Table of Contents

* 欢拓云直播Android SDK文档

简介

欢拓云直播播放SDK Android版提供了适用于Android平台的视频互动直播SDK，可定制化和开发，以及提供了简单的直播、控制接口和完整的开源调用示例，帮助开发者实现Android平台上的互动直播应用。

功能列表

- 创建课程：根据讲师的需要设定课程主题、课程开始时间以及结束时间
- 视频直播：可以切换手机前后置摄像头进行直播
- 文档展示：可以将云端的素材，导入到白板区域，发起直播后，可以向用户展示文档
- 涂鸦：支持在白板区域划动进行涂鸦，轻轻松松做板书
- 图片导入：选择手机照片导入到ppt区域
- 语音模式：关闭摄像头进行纯音频直播
- 静音模式：关闭直播声音
- 聊天功能：在聊天区域用户进行文字聊天
- 问答功能：在问答区域回复用户的问题

运行环境

支持 Android 4.4 及以上版本;

导入SDK所需jar包和so包

Android Studio

引用jar、 aar

1. 将libs目录下的jar文件复制、粘贴到项目工程的Application Module的libs目录中;
2. 右键点击jar文件，并点击弹出菜单中的“Add As Library”并将jar文件作为类库添加到项目中

3.在app module的build.gradle中加入

```
    compile(name: 'kcpclient', ext: 'aar')
    compile(name: 'QuicSdk-release', ext: 'aar')
    compile(name: 'QuicProxySdk-release', ext: 'aar')
```

引用so文件

1. 在项目工程的Application Module的src/main目录中新建名为“jniLibs”的目录；
2. 将libs目录拷贝到“jniLibs”目录内

配置

1.打开AndroidManifest.xml，添加SDK需要的权限到标签下：

```
<uses-permission android:name="android.permission.CAMERA" />

<uses-permission android:name="android.permission.INTERNET" />

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />

<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />

<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>

<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

<uses-permission android:name="android.permission.RECORD_AUDIO" />

<uses-permission android:name="android.permission.WAKE_LOCK" />

<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />

<uses-permission android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION" />

<uses-feature android:name="android.hardware.camera" />

<uses-feature android:name="android.hardware.camera.autofocus" />

<uses-permission android:name="android.permission.READ_PHONE_STATE” />

<uses-permission android:name="android.permission.BLUETOOTH" />

<uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />

<!-- //部分手机（如小米等）需要将下面两个权限添加进去，蓝牙功能才能正常使用-->
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
```

2.在 application module的build.gradle引用sdk所需第三方库

```
dependencies {
    compile 'com.github.bumptech.glide:glide:4.7.1'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.7.1'
    compile 'com.google.code.gson:gson:2.8.2'
    compile 'com.squareup.retrofit2:retrofit:2.3.0'
    compile 'com.squareup.retrofit2:adapter-rxjava2:2.3.0'
    compile 'com.squareup.retrofit2:converter-gson:2.3.0'
    compile 'io.socket:socket.io-client:0.8.3'

}
```

3.混淆过滤

```
-keep class com.talkfun.**{*;}

#socket.io
-keep class io.socket.**{*;}
-keep interface io.socket.** { *; }

#retrofit
-dontwarn retrofit.**
-keep class retrofit.** { *; }
-keep interface retrofit.** { *; }
-keepattributes Signature
-keepattributes Exceptions

#glide
-keep public class * implements com.bumptech.glide.module.GlideModule
-keep public enum com.bumptech.glide.load.resource.bitmap.ImageHeaderParser$** {
**[] $VALUES;
public *;
}


#agora
-keep class io.agora.**{*;}

#ftp
-keep class it.sauronsoftware.ftp4j.**{*;}

#alibaba oss
-keep class com.alibaba.sdk.android.oss.**{*;}

#kcp
-keep class kcpclient.** { *; }
-keep class go.** { *; }

#Quic
-keep class com.wangsu.quicsdk.**{*;}
-keep class com.wangsu.proxy.quicsdk.**{*;}
-keep class com.wangsu.muf.**{*;}

#xlog
-keep class com.tencent.mars.xlog.** { *; }
-keep class com.tencent.mars.comm.* { *; }
-keep class com.tencent.mars.app.* { *; }
```

4.添加网宿加速

4.1 配置插件

拷贝插件到项目中，如：拷贝 `plugin/muf-plugin-1.1.0.jar` 到项目中 `plugin` 。

项目的gradle中配置：

```
buildscript {
    dependencies {
        classpath fileTree(dir:'plugin', include:['*.jar'])
    }
}
```

4.2app的gradle中配置插件：

```
apply plugin: 'muf'
```

初始化

- 自定义Application配置，调用CloudLiveSdkIniter.init()进行初始化

```
public class HtApplication extends Application {
    @Override
    public void onCreate() {// 程序的入口
        //设置Application
        super.onCreate();
        CloudLiveSdkIniter.init(this);
    }

}
```

UserManager

com.talkfun.cloudlivepublish.presenter.UserManager

描述：用户管理类，单例

初始化

```
UserManager instance = UserManager.getInstance();
```

成员方法说明

getInstance()

描述：获取UserManager实例

---

void login(@NonNull final Context context, String userid, String password, final ILogin.LoginCallback callback)

描述：使用主播id和密码登录

参数说明

参数	类型	描述
context	Context	上下文
userid	String	主播ID
password	String	用户密码
callback	LoginCallback	登录回调

ex:

```
instance.login(mContext,loginId,password,
               new ILogin.LoginCallback{
        void onLoginFail(int code,String error){
            ...}
        void onLoginSuccess(){
            ...}
        });
```

---

void login(@NonNull final Context context, String token, final ILogin.LoginCallback callback)

描述：使用验证token登录

参数说明

参数	类型	描述
context	Context	上下文
token	String	令牌
callback	ILogin.LoginCallback	登录回调

ex:

```
instance.login(mContext,token,
                new ILogin.LoginCallback{
        void onLoginFail(int code,String error){
            ...}
        void onLoginSuccess(){
            ...}
        });
```

void cancelLogin()

描述：取消登录

void logout(@NonNull Context context)

描述：退出登录

参数

参数	描述
context	上下文

boolean isLogin(@NonNull Context context)

描述：判断是否已登录，true：已登录，false：为登录

参数	描述
context	上下文

getUserInfo（ ）

描述：获取用户数据

void appExit()

描述：退出app

destroy()

描述：资源释放

LoginCallback

com.talkfun.cloudlivepublish.interfaces.ILogin.LoginCallback

描述：登录回调

成员方法说明

onLoginFail(int code,String error)

描述：登录失败回调

参数说明

参数	描述
code	错误code码
error	错误信息

onLoginSuccess(UserBean userBean)

描述：登录成功回调

参数说明

参数	描述
userBean	用户数据

UserBean

com.talkfun.cloudlivepublish.model.bean.UserBean

描述：用户信息

参数说明

参数	描述
bid	主播ID
nickname	主播昵称
roomid	房间ID
xid	用户ID
partnerId	合作方ID
accessToken	验证码字符串
websocket	websocket服务器地址
authtoken	身份验证凭证
bitrate	码率
fps	帧率
heartbeat	心跳
avatarHost	头像host
defaultAvatar	默认头像
powerNum	rtc权限： 对应 512，powerNum & 512，大于 0 表示有权限，否则就无权限 生活直播权限:对应32768，powerNum & 32768 做位运算 ，大于 0 表示有权限，否则，无权限
createCoursePower	自建课程权限:0 无权限 1 有权限

课程创建

CreateCoursePresenterImpl

com.talkfun.cloudlivepublish.presenter.CreateCoursePresenterImpl

初始化

```
ICourse.CreateCoursePresenter = new CreateCoursePresenterImpl();
```

成员方法说明

createCourse(String courseName, String startTime, String endTime, int modeType, int scenes, int smallType,final ICourse.CreateCourseCallback callback)

描述：创建课程

参数说明

参数	类型	说明
courseName	String	课程标题
startTime	String	起始时间
endTime	String	结束时间
modeType	int	创建模式 3:大班模式 5：rtc模式 6 混合模式
scenes	int	课程场景 1：教育直播，2：生活直播。默认 1
smallType	int	rtc小班模式 1:1v1, 2:1v6, 3:1v16
callback	ICourse.CreateCourseCallback	创建课程回调

提示：modeType 调用com.talkfun.cloudlivepublish.consts.**ModeType**

scenes 调用com.talkfun.cloudlivepublish.consts.**ScenesType**

smallType 调用com.talkfun.cloudlivepublish.consts.**SmallType**

ex:

```
presenter.createCourse(title, startTime, endTime, modeType , scenes,smallType,new ICourse.C
reateCourseCallback() {
    @Override
    public void onCreateSuccess() {
    ...
    }
    @Override
    public void onCreateFail(int code,String msg) {
    ...
    }
});
```

CoursePresenterImpl

com.talkfun.cloudlivepublish.presenter.CoursePresenterImpl

描述：课程列表相关处理类

初始化

```
ICourse.CoursePresenter presenter = new CoursePresenterImpl();
```

成员方法说明

void loadCourses(final ICourse.LoadCoursesCallback callback)

描述:加载直播课程列表数据

ex：

```
coursePresenter.loadCourses(new ICourse.LoadCoursesCallback() {
    @Override
    public void onLoadCoursesSuccess(List<CourseBean> list) {
    //加载成功返回课程列表
    ...
    }

    @Override
    public void onNoCourseData() {
    //没有课程列表数据
    ...
    }
});
```

---

void checkAllowedToLive(final CourseBean data, final ICourse.CheckAllowToLiveCallback callback)

