1.播放器

播放器的对外核心接口为`IPLVMediaPlayer`，该接口的实现类是`PLVMediaPlayer`

2.初始化

您可以通过构造方法直接创建播放器实例，例如：

```
const player: PLVMediaPlayer = new PLVMediaPlayer();
```

3.设置渲染画面

播放器通过`surface`类型的 XComponent 组件进行渲染，libraryname 为`plvplayer_xcomponent`，您可以通过`setXComponent`方法将渲染画面设置给播放器：

```
XComponent({
  id: `plv_media_player_video_xcomponent`,
  type: "surface",
  libraryname: "plvplayer_xcomponent"
}) {
}
.onLoad((xcomponent) => {
  // 设置渲染画面
  this.player.setXComponent(xcomponent)
})
```

4.设置数据源

通过调用接口`setMediaResource()`设置数据源

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

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

5.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
 */
setPlayerOption(options: 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

6.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
 */
start(): void;

/**
 * 暂停播放
 */
pause(): void;

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

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

7.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：

```
const playingState: MutableState<PLVMediaPlayerPlayingState> = player.getStateListenerRegistry().playingState;
const observer = playingState.observe((state: PLVMediaPlayerPlayingState) => {
  // 处理逻辑
  const isPlaying = state === PLVMediaPlayerPlayingState.PLAYING;
});
```

8.销毁

播放结束后不再使用播放器时，应销毁播放器：

```
player.destroy();
```