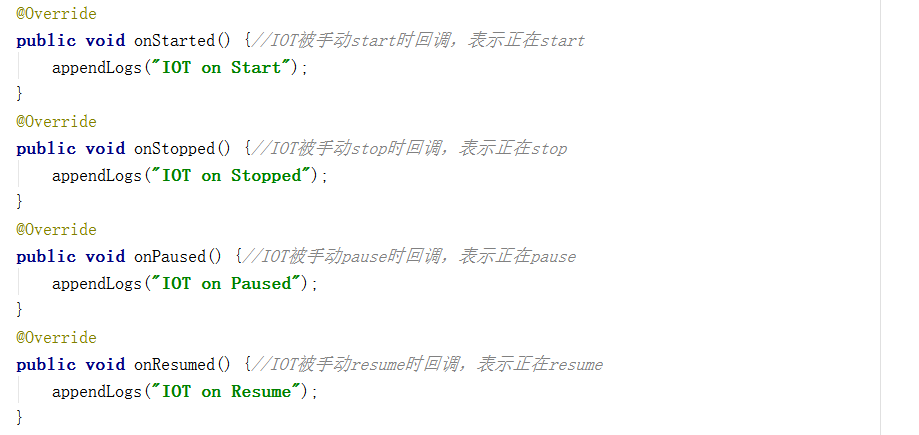


图7-9-2 IOTClientStateListener代理实现监听IOT Client状态变化



图7-10 代理实现监听Event的发送和接收情况

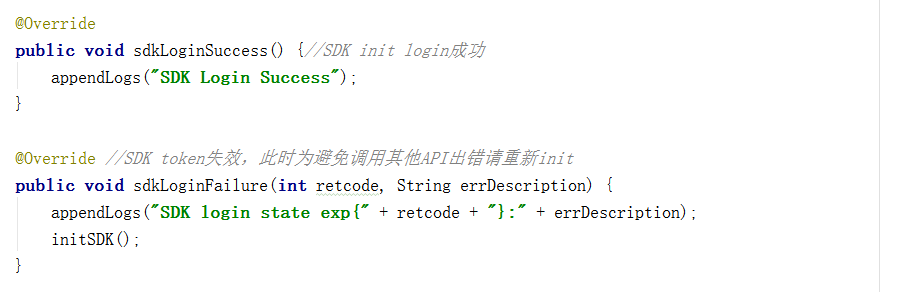


图7-11 代理实现监听SDK的Login状态是否有效

APP发送Event：

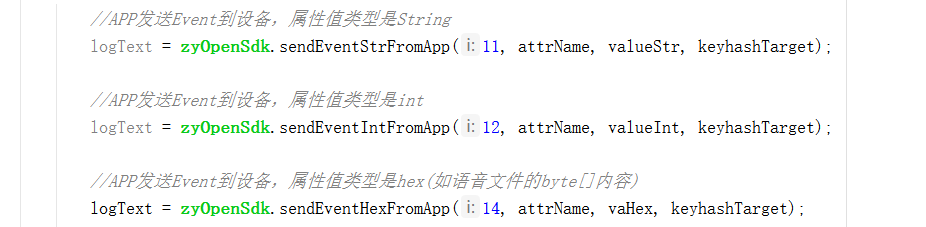


图7-11

## 设备固件升级

调用ZYIOTSDK中ZYFotaAPI的接口检测设备的固件版本更新情况，并实现对应的监听接口获取从筑云返回的版本状态，然后决定是否要升级。

代码示例：

检测新固件版本：

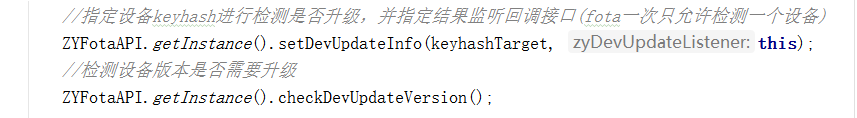


图7-12

得到版本信息决定升级：

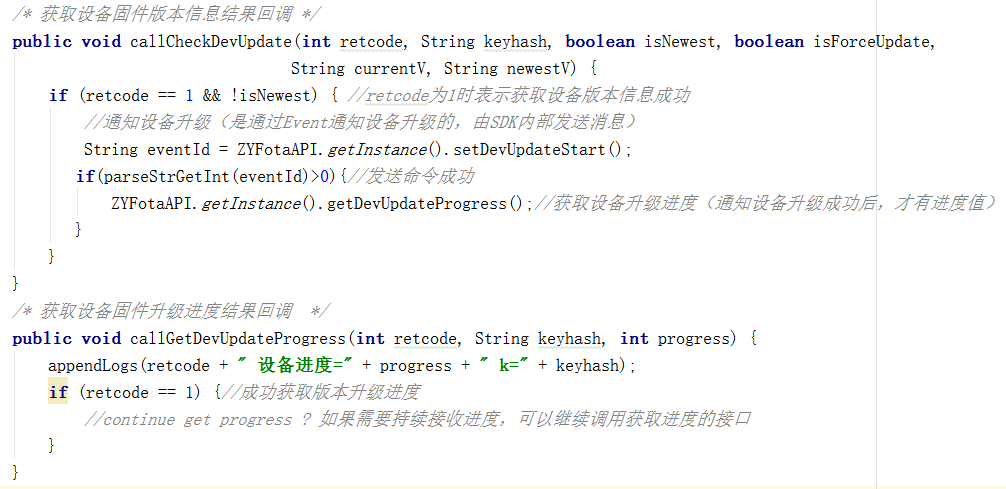


图7-13

## 关闭IOT Client

当SDK不需要与设备进行通信时，选择暂停或关闭IOT Client，可以降低通信损耗。

初始化SDK和获取设备列表时，会自动启用IOT Client。

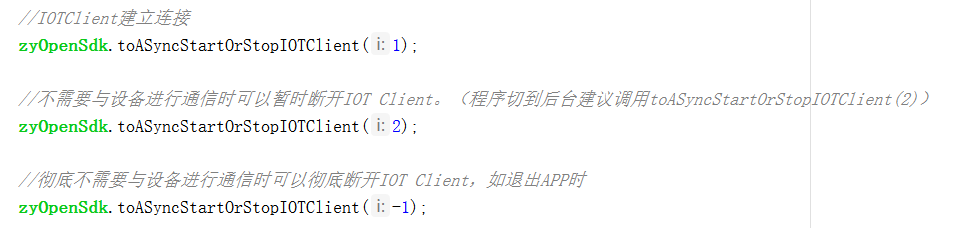


图7-14