描述：进入课程检测是否允许进入直播

ex：

```
coursePresenter.checkAllowedToLive(courseData, new ICourse.CheckAllowToLiveCallback() {
    @Override
    public void onAllowedToLive(CourseBean courseBean) {
    //允许进入直播
    ...
    }

    @Override
    public void notAllowedToLive(int code,String msg) {
    //当该帐号已经在直播时不允许进入直播
    ...
    }
});
```

void cancel()

描述：取消请求

---

void destroy()

描述：资源释放

大班直播

视图布局

添加摄像头预览视图**AspectTextureView**

```
<FrameLayout xmlns:android="http://    schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <com.talkfun.livepublish.view.AspectTextureView
        android:id="@+id/txv_preview"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
</FrameLayout>
```

添加白板视图

```
<com.talkfun.whiteboard.view.CloudWhiteBoardView
    android:id="@+id/white_board"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

初始化直播

- 创建直播逻辑实例

```
    ILive.LivePresenter mLivePresenterImpl;
    private void initLivePresenter() {
         mLivePresenterImpl = new LivePresenterImpl(mContext,cameraPreview);
         //设置推流事件监听
         mLivePresenterImpl.setStreamerListener(this);
         //设置直播事件监听
         mLivePresenterImpl.setLiveListener(this);
    }
```

- 初始化直播模块,如果prepare()返回false，则直播初始化失败不能直播。(prepare()失败一般是获取摄像头或音频录制权限失败引发)

```
    if (!mLivePresenterImpl.prepare()) {
            finish();
            return;
        }
```

初始化白板模块

- 初始化IWhiteBoard.WhiteBoardPresenter对象WhiteBoardPresenterImpl

```
IWhiteBoard.WhiteBoardPresenter mWhiteBoardPresenterImpl
mWhiteBoardPresenterImpl = new WhiteBoardPresenterImpl();
//设置白板视图对象
mWhiteBoardPresenterImpl.setWhiteBoardView(view);
```

文档列表逻辑、聊天逻辑、问答逻辑

- 初始化文档列表逻辑

```
//实例化文档、图片加载逻辑对象
IDocument.DocumentPresenter presenter;
presenter = new DocumentPresenterImpl(mContext);
//加载文档列表数据
presenter.getDocumentList(new IDocument.GetDocumentListCallback() {
        @Override
        public void onGetDocuments(List<DocDataBean> list) {
            //获取文档列表回调，如果没文档数据list为空
            if(list != null){
                ...
            }
        }
    });
//加载本地图片数据
 presenter.getPictureBucketList(new IDocument.GetDevicePictureBucketsCallback() {
        @Override
        public void onGetDevicePictureBuckets(List<DevicePictureBucket> list) {
              if(list != null){
                ...
            }
        }
    });
```

- 初始化聊天逻辑

```
IChat.ChatPresenter presenter;
presenter = new ChatPresenterImpl();
```

- 初始化问答逻辑

```
IQA.QAPresenter mQAPresenterImpl;
mQAPresenterImpl = new QAPresenterImpl();
```

载入文档、图片

- 初始化文件传输对象

```
    IFileTransfer.Presenter mFileTransfer;
    mFileTransfer = new FileTransferPresenterImpl();
```

- 载入文档资源

```
     mFileTransfer.loadDocRes(docDetailData, new IFileTransfer.LoadDocDataCallback() {
            @Override
            public void onLoadDocDataProgress(String progress) {

            }

            @Override
            public void onLoadDocDataSuccess(DocumentInfoBean data) {

            }

            @Override
            public void onLoadDocDataFail(int code,String msg) {

            }
        });
```

- 上传选中图片资源--插入图片

```
      mFileTransfer.uploadImagesByUri(context,selectImageList, new IFileTransfer.UpdateIm
ageCallback() {
            @Override
            public void onUploadImageProgress(int progress) {

            }

            @Override
            public void onUploadImageFail(String msg) {

            }

            @Override
            public void onUploadImageSuccess(List<UploadDocDataBean> dataList) {

            }
        });
```

开始直播

- 开始直播，调用ILiveIn.Presenter.startLive()申请直播

```
 mLivePresenterImpl.startLive();
```

- 开始直播回调

```
    //开始直播成功回调
public void onStartLiveSuccess() {
    setIsLiving(true);
    whiteBoardFragment.startSendOperateCommand(this);
    if (mIsStart) {
    showLiveStatusTip(UIUtil.getString(R.string.start_live_success));
    }
}

 /**
  * 开始直播失败回调
  */
 @Override
 public void onStartLiveFail(String error) {
    ivStartOrStop.setImageResource(R.mipmap.btn_start_live);
    mIsLiving = false;
    mIsStart = false;
      showLiveStatusTip(UIUtil.getString(R.string.start_live_fail));
  }

/**
 * 推流成功回调
 */
@Override
public void onStreamOpenSuccess() {
     Toast.makeText(this, "推流成功", Toast.LENGTH_SHORT).show();
}

/**
 * 推流失败回调
 */
@Override
public void onStreamOpenFail() {
    Toast.makeText(this, "推流失败", Toast.LENGTH_SHORT).show();
}
```

结束直播

```

//结束直播
mLivePresenterImpl.stopLive();
```

生命周期方法

在Activity的onResume()、onPause()、onDestroy()方法中调用相应方法

```
 @Override
protected void onResume() {
    super.onResume();
    mLivePresenterImpl.onResume();
}

@Override
protected void onPause() {
    super.onPause();
    mLivePresenterImpl.onPause();
}

@Override
protected void onDestroy() {
    super.onDestroy();
    mLivePresenterImpl.onDestroy();
}
```

小班直播

提示：具体的调用方式可参考Demo 中的LiveRtcActivity类

视图布局

添加白板视图

```
<com.talkfun.whiteboard.view.CloudWhiteBoardView
    android:id="@+id/white_board"
    android:layout_width="match_parent"
    android:layout_height="match_parent"/>
```

初始化直播

- 创建直播逻辑实例

```
LiveRtcPresenterImpl  mLiveRtcPresenterImpl = new
LiveRtcPresenterImpl(mContext,onRtcMemberListener);
          mLiveRtcPresenterImpl.setLiveListener(this);
          mLiveRtcPresenterImpl.setRtcMediaStatusListener(this);
          mLiveRtcPresenterImpl.setRtcStatusListener(this);
          mLiveRtcPresenterImpl.setWhiteboardPowerListener(this);
          mLiveRtcPresenterImpl.setRtcErrorListener(this);
          mLiveRtcPresenterImpl.setLiveDurationListener(this);
          mLiveRtcPresenterImpl.setWhiteboardRecoverListener(this);
          mWhiteBoardPresenter = new WhiteBoardPresenterImpl();
          mWhiteBoardPresenter.setWhiteboardView(pptContainer);
          mLiveRtcPresenterImpl.setWhiteBoardPresenter(mWhiteBoardPresenter);
```

注释：**LiveRtcPresenterImpl具体的调用细则请参考LiveRtcPresenterImpl类**

文档列表逻辑、聊天逻辑

- 初始化文档列表逻辑

```
//实例化文档、图片加载逻辑对象
IDocument.DocumentPresenter presenter;
presenter = new DocumentPresenterImpl(mContext);
//加载文档列表数据
presenter.getDocumentList(new IDocument.GetDocumentListCallback() {
@Override
public void onGetDocuments(List<DocDataBean> list) {
//获取文档列表回调，如果没文档数据list为空
    if(list != null){
        ...
    }
}
});
//加载本地图片数据
presenter.getPictureBucketList(new IDocument.GetDevicePictureBucketsCallback() {
@Override
public void onGetDevicePictureBuckets(List<DevicePictureBucket> list) {
        if(list != null){
            ...
        }
    }
});
```

- 初始化聊天逻辑

```
IChat.ChatPresenter presenter;
presenter = new ChatPresenterImpl();
```

- 初始化文件传输对象

```
IFileTransfer.Presenter mFileTransfer;
mFileTransfer = new FileTransferPresenterImpl();
```

- 载入文档资源

```
    mFileTransfer.loadDocRes(docDetailData, new IFileTransfer.LoadDocDataCallback() {

            @Override
            public void onLoadDocDataProgress(String progress) {

            }

            @Override
            public void onLoadDocDataSuccess(DocumentInfoBean data) {

            }

            @Override
            public void onLoadDocDataFail(int code,String msg) {

            }
        });
```

- 上传选中图片资源--插入图片

```
      mFileTransfer.uploadImagesByUri(context,selectImageList, new IFileTransfer.UpdateIm
ageCallback() {
            @Override
            public void onUploadImageProgress(int progress) {

            }

            @Override
            public void onUploadImageFail(String msg) {

            }

            @Override
            public void onUploadImageSuccess(List<UploadDocDataBean> dataList) {

            }
        });
```

开始直播

```
    //开始直播，申请直播
     mLiveRtcPresenterImpl.startLive();

        //开始直播成功回调
    public void onStartLiveSuccess() {
        setIsLiving(true);
        whiteBoardFragment.preToStart(this);
        if (mIsStart) {
        showLiveStatusTip(UIUtil.getString(R.string.start_live_success));
        }
    }

     /**
      * 开始直播失败回调
      */
     @Override
     public void onStartLiveFail(String error) {
        ivStartOrStop.setImageResource(R.mipmap.btn_start_live);
        mIsLiving = false;
        mIsStart = false;
          showLiveStatusTip(UIUtil.getString(R.string.start_live_fail));
      }
   //开启连麦上讲台功能
   mLiveRtcPresenterImpl.startRtc();

  //关闭连麦上讲台功能
  mLiveRtcPresenterImpl.stopRtc();
```

结束直播

```
//结束直播
mLiveRtcPresenterImpl.stopLive();
```

生命周期方法

在Activity的onResume()、onPause()、onDestroy()方法中调用相应方法

