

莫邪游戏 SDK 客户端(Android)接入文档

(v 1.2.0)

修订记录

| 版本 | 时间 | 修订人 | 说明 |
|-------|------------|----------|---------------------------------------------------------------------|
| 1.0.0 | 2018.5.25 | Readyuny | 初始版本 |
| 1.0.1 | 2018.6.1 | Readyuny | 优化更新适配 |
| 1.0.2 | 2018.6.6 | Readyuny | 增加用户中心版本控制 |
| 1.0.3 | 2018.6.11 | Readyuny | 修复因主题配置导致的文字显示问题 |
| 1.0.4 | 2018.6.26 | Readyuny | 增加设备激活上报 |
| 1.0.5 | 2018.8.1 | Readyuny | 1、增加小红点提示功能 2、增加游戏公告功能 3、增加渠道分包功能 4、优化界面分辨率适配 5、优化悬浮框隐藏 |
| 1.0.6 | 2018.8.8 | Readyuny | 修复 Eclipse 版本 SDK 的兼容问题 |
| 1.0.7 | 2018.11.20 | Readyuny | 1、修复 Https 协议兼容问题 2、增加目标编译版本要求 27 |
| 1.0.8 | 2018.12.6 | Readyuny | 修复目标编译版本低于 26 时的更新应用安装问题 |
| 1.0.9 | 2018.12.25 | Readyuny | 修复用户中心未加载完关闭导致的悬浮窗显示问题 |
| 1.1.0 | 2019.3.15 | Readyuny | 1、增加提示文字本地化 2、增加初始化隐式登陆接口 |
| 1.1.1 | 2019.3.20 | Readyuny | 1、调整初始化接口 2、增加无 UI 登录及扩展登录接口 |
| 1.1.2 | 2019.4.23 | Readyuny | 增加反馈接口 |
| 1.1.3 | 2019.7.3 | Readyuny | 修改悬浮窗逻辑，移除权限需求 |
| 1.2.0 | 2021.7.12 | Readyuny | 增加防沉迷系统 |

