择优互动 SDK for Android

目录

[一、修订记录 3](#_Toc485661592)

[二、简介 3](#_Toc485661593)

[三、权限开通申请 10](#_Toc485661594)

[1、申请 Key 10](#_Toc485661595)

[四、SDK 集成前必要的准备工作 10](#_Toc485661596)

[1、下载 SDK 与 DEMO 10](#_Toc485661597)

[2、开发环境要求 10](#_Toc485661598)

[3、需要导入的 AAR 10](#_Toc485661599)

[五、流媒体快速接入介绍 10](#_Toc485661600)

[1、初始化配置 10](#_Toc485661601)

[(1) 初始化 ZeyouSDK 10](#_Toc485661602)

[(2)基础参数说明 10](#_Toc485661603)

[2、用户标识 11](#_Toc485661604)

[(1) 登陆 11](#_Toc485661605)

[(2)登陆参数描述 11](#_Toc485661606)

[3、看直播 11](#_Toc485661607)

[(1) 看直播 11](#_Toc485661608)

[(2) 停止观看 13](#_Toc485661609)

[4、看回放 13](#_Toc485661610)

[(1) 看回放 13](#_Toc485661611)

[(2) 播放器方法 14](#_Toc485661612)

[六、功能接入介绍 14](#_Toc485661613)

[1、分辨率切换 14](#_Toc485661614)

# 一、修订记录

|  |  |  |  |
| --- | --- | --- | --- |
| 日期 | 版本号 | 描述 | 修订者 |
| 2017-06-19 | V0.1 | 初稿 | liulei |

# 二、简介

本文档为了指导开发者更快使用 Android 系统上的 “自助式网络直播服务 SDK”，默 认读者已经熟悉 IDE 的基本使用方法（本文以 Android Studio 为例），以及具有一定的编程 知识基础等。

业务层和 UI 层分离

为了方便中小型公司快速集成开发，业务层和 UI 层已经分离。分离后多了 upplayer，如 果用户不需要 UI 层的支持，可以依旧像以前一样当 Demo 使用。如果需要可以直接将upplayer依赖进自己的工程，upplayer中包含已经完成了的横竖屏发起页和观看直播回放包括功 能，用户可直接在我们的 UI 层上加上自己的业务逻辑，方便快速集成。

**1、支持的产品特性如下:**

|  |  |  |
| --- | --- | --- |
| 分类 | 特性名称 | 描述 |
| 观看直播 | 支持播放协议 | RTMP |
| 延时 | RTMP: 2-5 秒 |
| 支持解码 | H.264 |
| 观看回放 | 支持协议 | HLS |

# 三、权限开通申请

#### 1、申请 Key

请点击 API & SDK 权限申请 或致电010-59354686电话立即沟通申请， 申请后客户经 理会在线上与您直接联系。

审核通过后，可以获取开发应用的权限信息：App\_Key、App Secret\_Key。

# 四、SDK 集成前必要的准备工作

#### 1、下载 SDK 与 DEMO

Github ：<https://github.com/yunshijie/AndroidZeyouSDK>

## 2、开发环境要求

Pc 操作系统：64window 系统 JDK: 1.7 以上

Android studio : 建议使用 Android studio 2.0 以上 Android: 4.0 以上

备注： Android 设备操作系统需要 4.0 以上, 需要访问手机硬件,暂不支持模拟器开发

## 3、需要导入的 AAR

### SDK中的upplayer、zeyousdklib

# 五、流媒体快速接入介绍

### 1、初始化配置

### (1) 初始化 ZeyouSDK

在用户调用 ZeyouSDK中的任意方法前，一定要先调用 init 方法，初始化ZeyouSDK。

*/\*\**

* *VhallSDK 初始化*
* *@param Context*
* *@param APP\_KEY 权限申请时获得*
* *@param APP\_SECRET\_KEY 权限申请时获得*

*\*/*

ZeyouSDK.init(this , APP\_KEY , APP\_SECRET\_KEY);

### (2)基础参数说明

|  |  |
| --- | --- |
| 参数 | 描述 |
| id | 对应创建的房间 ID(在官网创建) |

### 2、用户标识

### (1) 登陆

调用SDK 中的 login 方法，如果不登录则默认是游客模式

以下是代码展示：

ZeyouSDK.getInstance().login(username, userpass, **new** UserInfoDataSource.UserInfoCallback()

{

@Override

public void onSuccess(UserInfo userInfo)

{ }

@Override

**public void** onError(**int** errorCode, String reason)

{}

});

### (2)登陆参数描述

|  |  |
| --- | --- |
| 参数字段 | 描述 |
| username | 用户名 |
| userpass | 用户密码 |
| UserInfoDataSource.UserInfoCallback | 回调信息 |

返回参数描述

|  |  |
| --- | --- |
| 参数字段 | 描述 |
| UserInfo | 用户信息 |

错误码

|  |  |
| --- | --- |
| 错误码 | 描述 |
| 260 | 用户不存在或登陆密码不正确 |

### 3、看直播

### (1) 看直播

WatchLive 实例：

watchLive 实例 , 这里需要将一些设置信息传入 列如 Context、containerLayout(这里需要传入一个 RelativeLayout,用于生成观看)、PlayView(自实现的播放器，播放器的控制页面需要实现com.zeyou.media.IMediaController)

一键观看直播：

WatchLive.Builder builder = new WatchLive.Builder()

.context()

.containerLayout() // 传入观看布局

.playView(**liveView**.getPlayView());

watchLive = builder.build();

一键观看直播，当 Activity 被创建 观看界面 Activity 必须包涵一个 RelativeLayout 布局 此布局需要往 ZeyouSDK 中传递 用于一键生成回放，获取 ZeyouSDK 的实例 调用 initWatch() 这里传入参数 WatchLive 实例、房间 ID(必填)、用户名、登陆密码、房间密码、用户类型、ip、操作系统、地理位置、操作系统版本等参数。