```
 @Override
protected void onResume() {
    super.onResume();
    mLiveInPresenterImpl.onResume();
}

@Override
protected void onPause() {
    super.onPause();
    mLiveInPresenterImpl.onPause();
}

@Override
protected void onDestroy() {
    super.onDestroy();
    mLiveInPresenterImpl.onDestroy();
}
```

课程

com.talkfun.cloudlivepublish.interfaces.ICourse.CoursePresenter

课程列表逻辑接口

```
/**
 * 加载课程信息
 * @param callback 回调
 */
void loadCourses(LoadCoursesCallback callback);

/**
 * 进入课程检测是否允许进入直播
 * @param data 指定的课程信息
 * @param callback 回调
 */
void checkAllowedToLive(CourseBean data,CheckAllowToLiveCallback callback);

/**
 * 取消
 */
void cancel();

/**
 * 注销
 */
void destroy();
```

com.talkfun.cloudlivepublish.interfaces.ICourse.LoadCoursesCallback

加载课程列表数据回调

```
/**
 * 课程列表信息获取成功
 */
void onLoadCoursesSuccess(List<CourseBean> list);

/**
 * 没有课程列表信息
 */
void onNoCourseData();
```

com.talkfun.cloudlivepublish.interfaces.ICourse.CheckAllowToLiveCallback

检测是否允许进入直播回调

```
/**
 * 允许进入直播
 * @param courseBean 进入的课程
 */
void onAllowedToLive(CourseBean courseBean);


/**
 * 不允许进入直播
 * @param code 响应码
 * @param msg 信息
 */
void notAllowedToLive(int code,String msg);
```

com.talkfun.cloudlivepublish.model.bean.CourseBean

课程信息

```
/**课程状态常量*/
/**1为未到直播时间*/
public static final int BEFORE_STATUS = 1;
/**2为在直播时间段内*/
public static final int TIMEING_STATUS = 2;
/**3为已过直播时间*/
public static final int TIME_OUT_STATUS = 3;

/**课程ID*/
public String courseId;
/**课程标题*/
public String courseName;
/**开始时间,10-01 09:00 */
public String startTime;
/**结束时间,10-1 09:00 */
public String endTime;
/**课程状态*/
public int status;
/**模式类型 (3：普通模式，5：rtc模式)*/
public int modeType;
/**rtc小班模式 1:1v1, 2:1v6, 3:1v16*/
public int smallType;
```

课程逻辑实现类

com.talkfun.cloudlivepublish.presenter.CoursePresenterImpl

课程列表逻辑实现类，实现ICourse.CoursePresenter接口

```
    public class class CoursePresenterImpl implements ICourse.CoursePresenter {
        ...
     }
```

com.talkfun.cloudlivepublish.interfaces.ICourse.CreateCoursePresenter

创建课程逻辑接口

```
    /**
     * 创建课程
     * @param courseName 课程标题
     * @param startTime 起始时间 (格式为：2016-08-25 12:10)
     * @param endTime 结束时间 (格式为：2016-08-25 12:10)
     * @param modeType 模式类型 (3：普通模式，5：rtc模式)
     * @param scenes 课程场景 1：教育直播，2：生活直播。默认 1
     * @param smallType  rtc小班模式 1:1v1, 2:1v6, 3:1v16
     * @param callback 回调
     */
    void createCourse(title, startTime, endTime, modeType，scenes,smallType,CreateCourseCa
llback callback);

    /**
     * 取消创建
     */
    void cancel();

    /**
     * 注销
     */
    void onDestroy();
```

com.talkfun.cloudlivepublish.interfaces.ICourse.CreateCourseCallback

创建课程回调接口

```
    /**
     * 创建成功
     * */
    void onCreateSuccess();

    /**
     * 创建失败
     * @param code 响应码
     * @param msg
     */
    void onCreateFail(int code,String msg);
```

com.talkfun.cloudlivepublish.presenter.CreateCoursePresenterImpl

创建课程逻辑实现类，实现ICreateCourse.Presenter接口