一.取得 appid 和 appkey

appid: SDK 初始化使用

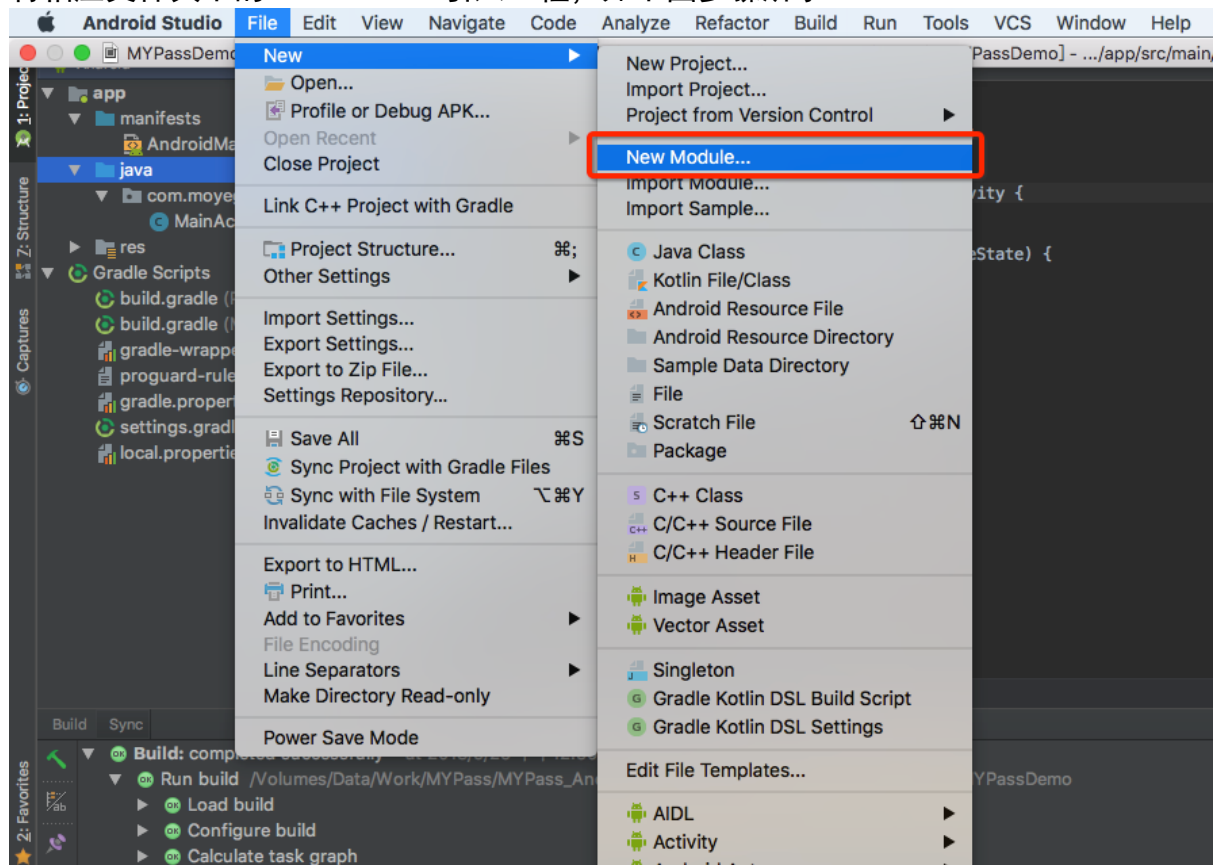
appkey: 服务器验证使用

