

- [1.PLVLiveScenesSDK](#)
- [2.SDK 初始化](#)
- [3.SDK 对象创建](#)
- [4.登录直播页面](#)
- [5.直播页面数据初始化](#)
- [6.SDK 对象销毁](#)

1.PLVLiveScenesSDK

PLVLiveScenesSDK 类是多场景 SDK 的入口，该类提供了 SDK 初始化、直播页面数据初始化，以及各功能模块的统一入口和生命周期管理。

2.SDK 初始化

您可以使用 PLVLiveScenesSDK 的 `init` 静态方法对 SDK 进行初始化，详细使用代码可以参考 demo 项目的 `PLVEntryAbility` 类：

```
PLVLiveSceneSDK.init(context, windowStage)
```

3.SDK 对象创建

在每次进入一个新的直播页面前，需要先创建一个新的 SDK 对象，该对象内部会生成一个 `sdkId` 与该直播页面相关联，之后也能在该直播页面使用 `sdkId` 去获取 SDK 对象，详细使用代码可以参考 demo 项目的 `PLVLoginPage` 类：

```
// 创建 SDK 对象
const sdk: PLVLiveSceneSDK = PLVLiveSceneSDK.create()
// 获取 SDK 对象
PLVLiveSceneSDK.get(sdk.uniqueId)
```

4.登录直播页面

在进入直播页面之前，需要先进行登录验证，可以使用 SDK 对象的 `loginManager` 进行直播或者回放的登录，详细使用代码可以参考 demo 项目的 `PLVLoginPage` 类：

```
// 登录直播
sdk.loginManager.loginLive(userId?: string, channelId?: string, appId?: string, appSecret?: string)
// 登录回放
sdk.loginManager.loginPlayback(userId?: string, channelId?: string, appId?: string, appSecret?: string,
videoId?: string)
```

因为每次登录意味着要进入一个新的直播页面，因此在登录前需要创建一个新的 SDK 对象，如果登录失败时则需要调用该 SDK 对象 `destroy` 方法进行销毁。登录成功后，把 SDK 对象的 `uniqueId` 传递给直播页面，后续就能在直播页面使用 `PLVLiveSceneSDK` 的 `get` 方法获取到 SDK 对象了。

5.直播页面数据初始化

在进入直播页面后，需要对直播页面的数据进行初始化，在调用 SDK 对象的 `initData` 方法后，SDK 内部会进行本地资源的初始化，以及网络资源的请求，详细使用代码可以参考 demo 项目的 `PLVLWatchLayout` 类：

```
// 初始化sdk数据，如果当前模块(scenes_live模块)为har类型，传getContext(); 如果为hsp类型，则需要传
getContext().createModuleContext('模块名')
// 因sdk内部需要读取rawfile，因此这里需要传入模块的context
sdk.initData(getContext() as common.UIAbilityContext)
```

6.SDK 对象销毁

除了登录失败需要销毁 SDK 对象外，在退出直播页面的时候，也需要对 SDK 对象进行销毁以释放资源，详细使用代码可以参考 demo 项目的 `PLVLWatchLayout` 类：

```
sdk.destroy()
```