```
public class CreateCoursePresenterImpl implements ICourse.CreateCoursePresenter {
       ...
 }
```

WhiteBoardPresenter

com.talkfun.cloudlivepublish.interfaces.IWhiteBoard.WhiteBoardPresenter

描述：画板操作类

成员方法说明

startSendOperateCommand()

描述：开始发送操作指令

---

stopSendOperateCommand()

描述：停止发送操作指令

---

setWhiteBoardView(CloudWhiteBoardView view)

描述：设置画板

参数说明

参数	描述
CloudWhiteBoardView	画板

---

addPageDatas(List list, boolean clearLastDoc)

描述：添加文档数据

参数说明

参数	描述
list	文档数据列表
clearLastDoc	是否清空上一个文档

---

addPageDatas(DocDetailBean docDetailBean, boolean clearLastDoc, boolean clearWhiteboard)

描述：添加文档数据

参数说明

参数	描述
docDetailBean	文档数据列表
clearLastDoc	是否清空上一个文档
clearWhiteboard	是否清除白板

nextPage()

描述：翻到 下一页。

prePage()

描述：翻到上一页。

getPageInfoList()

描述：获取当前导入的数据列表

gotoPage(int position)

描述：翻到指定页码

参数说明

参数	描述
position	页码（从第0位开始）

getCurrentPage()

描述:获取当前的页码（从第0位开始）

getTotalPage()

描述:获取总页数

undoDrawable()

描述：撤销涂鸦

redoDrawable()

描述：还原涂鸦

clearPage()

描述：清除当前页涂鸦

setPaintColor(int color)

描述：设置画笔颜色值

参数说明

参数	描述
color	颜色值

getPaintColor()

描述:获取画笔颜色值

setTextSize(int size)

描述：设置文字大小

参数说明

参数	描述
size	文字大小

getTextSize()

描述：获取文字大小

setStrokeSize(int size)

描述：获取画笔的粗细

参数说明

参数	描述
size	画笔粗细

getStrokeSize()

描述：获取画笔粗细

---

setDrawType(int type)

描述：获取涂鸦类型

参数说明

参数	描述
type	涂鸦类型

描述：具体的类型可参考 DrawType类

---

getDrawType()

描述:获取涂鸦类型

---

addWhiteBoard(int color)

描述：添加白板

参数说明

参数	描述
color	白板颜色值

---

addWhiteBoard(int color, boolean isInsertDocBefore)

描述：添加白板

参数说明

参数	描述
color	白板颜色值
isInsertDocBefore	是否将白板插到文档之前，默认为false

---

setOnRedoableEditListener(OnRedoableEditListener listener)

描述：设置涂鸦还原监听

参数说明

参数	描述
listener	涂鸦还原监听类

setOnUndoableEditListener(OnUndoableEditListener listener)

描述：设置涂鸦撤销监听

参数说明

参数	描述
listener	涂鸦撤销监听类

setOnPageChangeListener(OnPageChangeListener mOnPageChangeListener)

描述：设置翻页监听

参数说明

参数	描述
listener	翻页监听

scrollUp()

描述：向上滚动

scrollDown()

描述：向下滚动

setScroll(boolean isScroll)

描述：画板是否可滚动（只有长图才能滚动）

参数说明

参数	描述
isScroll	是否可滚动

destroy()

描述：数据清除

FileTransferPresenterImpl

com.talkfun.cloudlivepublish.presenter.FileTransferPresenterImpl 描述:文件传输逻辑实现类

成员方法说明

uploadImages(List uploadImageList, UpdateImageCallback callback)

描述：上传图片

参数与说明

参数	描述
uploadImageList	本地图片地址列表
callback	上传图片回调

loadDocRes(DocDetailBean data, LoadDocDataCallback callback)

描述：加载文档资源

参数与说明

参数	描述
data	文档详细信息
callback	回调

LoadDocDataCallback

com.talkfun.cloudlivepublish.interfaces.IFileTransfer.LoadDocDataCallback

描述：加载文档数据回调

监听回调方法

onLoadDocDataProgress(int progress)

描述：加载课件文件的进度

参数说明

参数	描述
progress	加载进度

onLoadDocDataSuccess(DocDetailBean data)

描述：上传或加载成功刷新数据

参数说明

参数	描述
data	文档详细数据

onLoadDocDataFail(int code,String msg)

描述：文档加载失败

参数	描述
code	响应码
msg	失败信息

UpdateImageCallback

com.talkfun.cloudlivepublish.interfaces.IFileTransfer.UpdateImageCallback 描述:上传图片回调

回调参数说明

onUploadImageProgress(int progress)

描述：上传图片的进度

参数说明

参数	描述
progress	进度

onUploadImageFail(String msg)

描述：上传图片失败

参数说明

参数	描述
msg	上传失败

onUploadImageSuccess(List dataList)

描述：上传图片成功

参数说明

参数	描述
dataList	上传文档信息列表

注释：UploadDocDataBean具体参数可参考 上传文档信息

文档列表

DocumentPresenter

com.talkfun.cloudlivepublish.interfaces.IDocument.DocumentPresenter

描述： 文档逻辑接口

成员方法说明

void getDocumentList(GetDocumentListCallback callback)

描述：获取文档列表

参数说明

参数	描述
callback	回调

---

void loadDocument(DocDataBean data,LoadDocumentDetailCallback callback)

描述：加载文档详细信息

参数说明

参数	描述
data	文档数据
callback	回调

---

void getPictureBucketList(GetDevicePictureBucketsCallback callback)

描述：获取本地相册列表（只显示20k以上的jpg、png格式的图片）

参数说明

参数	描述
callback	回调

---

void onDestory()

描述：注销

GetDocumentListCallback

com.talkfun.cloudlivepublish.interfaces.IDocument.GetDocumentListCallback

描述：获取文档列表回调

成员方法说明

void onGetDocuments(List list)

描述：获取文档列表

参数说明

参数	描述
list	文档列表

GetDevicePictureBucketsCallback

com.talkfun.cloudlivepublish.interfaces.IDocument.GetDevicePictureBucketsCallback

描述：获取本地相册列表数据回调

成员方法说明

void onGetDevicePictureBuckets(List list)

描述：获取本地相册列表数据回调

参数说明

参数	描述
list	相册数据列表

LoadDocumentDetailCallback

com.talkfun.cloudlivepublish.interfaces.IDocument.LoadDocumentDetailCallback

描述： 加载文档详细数据回调

成员方法说明

void onLoadDocumentDetailSuccess(DocDetailBean data)

描述：加载详细文档数据成功

参数说明

参数	描述
data	详解文档数据

void onLoadDocumentDetailFailure(int code,String msg)

描述：加载详细文档数据成功

参数说明

参数	描述
code	响应码
msg	失败原因

DocumentPresenterImpl

com.talkfun.cloudlivepublish.presenter.DocumentPresenterImpl

描述：文档逻辑实现类，实现IDocument.DocumentPresenter接口

DocDataBean

com.talkfun.cloudlivepublish.model.bean.DocDataBean

描述：文档信息

参数说明

成员变量	参数类型	描述
id	String	文档ID
name	String	文档名称
thumbnail	String	缩略图url

DocDetailBean

com.talkfun.cloudlivepublish.model.bean.DocDetailBean

描述：文档详细信息

参数说明

成员变量	参数类型	描述
id	String	文档ID
name	String	文档名称
url	String	文档路径
thumbnail	String	缩略图url
pictures	List	文档页列表

PicturesBean

com.talkfun.cloudlivepublish.model.bean.PicturesBean

描述：文档页信息

参数说明

成员变量	参数类型	描述
page	String	页码
title	String	标题
urls	List	文档页url列表
thumbnailUrls	List	缩略图列表

DevicePictureBucket

com.talkfun.cloudlivepublish.model.bean.DevicePictureBucket

描述：设备相册信息

参数说明

成员变量	参数类型	描述
picList	List	设备图片路径列表
picUriList	List	设备图片Uri列表

文档扫描和上传

ScanFilesPresenter

com.talkfun.cloudlivepublish.interfaces.IScanFiles.ScanFilesPresenter

描述：文档扫描逻辑接口

成员方法说明

void doScanLocalDoc()

描述：执行扫描文件

---

void stopScanLocalDoc()

描述：停止执行扫描文件

List getScanDocList()

描述：获取已扫描的文件

---

void destroy()

描述：注销

ScanFilesPreseneterImpl

com.talkfun.cloudlivepublish.presenter.ScanFilesPreseneterImpl

描述：文档扫描逻辑实现类，实现IScanFiles.ScanFilesPresenter接口

OnScanFileCallback

com.talkfun.cloudlivepublish.interfaces.IScanFiles.OnScanFileCallback

描述：本地文档扫描回调

成员方法说明

void onScanComplete(List docList)

描述：扫描完成回调

参数说明

成员变量	参数类型	描述
docList	List	已经扫描到的本地文档信息

void onScanning(List docList)

描述：正在扫描回调

参数说明

成员变量	参数类型	描述
docList	List	已经扫描到的本地文档信息

void onScanFail(int code, String msg)

描述：扫描失败回调

参数说明

成员变量	参数类型	描述
code	int	状态码
msg	String	失败原因

IUploadDoc.UploadDocPresenter

com.talkfun.cloudlivepublish.interfaces.IUploadDoc.UploadDocPresenter

描述：文档上传逻辑接口

成员方法说明

void startUploadAndCover(LocalDocInfoBean infoBean)

描述：开始上传和转换文档

参数说明

成员变量	参数类型	描述
infoBean	LocalDocInfoBean	本地文档信息

void stopUploadOrCover(LocalDocInfoBean infoBean)

描述：停止上传或转换文档

参数说明

成员变量	参数类型	描述
infoBean	LocalDocInfoBean	本地文档信息

void stopAll()

描述：停止所有的文档上传和转换

void isUploadOrCovering()

描述：是否正在上传或转换文档

void void addObserver(IUploadDoc.IUploadObserver observer)

描述：添加文档上传观察监听

void removeAllObserver()

描述：删除所有文档上传观察监听

void destroy()

描述：注销

IUploadObserver

com.talkfun.cloudlivepublish.interfaces.IUploadDoc.ScanDocCallback.IUploadObserver

描述：文档上传观察监听接口

成员方法说明

void void onUploadInfoChange(LocalDocInfoBean infoBean)

描述：上传信息改变监听回调

参数说明

成员变量	参数类型	描述
infoBean	LocalDocInfoBean	文档信息

LocalDocInfoBean

com.talkfun.cloudlivepublish.model.bean.LocalDocInfoBean

描述：本地文档信息

成员方法说明

int getIndex()

描述：获取序列号

int getName()

描述：获取文档名称

int getPath()

描述：获取文档路径

int getSize()

描述：获取文档大小

int getProgress()

描述：获取上传进度或转换进度

int getStatus()

描述：获取文档上传状态

Type getFileType()

描述：获取文档类型

Type

描述：文档类型枚举 参数说明

成员变量	参数类型	描述
PDF	Type	pdf文档
PPT	Type	ppt文档
WORD	Type	word文档
EXCEL	Type	excel文档
UNSPECIFIED	Type	未支持文档

UploadDocState

com.talkfun.cloudlivepublish.consts.UploadDocState

描述：上传文档状态

参数说明

成员变量	参数类型	描述
UN_UPLOAD	static int	还未上传
STOP_UPLOAD	static int	停止上传
WAITING_UPLOAD	static int	等待上传
UPLOADING	static int	正在上传
FAIL_UPLOAD	static int	上传失败
SUCCESS_UPLOAD	static int	上传成功
COVERING	static int	正在转换处理
SUCCESS_COVER	static int	转换成功
FAIL_COVER	static int	转换失败
STOP_COVER	static int	停止转换

聊天

com.talkfun.cloudlivepublish.interfaces.IChat.ChatPresenter

聊天逻辑接口

```
/**
 * 发送消息
 * @param content 消息内容
 * @param callback 发送回调
 */
void sendMessage(String content,SendMessageCallback callback);

/**
 * 设置更新聊天内容监听
 * @param listener
 */
void setUpdateChatListener(UpdateChatListener listener);

/**
 * 注销
 */
void onDestory();
```

com.talkfun.cloudlivepublish.interfaces.IChat.SendMessageCallback

发送聊天回调

```
/**
 * 发送聊天消息成功
 */
void sendMessageSuccess();

/**
 * 发送聊天消息失败
 * @parm code 响应码
 * @param error
 */
void sendMessageFail(int code,String error);

/**
 *  更新聊天消息列表
 * @param list 消息列表
 * @param newNum 更新聊天条数
 */
void updateChatList(final List<ChatBean> list,int newNum);
```

com.talkfun.cloudlivepublish.interfaces.IChat.UpdateChatListener 更新聊天消息列表监听接口

```
    /**
     *  更新聊天消息列表
     * @param list 消息列表
     * @param newNum 更新聊天条数
     */
    void updateChatList(final List<ChatBean> list,int newNum);
```

com.talkfun.cloudlivepublish.presenter.ChatPresenterImpl

聊天逻辑实现类，实现IChat.ChatPresenter接口