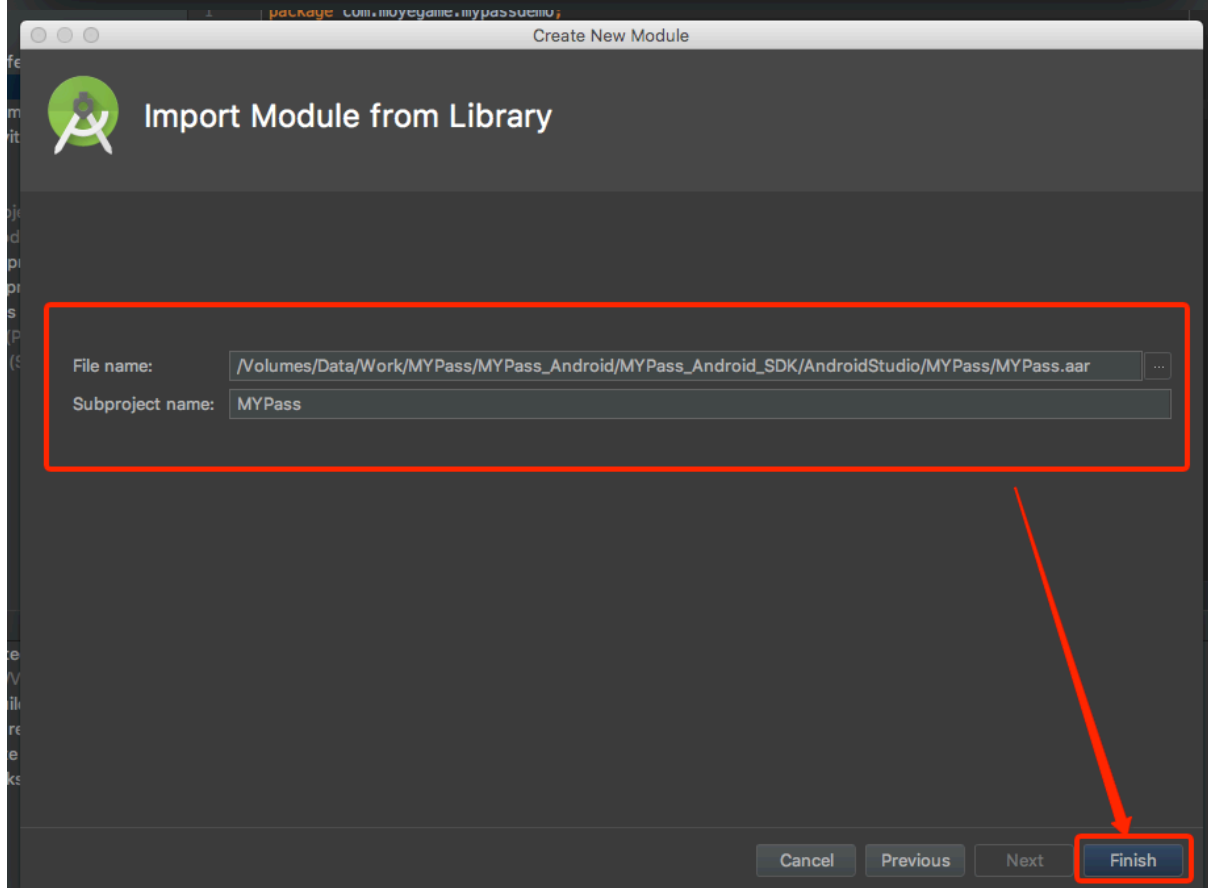
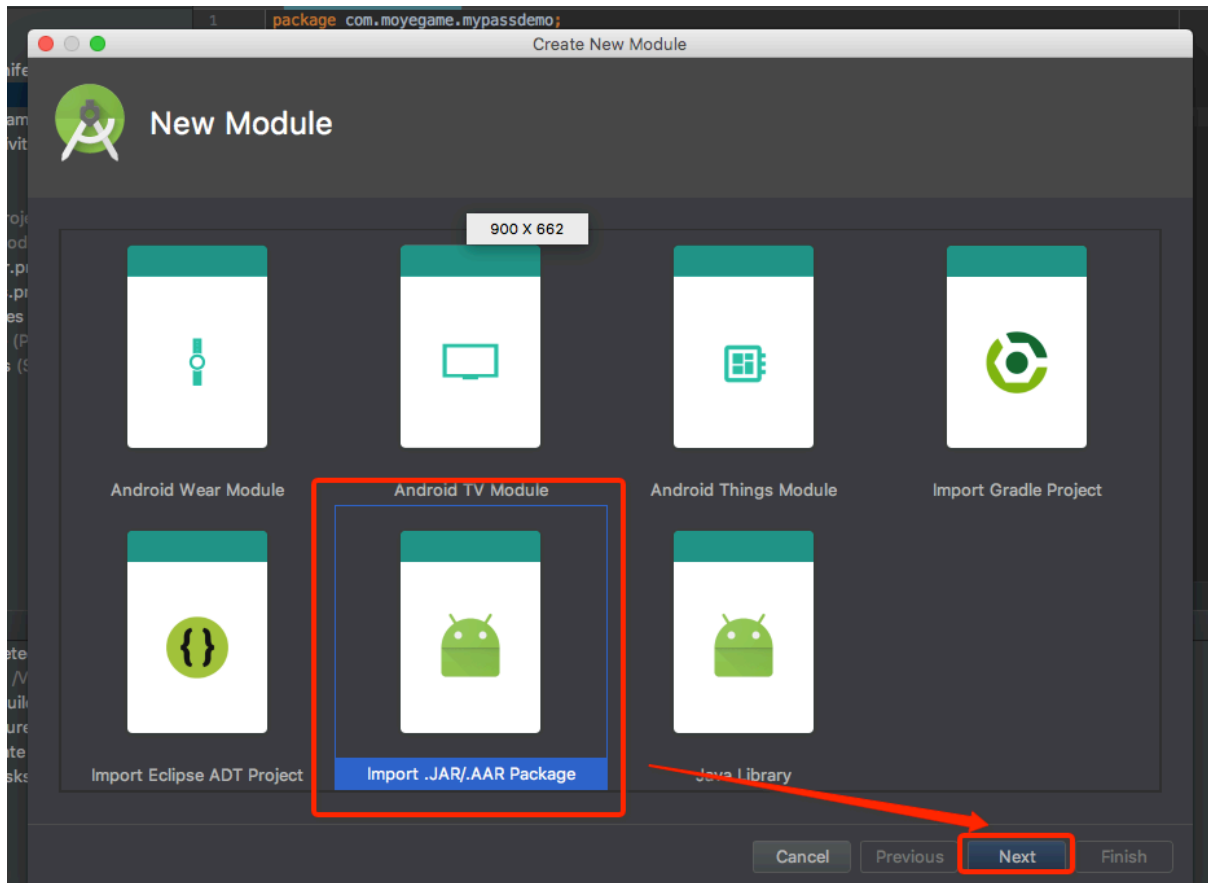
二.接入方法