String id, String nickname, String email, String userId, String password, String roomPwd, String userType, String ip, String system, String addr, String src,

参数描述

|  |  |
| --- | --- |
| 参数字段 | 描述 |
| id | 房间ID |
| nickname | 用户名 |
| email | 暂时无用，传入null |
| userId | 暂时无用，传入null |
| password | 密码 |
| roomPwd | 房间密码，无密码传入空字符串 |
| userType | 用户类型，登陆接口返回 |
| Ip | Ip地址，登陆接口返回 |
| system | 默认Android |
| addr | 地理位置，登陆接口返回 |
| src | Android系统版本 |
| WatchLive | 观看直播实例 |
| RequestCallback | 回调信息 |

以下是代码展示 详细见 Demo

ZeyouSDK.*getInstance*().initWatch(String id, String nickname, String email, String userId, String password, String roomPwd, String userType, String ip, String system, String addr, String src, Watch watch,

, **new** ZeyouSDK.RequestCallback() {  
 @Override  
 **public void** onSuccess() {  
   
 }  
 @Override  
 **public void** onError(**int** errorCode, String msg) {  
  
  
 **switch** (errorCode) {  
 **case** ErrorCode.***REJECT\_KNOCK\_DOOR***:  
  
 **break**;  
 **case** ErrorCode.***PROCESSING\_KNOCK\_DOOR***:  
*// 2s获取一次申请状态* processingKnockDoor();  
 **break**;  
 **case** ErrorCode.***NEED\_KNOCK\_DOOR***:  
 **liveView**.showKnock();  
 **break**;  
 **case** ErrorCode.***NEED\_PASSWORD***:  
 **liveView**.showPwd();  
 **break**;  
 }  
  
 **liveView**.showToast(msg);  
 }  
  
 });

### (2) 停止观看

当用户停止观看时，需要调用 ZeyouSDK 中停止观看直播方法，调用此方法，SDK 会断开拉流。 代码展示如下

getWatchLive().stop();

## 4、看回放

### (1) 看回放

Watchplayback 实例：

Watchplayback 实例，和观看直播类似，传入 Context , ContainerLayout(这里需要传入一个 RelativeLayout,用于生成观看回放) ，PlayView(自实现的播放器，播放器的控制页面需要实现com.zeyou.media.IMediaController)。观看回放的操作和观看直 播一样，请求的方法相同，参数相同。 代码可以参考上面的观看直播。

一键观看回放：（参数和观看直播相同）

WatchPlayback.Builder builder = new WatchPlayback.Builder()

.context() // 上下文

.containerLayout() // 生成回放的布局

.playView(**liveView**.getPlayView());

watchPlayback = builder.build();

一键观看回放，参数和观看直播相同，传递的观看实例变成 WatchedPlayBack ,

|  |  |
| --- | --- |
| 参数字段 | 描述 |
| id | 房间ID |
| nickname | 用户名 |
| email | 暂时无用，传入null |
| userId | 暂时无用，传入null |
| password | 密码 |
| roomPwd | 房间密码，无密码传入空字符串 |
| userType | 用户类型，登陆接口返回 |
| Ip | Ip地址，登陆接口返回 |
| system | 默认Android |
| addr | 地理位置，登陆接口返回 |
| src | Android系统版本 |
| WatchLive | 观看直播实例 |
| RequestCallback | 回调信息 |

ZeyouSDK.*getInstance*().initWatch(String id, String nickname, String email, String userId, String password, String roomPwd, String userType, String ip, String system, String addr, String src, Watch watch,

, **new** ZeyouSDK.RequestCallback() {  
 @Override  
 **public void** onSuccess() {  
   
 }  
 @Override  
 **public void** onError(**int** errorCode, String msg) {  
  
  
 **switch** (errorCode) {  
 **case** ErrorCode.***REJECT\_KNOCK\_DOOR***:  
  
 **break**;  
 **case** ErrorCode.***PROCESSING\_KNOCK\_DOOR***:  
*// 2s获取一次申请状态* processingKnockDoor();  
 **break**;  
 **case** ErrorCode.***NEED\_KNOCK\_DOOR***:  
 **liveView**.showKnock();  
 **break**;  
 **case** ErrorCode.***NEED\_PASSWORD***:  
 **liveView**.showPwd();  
 **break**;  
 }  
  
 **liveView**.showToast(msg);  
 }  
  
 });

### (2) 播放器方法

当观看信息请求成功，虽然和观看直播请求的是相同的方法，但是逻辑处理不同，SDK 会默认得到播放地 址并设置进播放器中，用户只需调用 watchPlayback 实例中的各种方法来获取想要得到的信息。

开始播放：

getWatchPlayback().start();

暂停播放：

getWatchPlayback().pause();

停止播放：

getWatchPlayback().stop();

获取播放进度：

getWatchPlayback().seekTo(playerCurrentPosition);

获取当前播放进度：

getWatchPlayback().getCurrentPosition();

获取播放时长：

getWatchPlayback().getDuration();

是否正在播放

getWatchPlayback().isPlaying();

# 六、功能接入介绍

### 1、分辨率切换

1) 目前定义的分辨率有

public static final int DPI\_DEFAULT= 0; *// 默认* public static final int DPI\_SD= 1; *// 标清* public static final int DPI\_HD= 2; *// 高清* public

2) 功能实现 ：只需要将定义好的常量传到 watchLive 实例中 ,调用 setDefinition(pixel)

getWatchLive().setDefinition(level)

3) 获取分辨率是否可用 返回一个 map 状态为 0 不可用 ： 1 可用

getWatchLive().getDefinitionAvailable();