```
    public class ChatPresenterImpl implements IChat.ChatPresenter {
        ...
     }
```

com.talkfun.cloudlivepublish.model.bean.ChatBean

聊天数据类

参数	类型	描述
xid	int	用户ID
uid	String	第三方合作ID
nickname	String	用户昵称
role	String	用户角色 spadmin:主播 admin:助播 user:普通用户
avatar	String	头像
msg	String	消息
time	int	时间
gender	int	性别
state	ChatState	是否禁言

注释：role 可调用com.talkfun.cloudlivepublish.consts.**MemberRole**

Socket管理

com.talkfun.cloudlivepublish.model.SocketManager

描述:socket管理类单例

成员方法说明

getInstance()

描述：获取socketManager实例

---

connected()

描述：socket 是否已连接

---

emit(String cmd,Object... args)

描述：发送消息

参数说明

参数	描述
cmd	指令
args	参数

---

on(String cmd, Emitter.Listener callback)

描述： 注册事件监听

参数说明

参数	描述
cmd	指令
callback	事件回调

---

off()

描述：取消所有注册事件的监听

---

off(String cmd, Emitter.Listener listener)

描述：取消某个注册事件的监听

参数说明

参数	描述
cmd	指令
callback	事件回调

destroy()

描述：销毁SockManager实例

LiveToolsCommendSendManager

com.talkfun.cloudlivepublish.presenter.LiveToolsCommendSendManager

描述：直播互动工具指令发送管理类

成员方法说明

sendAnnouncement(String content, final Callback callback)

描述：发送公告

参数说明

参数	描述
content	公告内容
callback	回调

sendNotice(String content, String link, int duration, final Callback callback)

描述：发送滚动通知

参数说明

参数	描述
content	通知内容
link	通知跳转链接
duration	通知持续时间 单位秒
callback	回调

sendBroadCast(int auto, String message, final Callback callback)

描述：发送广播

参数说明

参数	描述
auto	自动广播消息 自动广播时值为1 否则为0
message	广播消息内容
callback	回调

sendInteractionBroadCast(String message, final Callback callback)

描述：互动工具广播

参数说明

参数	描述
message	数据结构
callback	回调

message数据结构说明

> toolType|operateType|time|other|... // message 信息可根据需要添加字段。

字段	类型	描述
toolType	int	互动工具类型。1：转盘，2：定时器，3：抢答器，4：视频，5：音频
operateType	int	操作类型。0：应用，1：播放，2：暂停，3：拖动，4：关闭，5：结束，6：学员抢答
time	int	操作时间点

点名模块

getRollRecordList(final Callback callback)

描述：获取点名签到记录

参数说明

参数	描述
callback	回调

startRoll(int duration, final Callback callback)

描述：发起点名签到

参数说明

参数	描述
duration	时长(传入秒数，例如，1分钟传60， 3分钟传180，以此类推)
callback	回调

endRoll(int signId, final Callback callback)

描述：结束点名签到

参数说明

参数	描述
signId	发起点名时获取到的ID
callback	回调

checkRollDetail(int signId, int page, int signed, int rows, int total, final Callback callback)

描述：查看点名详情

参数说明

参数	描述
signId	发起点名时获取到的ID
page	页数
signed	是否只获取已签到的用户。0或不传，获取全部用户；1 只获取已签到用户
rows	总数
total	总数，获取第一页时传入0，接口返回总数后，后续的请求需要带上总数

抽奖模块

getLotteryList(String searchType, String searchVal, final Callback callback)

描述：抽奖记录

参数说明

参数	描述
searchType	查询类型。 courseId, liveId, roomId
searchVal	查询值
callback	回调

startLottery(final Callback callback)

描述：开始抽奖

参数说明

参数	描述
callback	回调

stopLottery(String lotteryNum, final Callback callback)

描述：结束抽奖

参数说明

参数	描述
lotteryNum	中奖人数
callback	回调

投票模块

getVoteList(final Callback callback)

描述：投票记录

参数说明

参数	描述
callback	回调

checkVoteDetail(String vid, final Callback callback)

描述：查看投票详情

参数说明

参数	描述
vid	投票ID
callback	回调

deleteVote(String vid, final Callback callback)

描述：删除投票

参数说明

参数	描述
vid	投票ID
callback	回调

endVote(int publicVote, String vid, final Callback callback)

描述：结束投票

参数说明

参数	描述
publicVote	是否公布投票答案 公布为1 不公布为0
vid	投票ID
callback	回调

---

voteUploadPic(File file, final Callback callback)

描述：上传投票的图片

参数说明

参数	描述
File	图片文件
callback	回调

---

publishVote(VoteBean voteBean, final Callback callback)

描述：传统的发起投票（保存并广播出去）

参数说明

参数	描述
VoteBean	投票数据
callback	回调

注释 ：VoteBean 类具体请参考 投票数据

saveVote(VoteBean voteBean, final Callback callback)

描述： 保存投票

参数说明

参数	描述
VoteBean	投票数据
callback	回调

注释 ：VoteBean 类具体请参考 投票数据

publishSavedVote(String vid, final Callback callback)

描述： 发布已保存(预设)的投票

参数说明

参数	描述
vid	投票ID
callback	回调

常量

com.talkfun.cloudlivepublish.consts.Constants

```
    //版本号
    public static final String VERSION;
```

com.talkfun.cloudlivepublish.consts.MemberRole 成员角色常量

```
    //超级管理员（主播）
    public  static final String MEMBER_ROLE_SUPER_ADMIN;

    //普通管理员 （助播）
    public  static final String MEMBER_ROLE_ADMIN;

    //普通用户
    public  static final String MEMBER_ROLE_USER;
```

com.talkfun.cloudlivepublish.consts.PushStreamState

描述：推流状态

```
public static final int STOP_STATE = 0; //停止状态
public static final int STARTED_STATE = 1; //开始状态
public static final int CONNECTING_STATE = 2; //连接状态
public static final int RECONNECTING_STATE =3; //重连状态
```

Setting设置管理

com.talkfun.cloudlivepublish.model.SettingManager

描述:设置管理类

成员方法说明

setApplication(@NonNull Application application)

描述：设置Application

---

getCacheSize()

描述：获取缓存大小

---

setIsOpenSkinBlurFunction(Context context, boolean isOpen)

描述：设置是否开启美颜功能

参数说明

参数	描述
context	上下文
isOpen	是否开启美颜功能

---

isOpenSkinBlurFunction()

描述： 是否开启美颜功能

---

clearCache()

描述：清除缓存

---

setIsOpenUploadAutoFunction(Context context, boolean isOpen)

描述：取消某个注册事件的监听

参数说明

参数	描述
context	上下文
isOpen	是否开启自动上传功能

isOpenUploadAutoFunction()

描述：是否有开启自动上传功能

回放资源服务管理

com.talkfun.cloudlivepublish.vod.VODResServiceManager

```
 /**
     * 绑定启动回放资源管理服务
     * @param context
     * @param callback ServiceConnection回调
     * @return
     */
public static boolean bindService(Context context,final ServiceConnection callback);

 /**
     * 解绑启动回放资源管理服务
     * @param context
     */
 public static void unBindService(Context context);


/**
 * 在上传回放页--获取所有的课程上传信息
 *
 * @return List<CourseUploadInfo> 上传资源信息列表
 */
public static List<CourseUploadInfo> getCourseUploadInfos()


/**
 * 上传全部的课程回放资源
 */
public static void uploadAll()

/**
     * 上传指定课程回放资源
     *
     * @param courseId 课程id
     */
    public static void upload(String courseId) ;

    /**
     * 上传指定课程回放资源
     * @param uploadInfo 课程上传信息
     */
    public static void upload(CourseUploadInfo uploadInfo);

    /**
     * 重新上传指定课程回放资源
     * @param courseId 课程id
     */
    public static void reUpload(String courseId);
```

```
    /**
     * 重新上传指定课程回放资源
     * @param uploadInfo 课程上传信息
     */
    public static void reUpload(CourseUploadInfo uploadInfo) ;

    /**
     * 暂停上传指定课程回放资源
     * @param courseId 课程id
     */
    public static void pause(String courseId) ;

    /**
     * 暂停上传指定课程回放资源
     * @param uploadInfo 课程上传信息
     */
    public static void pause(CourseUploadInfo uploadInfo);

    /**
     * 暂停上传所有课程回放资源
     */
    public static void pauseAll() ;

    /**
     * 判断课程是否正在上传回放资源
     * @param uploadInfo 课程上传信息
     * @return
     */
    public static boolean isUploading(CourseUploadInfo uploadInfo);

    /**
     * 判断课程是否正在上传回放资源
     * @param courseId 课程id
     * @return
     */
    public static boolean isUploading(String courseId);

    /**
     * 获取所有上传课程回放资源信息
     * @return
     */
    public static List<CourseUploadInfo> getCourseUploadInfos();

    /**
     * 删除课程上传数据
     * @param courseId 课程id
     */
    public static void deleteCourseData(String courseId);

    /**
     * 删除课程上传数据
     * @param uploadInfo 上传信息
     */
```

```
    public static void deleteCourseData(CourseUploadInfo uploadInfo);

    /**
     * 设置课程上传状态改变监听
     * @param courseId 课程id
     * @param listener 状态监听
     */
    public static void setOnCourseUploadStatusChangeListener(String courseId, OnCourseUpl
oadStatusChangeListener listener);

    /**
     * 删除所有课程上传状态改变监听
     */
    public static void removeAllOnCourseUploadStateListener();    //回放资源上传列表的适配器
class MyUploadAdapter extends UploadAdapter<UploadInfoBean> {
        public MyUploadAdapter(@NonNull Context context, @LayoutRes int resource, List li
st) {
            super(context, resource, list);
        }

        @Override
        public void onBindViewHolder(ViewHolder holder, int position) {
                //添加观察对象
             PlaybackResServiceManager.getInstance(context)
                             .addObserver(uploadInfoBean.courseId, holder);
        }
 }

 class ViewHolder implements PlaybackResServiceManager.UploadObserver{
        public ViewHolder(Context context) {
            ......
        }
           //实现该方法
        @Override
        public void onUploadInfoChange(final UploadInfoBean uploadInfoBean) {
         if (uploadInfoBean.state == UploadInfoBean.State.SUCCESS_UPLOAD) {
            //移除上传完成的对象
             PlaybackResServiceManager.getInstance(context)
                             .removeObserver(uploadInfoBean.courseId);
                CommomUtil.showShortToast("上传成功");
            }
            ......
        }
 }

 @Override
    protected void onDestroy() {
        super.onDestroy();
        //移除所有观察对象
        PlaybackResServiceManager.getInstance(UploadActivity.this).removeAllObserver();
    }
```

上传信息

com.talkfun.cloudlivepublish.model.bean.CourseUploadInfo

```
public class CourseUploadInfo implements Parcelable{

    //上传的课程title
    public String title;

    //上传的课程id
    public String courseId;

    //上传的状态
    public int state;

    //上传的完成的大小
    public long finishSize;

    //文件总大小
    public long totalSize;

}
```

回放资源上传状态

com.talkfun.cloudlivepublish.consts.VODResUploadStatus

```
public class VODResUploadStatus {
    public final static int UN_UPLOAD = 1;   //还未上传
    public final static int LIVEING = 0; //正在直播中
    public final static int UN_COMPRESS = 1;   //还未压缩
    public final static int COMPRESSING = 2;  //压缩中
    public final static int SUCCESS_COMPRESS = 3;  //压缩成功

    public final static int UN_UPLOAD = 4;   //还未上传
    public final static int PAUSE_UPLOAD = 5;   //暂停上传
    public final static int WAITING_UPLOAD = 6;  //等待上传
    public final static int UPLOADING = 7;    //正在上传
    public final static int FAIL_UPLOAD = 8;  //上传失败
    public final static int SUCCESS_UPLOAD = 9;  //上传成功
}
```

LivePresenterImpl

com.talkfun.cloudlivepublish.presenter.LivePresenterImpl

描述：LivePresenterImpl类是大班直播入口类