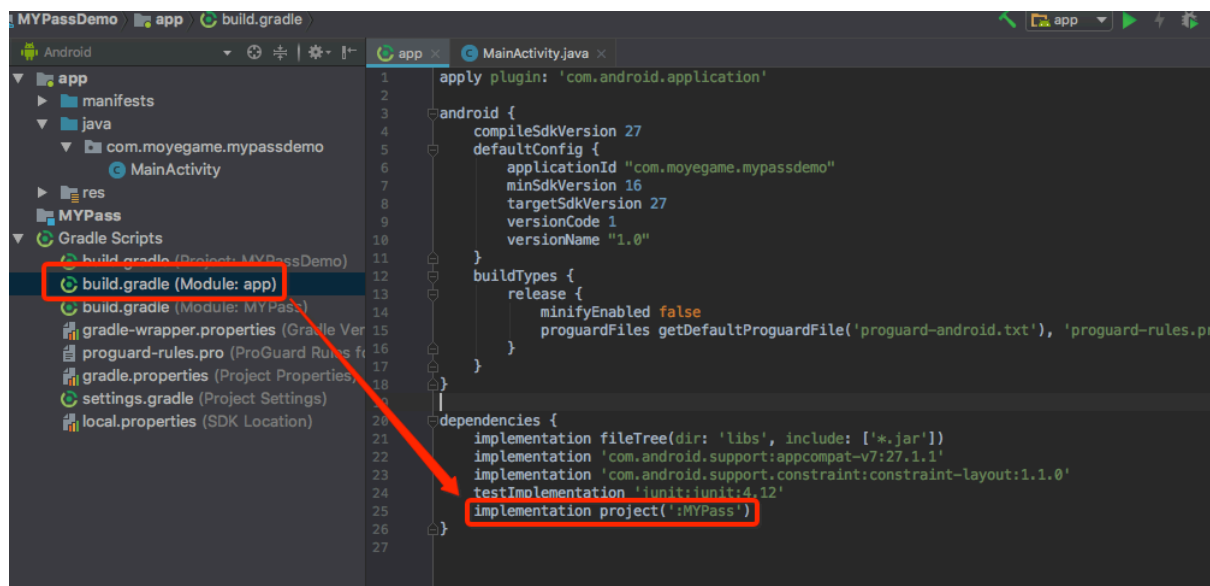
1.导入 SDK 库文件及资源配置

1.1 Android Studio 工程导入:

将相应文件夹下的 MYPass.aar 引入工程, 如下图步骤所示:



