

1.播放器

播放器的对外核心接口为 `IPLVMediaPlayer`，该接口对外的实现类有 `PLVMediaPlayer` 和 `PLVVideoView`，区别如下：

- `PLVMediaPlayer`：播放器核心类，提供播放器的完整功能，但不包含渲染的界面
- `PLVVideoView`：封装了 `PLVMediaPlayer`，默认添加了渲染的界面，可以当作一个 `View` 来使用

2.初始化

您可以通过构造方法直接创建播放器实例，例如：

```
new PLVMediaPlayer();
new PLVVideoView(context);
```

另外，由于 `PLVVideoView` 继承自 `FrameLayout`，因此您也可以在布局文件中直接声明 `PLVVideoView`，随后在代码中通过 `findViewById()` 获取实例。

3.设置数据源

通过调用接口 `setMediaResource()` 设置数据源

```
/**
 * 设置播放资源
 */
fun setMediaResource(mediaResource: PLVMediaResource)
```

在调用该接口后，默认会自动开始播放，您也可以通过播放参数配置来控制不自动起播

4.播放参数配置

通过调用接口 `setPlayerOption()` 设置播放参数

```
/**
 * 设置播放参数
 */
fun setPlayerOption(options: List<PLVMediaPlayerOption>)
```

`PLVMediaPlayerOptionEnum` 类中提供了一些常用的播放参数，您可以直接引用其中的常量，例如：

```
// 开启精准seek的参数
PLVMediaPlayerOptionEnum.ENABLE_ACCURATE_SEEK.value("1")
```

对于重复设置的参数，新设置的参数会覆盖旧的参数；如果想要清空参数，可以在 value 字段中传入空字符串

5.播放控制

播放器提供了一系列的播放控制接口，例如：

```
/**
 * 开始播放
 */
fun start()

/**
 * 暂停播放
 */
fun pause()

/**
 * 跳转播放进度到指定位置
 * @param position 指定位置，单位：毫秒
 */
fun seek(position: Long)
```

更多控制操作可以参考 `IPLVMediaPlayer` 以及它的父接口 `IPLVMediaPlayerControl`

6. 回调

播放器的状态、事件回调可以通过回调注册中心进行监听，包括：

- `IPLVMediaPlayerBusinessListenerRegistry`: 播放器业务回调注册中心
- `IPLVMediaPlayerEventListenerRegistry`: 播放器事件回调注册中心
- `IPLVMediaPlayerStateListenerRegistry`: 播放器状态回调注册中心

以监听播放/暂停状态为例，可以通过以下方式进行监听：

```
State<PLVMediaPlayerPlayingState> playingState = mediaPlayer.getStateListenerRegistry().getPlayingState();
MutableObserver<PLVMediaPlayerPlayingState> observer = playingState.observe(new
Function1<PLVMediaPlayerPlayingState, Unit>() {
    @Override
    public Unit invoke(PLVMediaPlayerPlayingState playingState) {
        // 处理逻辑
        return null;
    }
});

// 在不需要继续监听时，可以取消监听
observer.dispose();
```

7.销毁

播放结束后不再使用播放器时，应销毁播放器：

```
mediaPlayer.destroy();
```