```
LivePresenterImpl(@NonNull Context context, @NonNull TextureView preview)
```

构造方法说明

参数	类型	描述
context	Context	上下文
preview	TextureView	视频容器

成员方法说明

boolean prepare()

描述：直播初始化

返回值：是否初始化成功

---

void startLive()

描述：开启直播

---

void stopLive()

描述：结束直播

---

void openLiveStream()

描述：打开直播流

---

void closeLiveStream()

描述：关闭直播流

---

int getStreamState()

描述：获取流状态,详细请参考PushStreamState

---

void closeCamera()

描述：关闭摄像头

void openCamera()

描述：打开摄像头

int getSendFrameRate()

描述：获取直播流帧率

void swapCamera()

描述：切换摄像头

boolean isMute()

描述：当前直播是否是静音

void setIsMute(boolean isMute)

描述：设置是否静音

boolean toggleMute()

描述：静音切换

boolean isCloseCamera()

描述：视频是否关闭

boolean toggleCamera();

描述：视频开关切换

注释：返回值为是否切换成功

boolean isCloseCamera()

描述：视频是否已关闭

---

void setOrientation(int orientation)

描述：设置屏幕方向

注释：orientation 值为：Configuration.ORIENTATION_PORTRAIT 或 Configuration.ORIENTATION_LANDSCAPE

---

int getOrientation()

描述：获取屏幕方向

---

void setZoomByPercent(float targetPercent);

描述：设置摄像头显示焦距缩放百分比 ，取值范围为：[0.0f,1.0f]

---

float getZoomPercent()

描述：获取焦距缩放百分比

---

void setSkinBlur(boolean isOpen)

描述：是否开启美颜

---

void setLiveListener(ILive.LiveListener listener)

描述：设置直播状态监听，具体请参考LiveListener

---

void setStreamerListener(PublishStreamListener listener)

描述：设置直播推流监听，具体请参考PublishStreamListener

List getDefinitionList()

描述：获取清晰度列表,具体请参考DefinitionBean

---

boolean setDefinition(String desc)

描述:设置清晰度

List getSpeedLineList()

描述：获取加速线路

---

void setSpeedLine(SpeedLine proxy, Callback callback)

描述：设置加速线路,详细参数参考proxy

---

void onResume()

描述：对应生命周期onResume

---

void onPause()

描述：对应生命周期onPause

---

void onPause(boolean releasCamera)

描述：对应生命周期onPause

注释：releasCamera：是否释放摄像头

---

void onDestroy()

描述：对应生命周期onDestroy，对资源进行释放。

LiveListener

com.talkfun.livepublish.event.PublishStreamListener.ILive.LiveListener

描述：直播监听接口

```
/**
* 开始直播成功
*/
void onStartLiveSuccess();

/**
 * 开始直播失败
 * @param error 失败原因
 */
void onStartLiveFail(String msg);

/**
* 结束直播成功
*/
void onStopLiveSuccess();

/**
* 结束直播失败
* @param msg
*/
void onStopLiveFail(String msg);

/**
* 踢出主播
*/
void onKickZhubo();

/**
* 直播超时
*/
void onLiveExpired();
```

PublishStreamListener

com.talkfun.livepublish.event.PublishStreamListener

描述：直播推流监听接口

```
/**
     * 直播推流成功回调
     */
    public  void onStreamOpenSuccess();

    /**
     * 直播推流失败回调
     */
    public  void onStreamOpenFail();

    /**
     * 关闭停止直播推流成功回调
     */
    public  void onStreamCloseSuccess();

    /**
     * 关闭停止直播推流失败回调
     */
    public  void onStreamCloseFail();

    /**
     * 直播推流正在重连
     */
    public void onStreamReconnecting();

    /**
     * 直播推流重连成功
     */
    public void onStreamReconnectSuccess();

    /**
     * 直播推流重连失败
     */
    public void onStreamReconnectFail();
```

清晰度

com.talkfun.cloudlivepublish.model.bean.DefinitionBean

参数类型

参数	类型	描述
desc	String	清晰度名称
select	int	是否为默认清晰度（1：是，0：否）

com.talkfun.cloudlivepublish.model.bean.SpeedLine

参数	类型	描述
name	String	加速线路名称
id	String	加速线路ID
type	int	加速线路类型
index	int	加速线路下角标
select	select	是否为默认线路

问答

com.talkfun.cloudlivepublish.interfaces.IQA.QAPresenter

问答逻辑接口

```
/**
 * 发送回复消息
 * @param content 回复内容
 * @param replyId 提问id
 * @param callback 回调
 */
void sendMessage(String content, String replyId,SendMessageCallback callback);

/**
 * 设置更新问答数据监听
 * @param listener
 */
void setUpdateQAListener(UpdateQAListener listener);

/**
 * 注销
 */
void onDestory();
```

com.talkfun.cloudlivepublish.interfaces.IQA.SendMessageCallback

发送回复信息回调

```
/**
 * 发送消息成功
 */
void sendMessageSuccess();

/**
 * 发送消息失败
 * @param code 响应码
  * @param error 失败原因
  */
void sendMessageFail(int code,String error);

/**
 * 更新问答数据列表
 * @param QAList 问答数据列表
 * @param newNum 更新条数
 */
void updateQAList(final ArrayList<QABean> QAList,int newNum);
```

com.talkfun.cloudlivepublish.interfaces.IQA.UpdateQAListener 更新问答数据监听接口

```
/**
 * 更新问答数据列表
 * @param QAList 问答数据列表
 * @param newNum 更新条数
 */
void updateQAList(ArrayList<QABean> QAList,int newNum);
```

com.talkfun.cloudlivepublish.presenter.QAPresenterImpl

问答逻辑实现类，实现.IQA.QAPresenter接口

```
public class QAPresenterImpl implements IQA.QAPresenter {
    ...
}
```

com.talkfun.cloudlivepublish.model.bean.QABean

提问数据

```
/**用户id*/
public String uid;
/**提问id*/
public int qid;
/**提问内容*/
public String content;
/**创建时间*/
public long time;
/**用户昵称*/
public String nickname;
/**
 * 角色
 * spadmin:主播 admin:助播 user:普通用户
 * 对应常量分别为
 * MemberRole.MEMBER_ROLE_SUPER_ADMIN
 * MemberRole.MEMBER_ROLE_ADMIN
 * MemberRole.MEMBER_ROLE_USER
 * */
public String role;
/**头像*/
public String avatar;
/**
 * 获取回复列表
 * @return
 */
public ArrayList<ReplyBean> getReplyList();
```

com.talkfun.cloudlivepublish.model.bean.ReplyBean 回复数据

```
/**id*/
public int qid;
public int xid;
/**回复的提问id*/
public int replyId;
/**头像*/
public String avatar;
/**提问内容*/
public String content;
/**
 * 角色
 * spadmin:主播 admin:助播 user:普通用户
 * 对应常量分别为
 * MemberRole.MEMBER_ROLE_SUPER_ADMIN
 * MemberRole.MEMBER_ROLE_ADMIN
 * MemberRole.MEMBER_ROLE_USER
 * */
public String role;
/**用户昵称*/
public String nickname;
/**创建时间*/
public long time;
```

在线成员

com.talkfun.cloudlivepublish.interfaces.IMember.MemberPresenter 在线成员逻辑接口

```
/**
 * 更新人数监听
 * @param listener
 */
void setUpdateMemberListener(UpdateMemberListener listener);

/**
 * 获取成员列表
 *
 * @param callback
 */
 void getMemberList(IMember.GetMemberListCallback callback);

/**
 * 注销
 */
void destroy();
```

com.talkfun.cloudlivepublish.interfaces.IMember.UpdateMemberListener 在线成员更新监听

```
/**
 * 更新总人数
 * @param total
 */
void onUpdateMemberNum(int total);

/**
 * 更新成员列表
 *
 * @param memberList
 */
void onUpdateMemberList(ArrayList<MemberInfoBean> memberList);
```

com.talkfun.cloudlivepublish.interfaces.IMember.GetMemberListCallback 获取成员列表回调

```
public interface GetMemberListCallback {
    void success(List<MemberInfoBean> var1);
    void fail(String var1);
}
```

com.talkfun.cloudlivepublish.presenter.MemberPresenterImpl

在线成员逻辑实现类，实现IMember.MemberPresenter接口

```
public class MemberPresenterImpl implements IMember.MemberPresenter {
    ...
}
```

LiveRtcPresenterImpl

com.talkfun.cloudlivepublish.rtc.LiveRtcPresenterImpl

描述：LiveRtcPresenterImpl类是RTC连麦直播逻辑类