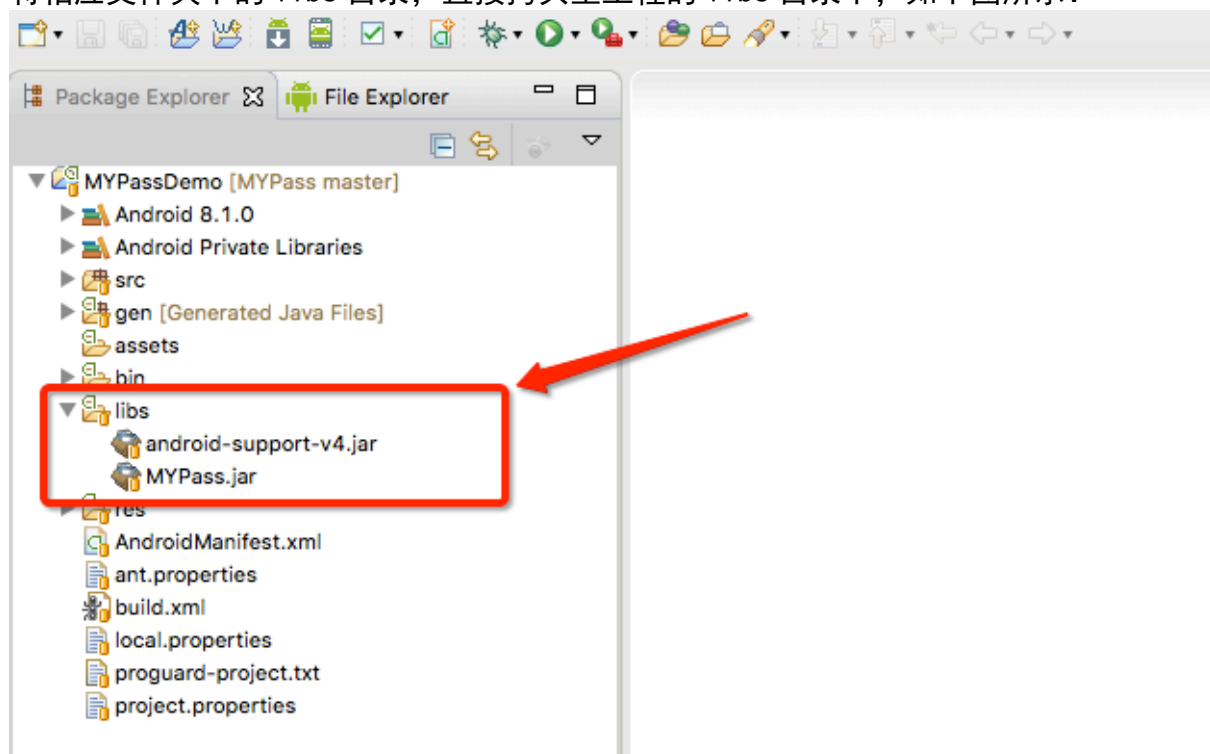




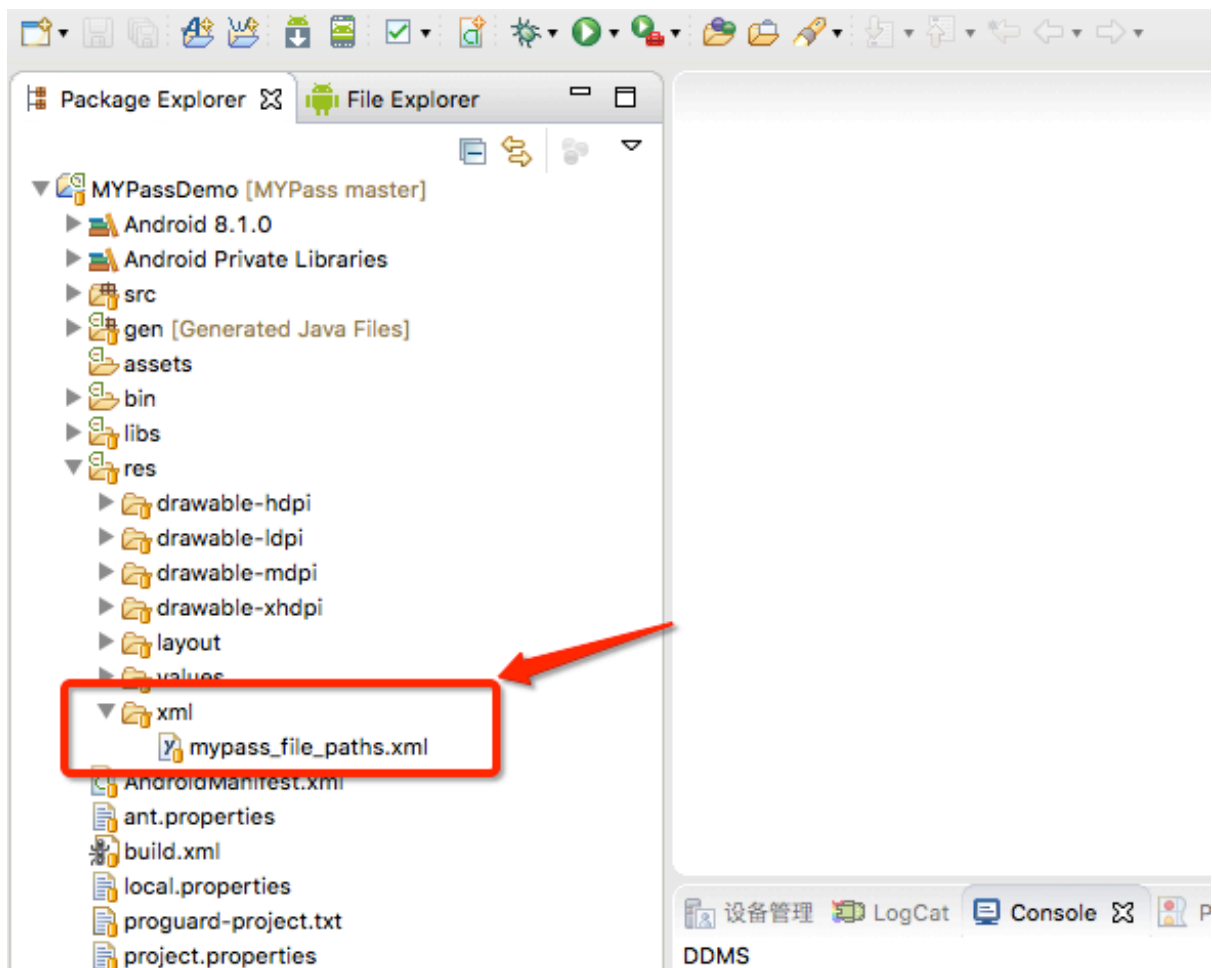
最后将其加入工程依赖，由于 SDK 需要 support 库的支持，如编译出错或运行失败，引入最新的 com.android.support 库。

1.2 Eclipse 工程导入：

将相应文件夹下的 libs 目录，直接拷贝至工程的 libs 目录下，如下图所示：



将相应文件夹下的 res 目录，直接拷贝至工程的 res 目录下，如下图所示：



1.3 AndroidManifest.xml 配置：

编译版本配置：

```
<uses-sdk  
    android:minSdkVersion="16"  
    android:targetSdkVersion="27" />
```