```
LiveRtcPresenterImpl(@NonNull Context context, @NonNull OnRtcMemberListener onRtcMemberLi
stener)
```

构造方法说明

参数	类型	描述
context	Context	上下文
onRtcMemberListener	OnRtcMemberListener	rtc人员相关监听

注释：OnRtcMemberListener具体请参考 用于学员连接,断开连麦等功能的监听

成员方法说明

startLive()

描述：开启直播

stopLive()

描述：结束直播

openRtc(Callback callback)

描述：开启连麦功能

参数说明：

参数	描述
callback	开启连麦回调

closeRtc(Callback callback)

描述：关闭连麦功能

参数说明：

参数	描述
callback	关闭连麦返回值

up(int xid, Callback callback)

描述：同意用户上麦

参数说明：

参数	描述
xid	用户Id
callback	回调

kick(int xid, Callback callback)

描述：移除用户连麦功能

参数说明：

参数	描述
xid	用户Id
callback	回调

giveInvite(int xid, Callback callback)

描述：邀请某个用户连麦

参数说明：

参数	描述
xid	用户Id
callback	回调

cancelInvite(int xid, Callback callback)

描述：取消邀请某个用户连麦

参数说明：

参数	描述
xid	用户Id
callback	回调

openVideo(int xid, Callback callback)

描述：开启某个用户的摄像头

参数说明：

参数	描述
xid	用户Id
callback	回调

closeVideo(int xid, Callback callback)

描述：移除用户连麦功能

参数说明：

参数	描述
xid	用户Id
callback	回调

swapCamera()

描述：前后摄像头转换

openAudio(int xid, Callback callback)

描述：开启用户音频

参数说明：

参数	描述
xid	用户Id
callback	回调

closeAudio(int xid, Callback callback)

描述：移除用户连麦功能

参数说明：

参数	描述
xid	用户Id
callback	回调

openAllVideo(Callback callback)

描述：开启全体用户摄像头

参数说明：

参数	描述
callback	回调

closeAllVideo(Callback callback)

描述：关闭全体用户摄像头

参数说明：

参数	描述
callback	回调

openAllAudio(Callback callback)

描述：开启全体用户音频

参数说明：

参数	描述
callback	回调

closeAllAudio(Callback callback)

描述：关闭全体用户音频

参数说明：

参数	描述
callback	回调

giveDrawPower(int xid, Callback callback)

描述：赋予用户涂鸦功能

参数说明：

参数	描述
xid	用户Id
callback	回调

cancelDrawPower(int xid, Callback callback)

描述：取消用户涂鸦功能

参数说明：

参数	描述
xid	用户Id
callback	回调

void setLocalVideoMirrorMode(int mode)

描述：设置直播间主播摄像头画面镜像

参数说明：

参数	描述
mode	1为开启镜像 2为关闭镜像

setBeautyEffectOptions(boolean enabled, int beautyLevel)

描述：设置直播间主播摄像头美颜

参数说明：

参数	描述
enabled	是否开启美颜
beautyLevel	数值0-9 美颜级别

muteAllRemoteAudio(boolean mute)

描述：设置直播间静音

参数说明：

参数	描述
mute	是否静音

setWhiteBoardPresenter（IWhiteBoard.WhiteBoardPresenter mWhiteBoardPresenter）

描述：添加白板操作类

---

setLiveDurationListener(OnLiveDurationListener mOnLiveDurationListener)

描述：直播时长的监听

参数	描述
OnLiveDurationListener	直播时长监听类

注释：OnLiveDurationListener具体请参考 直播时长监听

---

setLiveListener(ILive.LiveListener listener)

描述：设置开启直播，关闭直播，及主播被踢出的监听

参数	描述
ILive.LiveListener	直播监听类

注释：ILive.LiveListener具体请参考 直播监听

---

setRtcMediaStatusListener(OnRtcMediaStatusListener mOnRtcMediaStatusListener)

描述：设置视频连麦多媒体状态的监听

参数	描述
OnRtcMediaStatusListener	摄像头及音频开关的监听类

注释：OnRtcMediaStatusListener具体请参考 摄像头及音频开关监听

---

setRtcErrorListener(OnRtcErrorListener mOnRtcErrorListener)

描述：设置连麦错误监听

参数	描述
OnRtcErrorListener	视频连麦错误监听类

注释：OnRtcErrorListener具体请参考 视频连麦错误监听

---

setWhiteboardPowerListener(OnWhiteboardPowerListenter mOnWhiteboardPowerListenter)

描述：设置涂鸦权限的监听

参数	描述
OnWhiteboardPowerListenter	涂鸦监听类

注释：OnWhiteboardPowerListenter具体请参考 画板涂鸦监听

---

setWhiteboardRecoverListener(OnWhiteboardRecoverListener mOnWhiteboardRecoverListener)

描述：设置涂鸦数据恢复的监听

参数	描述
OnWhiteboardRecoverListener	涂鸦数据恢复监听类

注释：OnWhiteboardPowerListenter具体请参考 画板数据恢复监听

---

setRtcStatusListener(OnRtcStatusListener mOnRtcStatusListener)

描述：设置视频状态连接监听

参数	描述
OnRtcStatusListener	视频状态连接监听

注释：OnRtcStatusListener具体请参考 视频状态连接监听

---

setOnRtcInteractionTypeListener(OnRtcInteractionTypeListener mOnRtcInteractionTypeListener)

描述：互动工具定时器 抢答器 转盘 广播接收监听

参数	描述
mOnRtcInteractionTypeListener	广播接收监听

注释：OnRtcInteractionTypeListener具体请参考 互动工具定时器、抢答器、转盘 广播接收监听

---

getRtcApplyList()

描述：获取连麦人员申请的列表 ，申请人员相关的参数具体请参考申请上麦用户数据

---

getLiveInfo()

描述：获取直播的初始化数据，具体的参数请参考直播初始化数据

---

getRtcUserEntityList()

描述：获取连麦人员列表，具体的参数请参考连麦用户数据

---

void setVideoProfile(VideoProfile videoProfile)

描述：设置主播连麦视频分辨率

参数	类型	描述
videoProfile	VideoProfile	连麦视频分辨率

注释：VideoProfile 连麦视频分辨率

List getRtcVideoProfileList()

描述：获取连麦视频分辨率列表，具体的参数请参考连麦视频分辨率

---

void setOnInitListener(ILive.OnInitListener listener)

描述：设置初始化回调监听

参数	描述
listener	初始化回调监听

注释：OnInitListener具体请参考 视频状态连接监听

---

onResume()

描述：在Activity的生命周期 触发onResume()的时候调用。

---

onDestroy()

描述：在Activity的生命周期 触发onDestroy()的时候调用，对资源进行释放。

LiveInfo

com.talkfun.cloudlivepublish.model.LiveInfo

描述:直播初始化数据

成员方法说明

void isRtcOpen()

描述：rtc功能是否打开

ILive.LiveListener

com.talkfun.cloudlivepublish.interfaces.ILive.LiveListener

描述：直播监听

成员方法说明

void onStartLiveSuccess()

描述：开启直播成功.

---

void onStartLiveFail(String error)

描述：开启直播失败

参数说明

参数	描述
error	开启直播失败原因

---

void onStopLiveSuccess()

描述：结束直播成功。

---

void onStartLiveFail()

描述：结束直播成功。

参数说明

参数	描述
error	结束直播失败原因

---

void onKickZhubo()

描述：主播被踢出，服务端可以讲主播踢出直播间。

OnRtcMemberListener

详细描述：**com.talkfun.cloudlivepublish.rtc.interfaces.OnRtcMemberListener** 接口类用于学员连接,断开连麦等功能的监听。

成员方法说明

onKick(RtcUserEntity rtcUserEntity);

描述：用户 被移除 连麦功能

参数说明

参数	描述
rtcUserEntity	连麦的用户数据

注释 ：RtcUserEntity 类具体请参考 连麦用户数据

---

onUp(RtcUserEntity rtcUserEntity, View videoView)

描述：用户上麦

参数说明

参数	描述
rtcUserEntity	连麦的用户数据
videoView	视频View

注释 ：RtcUserEntity 类具体请参考 连麦用户数据

---

onDown(RtcUserEntity rtcUserEntity);

描述：用户下麦

参数说明

参数	描述
rtcUserEntity	连麦的用户数据

注释 ：RtcUserEntity 类具体请参考 连麦用户数据

---

void onOffline(RtcUserEntity rtcUserEntity, int reason);

参数说明

参数	描述
rtcUserEntity	连麦的用户数据
reason	离线原因

注释 ：RtcUserEntity 类具体请参考 连麦用户数据

---

onApply(RtcApplyEntity mRtcApplyEntity)

描述：用户申请上麦

参数说明

参数	描述
RtcApplyEntity	申请上麦的用户数据

注释 ：RtcApplyEntity类具体请参考 上麦用户数据

---

onCancle(RtcApplyEntity mRtcApplyEntity)

描述：用户取消上麦

参数说明

参数	描述
RtcApplyEntity	申请上麦的用户数据

注释 ：RtcApplyEntity类具体请参考 上麦用户数据

---

onRespondInvite(RtcApplyEntity mRtcApplyEntity)

描述：用户接受主播邀请上麦

参数说明

参数	描述
RtcApplyEntity	申请上麦的用户数据

注释 ：RtcApplyEntity类具体请参考 上麦用户数据

RtcApplyEntity

详细描述： **com.talkfun.cloudlivepublish.rtc.entity.RtcApplyEntity类 申请上麦用户数据。**

参数说明

成员变量	参数类型	描述
xid	int	用户唯一ID
uid	String	合作方用户ID
nickname	String	昵称
role	String	角色
time	int	时间
roomid	int	房间id
status	int	申请状态
avatar	String	用户头像

RtcUserEntity

详细描述： **com.talkfun.cloudlivepublish.rtc.entity.RtcUserEntity** 类 连麦用户数据。

参数说明

成员变量	参数类型	描述
xid	int	用户唯一ID
uid	String	合作方用户ID
nickname	String	昵称
role	String	角色
time	int	时间
video	int	视频权限
audio	int	音频权限
score	int	评分
drawPower	int	涂鸦权限
avatar	String	用户头像

注释 ：角色 类型具体参考**MemberRole** ,视频权限及音频权限具体参考***MediaStatus***,涂鸦权限 具体参考***DrawPowerStatus***

公有方法

```
boolean isMe()
```

描述：是否是当前连麦用户

```
boolean isVideoOpen()
```

描述:视频是否开启

```
boolean isAudioOpen()
```

描述：音频是否开启

MemberRole

详细描述：com.talkfun.cloudlivepublish.consts.MemberRole 角色接口类

参数说明

参数	类型	描述
MEMBER_ROLE_SUPER_ADMIN（spadmin）	String	超级管理员（主播）
MEMBER_ROLE_ADMIN(admin)	String	普通管理员（助播）
MEMBER_ROLE_USER(user)	String	普通用户

MediaStatus

详细描述：com.talkfun.cloudlivepublish.rtc.consts.MediaStatus rtc 多媒体 状态类

参数说明

参数	类型	描述
OPEN_FOR_ZHUBO(1)	int	被主播打开
CLOSE_FOR_ZHUBO(0)	int	被主播关闭
OPEN_FOR_USER(11)	int	用户打开
CLOSE_FOR_USER()	int	用户关闭

DrawPowerStatus

详细描述:com.talkfun.cloudlivepublish.rtc.consts.DrawPowerStatus 涂鸦权限状态类

参数说明

静态参数	类型	描述
OPEN(1）	int	涂鸦开启
CLOSE(2)	int	涂鸦关闭

VideoProfile

详细描述： **com.talkfun.cloudlivepublish.rtc.entity.VideoProfile 连麦视频分辨率数据**

参数说明

成员变量	参数类型	描述
aspectRatio	AspectRatio	宽高分辨率
fps	int	帧率
kbps	int	码率
profile	String	分辨率名称

AspectRatio

详细描述： **com.talkfun.cloudlivepublish.rtc.entity.VideoProfile.AspectRatio**

参数说明

成员变量	参数类型	描述
w	int	宽度
h	int	高度

OnLiveDurationListener

com.talkfun.cloudlivepublish.interfaces.OnLiveDurationListener

描述：直播数据初始化监听

成员方法说明

void onInitSuccessful()

描述：直播数据初始化成功

---

void onInitFailed(String error)

描述：直播数据初始化成功失败

参数说明

参数	描述
error	失败描述

OnLiveDurationListener

com.talkfun.cloudlivepublish.interfaces.OnLiveDurationListener

描述：直播时长监听

成员方法说明

void onTime(long totalTime)

描述：直播成功之后，开启计时，返回值为时间戳。

OnRtcMediaStatusListener

com.talkfun.cloudlivepublish.rtc.interfaces.OnRtcMediaStatusListener

描述：连麦多媒体状态

成员方法说明

void onVideoClose(RtcUserEntity rtcUserEntity)

描述：监听摄像头关闭

参数说明

参数	描述
rtcUserEntity	连麦用户数据

注释：RtcUserEntity 具体参数请参考连麦用户数据

---

void onVideoOpen(RtcUserEntity rtcUserEntity)

描述：监听摄像头开启

参数说明

参数	描述
rtcUserEntity	连麦用户数据

注释：RtcUserEntity 具体参数请参考连麦用户数据

---

void onAudioOpen(RtcUserEntity rtcUserEntity)

描述：监听音频开启

参数	描述
rtcUserEntity	连麦用户数据

注释：RtcUserEntity 具体参数请参考连麦用户数据

---

void onAudioClose(RtcUserEntity rtcUserEntity)

描述：监听音频关闭

参数说明

参数	描述
rtcUserEntity	连麦用户数据

注释：RtcUserEntity 具体参数请参考连麦用户数据

---

void onOpenVideoAll()

描述：监听全体摄像头开启

---

void onCloseVideoAll();

描述：监听全体摄像头关闭

---

void onOpenAudioAll();

描述：监听全体麦克风开启

---

void onCloseAudioAll();

描述：监听全体麦克风关闭

OnRtcErrorListener

com.talkfun.cloudlivepublish.rtc.interfaces.OnRtcErrorListener

描述： 视频连麦错误监听类

成员方法说明

void onError(int code, String msg)

描述：监听错误信息

参数说明

参数	描述
code	错误码
msg	错误信息

---

void failed(String failed);

描述：回调失败。返回失败的原因

OnRtcStatusListener

com.talkfun.cloudlivepublish.rtc.interfaces.OnRtcErrorListener

描述： 视频连麦状态监听类

成员方法说明

void onConnectionInterrupted()

描述：视频连接中断

---

void onReConnectSuccess()

描述：视频重连成功

OnWhiteboardPowerListener

com.talkfun.cloudlivepublish.whiteboard.OnWhiteboardPowerListener

描述：涂鸦权限监听类

成员方法说明

void onDrawEnable(int xid)

描述：赋予涂鸦功能 监听

参数说明

参数	描述
xid	获取涂鸦权限用户的id

---

void onDrawDisable(int xid)

描述：取消涂鸦功能监听

参数说明

参数	描述
xid	被取消涂鸦权限用户id

OnWhiteboardRecoverListener

com.talkfun.cloudlivepublish.interfaces.OnWhiteboardRecoverListener

描述：画板数据恢复监听。

成员方法说明

onRecoverSuccess(boolean isPpt)

描述：画板恢复成功

参数说明

参数	描述
isPpt	是否含有文档

---

onFail(int code, String msg)

描述：画板恢复失败

参数说明

参数	描述
code	错误码
msg	错误信息

SDK下载

- Version 2.6.8（2021-8-22)

```
1、小班直播新增主讲模式、画廊模式

2、小班直播支持未连麦获取涂鸦权限

3、支持聊天@回复、删除指定聊天内容

4、修复一些已知问题
```

SDK下载

- Version 2.6.5（2021-6-22)

```
1、优化虚拟机器人聊天信息显示头像

2、修复一些已知问题
```

SDK下载

- Version 2.6.3（2021-1-15)

```
1、支持创建一对一、一对六课程类型

2、优化若干体验问题
```

  - Version 2.6.2（2020-12-4)

```
1、小班直播页面改版

2、小班直播新增多种互动工具

3、生活直播页面改版
```

- Version 2.6.1（2020-12-8)

```
1、大班直播新增摄像头镜像切换接口

2、sdk内部去除AndroidX引用
```

- Version 2.6.0（2020-11-26)

1、小班直播页面改版

2、新增公告、通知、广播、点名、抽奖、投票、抢答器、定时器、转盘等互动工具

3、批量操作连麦用户麦克风、摄像头开关

4、邀请用户连麦

5、其他优化

- Version 2.5.9（2020-10-16)

1、直播超时响应后台结束直播指令

- Version 2.5.8（2020-9-17)

1、新增小班连麦视频分辨率设置接口

  - Version 2.5.7（2020-8-14)

1、优化生活直播模式二分屏视频分辨率的设置

- Version 2.5.6（2020-8-7)

1、优化显示读取本地图片和上传

2、优化保存本地回放数据

  - Version 2.5.5（2020-7-15)

1、直播音频优化

- Version 2.5.4（2020-6-24)

1、升级生活直播聊天表情包

2、优化直播蓝牙拾音杂音

3、其他优化

- Version 2.5.3（2020-5-27)

1、直播支持蓝牙录音

2、修复直播切换清晰度和摄像头时画面显示偶尔异常

3、其他优化

- Version 2.5.2 （2020-5-6)

```
1、新增支持虚拟人数的显示

2、优化上传课件逻辑

3、修复直播过程可能出现崩溃的问题

4、修复自动更新检测
```

- Version 2.5.1 （2020-4-13)

```
1、添加大班直播时间回调

2、优化在线人员信息处理逻辑

3、修复一些BUG
```

- Version 2.5.0 （2020-3-30)

```
1、优化小班统计
```

- Version 2.4.1 （2020-3-3)

```
1、修复接收不到鲜花信息的Bug
```

- Version 2.4.0 （2020-2-26)

```
1、新增生活直播清晰度切换

2、新增生活直播加速线度切换

3、新增直播网络状态监测
```

- Version 2.3.0

```
1、新增生活直播模板页

2、支持创建生活直播
```

- Version 2.2.3 （2019-11-19)

```
1、优化Socket重连逻辑

2、优化推流重连逻辑

3、优化日志

4、修复白板PPT图片判断是否长图问题
```

- Version 2.2.2 （ 2019-10-16)

```
1、优化文档上传功能

2、修复一些BUG
```

- Version 2.2.1 （ 2019-5-23)

```
1、优化文档上传功能

2、修复一些BUG
```

- Version 2.2.0 （ 2019-4-18)

```
1、优化回放上传功能

2、优化文档上传功能

3、优化大班本地录制视频逻辑
```

- Version 2.1.0

```
1、优化RTC逻辑

2、优化画板模块

3、修复bug
```

- Version 2.0.0

```
1、新增RTC视频互动功能

2、优化一些代码逻辑
```

- Version 1.4.0

1、重构白板模块

2、优化上传逻辑

3、修复bug

- Version 1.3.5

1、增加视频焦距设置

2、修复bug

- Version 1.3.4

1、优化推流逻辑

2、修复bug

- Version 1.3.3

1、优化直播逻辑，新增推流操作接口

- Version 1.3.2

1、增加设置直播屏幕方向

2、优化代码

3、修复bug

- Version 1.3.1

1、优化代码

- Version 1.3.0

1、新增文档导入方式

2、白板增加图形与文字输入

3、在线用户列表

- Version 1.2.0

1、优化直播命令

- Version 1.1.0

  1、增加直播视频本地录制、回放上传功能

  2、增加一键美颜，涂鸦功能开关设置

  3、更新demo，增加自动上传逻辑及优化直播和回放过程中的一些切换提示

- Version 1.0.2

  1、优化推流前置摄像头镜像问题

  2、架构优化一些切换提示

- Version 1.0.1