SDK 支持的最低版本为 16，目标最高版本为 27，因此在游戏应用设置最低及目标版本时，最低版本不可低于 SDK 要求最低版本，目标版本不可高于 SDK 要求的目标编译版本。

权限配置：

```
<!-- MYPass Permission Start -->  
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES" />  
<!-- MYPass Permission End -->
```

Application 中的配置：

```
<!-- MYPass Start -->  
<activity  
    android:name="com.moyegame.pass.account.MYAccountActivity"  
    android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"  
    android:screenOrientation="behind"  
    android:configChanges="orientation|keyboardHidden|screenSize"/>  
<activity  
    android:name="com.moyegame.pass.order.MYOrderActivity"
```

```

        android:theme="@android:style/Theme.Translucent.NoTitleBar.Fullscreen"
        android:screenOrientation="behind"
        android:configChanges="orientation|keyboardHidden|screenSize"/>
<service
    android:name="com.moyegame.pass.common.MYDownloadService"
    android:exported="false" />
<provider
    android:name="android.support.v4.content.FileProvider"
    android:authorities="应用包名.fileProvider"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/mypass_file_paths" />
</provider>
<!-- MYPass End -->

```

由于需要适配更高版本的，因此需要引入 FileProvider 配置，需要将对应配置处修改为应用的包名（如应用包名为：com.moyegame.test，则"应用包名.fileProvider"要改为"com.moyegame.test.fileProvider"）。

1.4 其他配置：

包名后缀：.moye

角标：使用素材包中的圆角角标（请勿缩放大小）

闪屏：可直接使用素材包中的闪屏图片，如需要显示贵方的 Logo 可用素材包中的 Logo 重新制作闪屏

PS:如还有疑惑，参考相应 Demo 配置！

2.SDK 的初始化和接口调用

2.1 接口定义:

| | |
|------|--------------------------------------------------------------------------------------------|
| 接口定义 | public static void initPass(Application application, String appId) |
| 接口说明 | 初始化：传入应用实例及 appId（ 请务必在 Application 的 onCreate 方法中调用，必须在初始化之后再调用其他接口 ）。 |
| 接口定义 | public static void setLogoutCallback(MYAccountInfo.MYLogoutCallback callback) |
| 接口说明 | 设置注销回调：返回注销结果 result (0: 注销成功；1: 注销失败；2: 注销取消)，注销结果信息 resultMsg（ 初始化后务必设置 ）。 |
| 接口定义 | public static void toLogin(MYAccountInfo.MYLoginCallback callback) |
| 接口说明 | 登陆：传入登陆回调，返回登陆结果 result (0: 登陆成功；1: 登陆失败；2: 登陆取消)，登陆结果信息 resultMsg，账号信息（只有登陆成功的情况下该值才不为空）。 |
| 接口定义 | public static void toLogout() |
| 接口说明 | 注销（调用后会清理用户账号信息）。 |
| 接口定义 | public static void showUserCenter() |
| 接口说明 | 显示用户中心。 |
| 接口定义 | public static void toOrder(MYOrderInfo orderInfo, MYOrderInfo.MYOrderCallback callback) |
| 接口说明 | 下订单：传入订单信息及下单回调，回调返回登录结果 result (0: 下单成功；1: 下单失败；2: 下单取消)，下单结果信息 resultMsg，订单信息。 |

2.2 接口调用方法

在需要调用 SDK 的类中引入相关类：

```
import com.moyegame.pass.MYPass;
import com.moyegame.pass.account.MYAccountInfo;
import com.moyegame.pass.order.MYOrderInfo;
```

接口初始化（请务必在 Application 的 onCreate 方法中调用该初始化接口，必须在初始化之后再调用其他接口）：

```
// 初始化 SDK
MYPass.initPass(this, "46571891337916");
```

设置注销回调（务必在初始化后调用）：

```
// 设置注销回调
MYPass.setLogoutCallback(new MYAccountInfo.MYLogoutCallback() {
    @Override
    public void onCallBack(MYAccountInfo.MYLogoutResultType result, String
resultMsg) {
        Log.d("MYPass", "----->>>>>>> Logout ResultMsg:" + resultMsg);
        switch (result) {
            case MY_LOGOUT_RESULT_SUCCESS:
                break;
            case MY_LOGOUT_RESULT_CANCEL:
                break;
            case MY_LOGOUT_RESULT_FAIL:
                break;
        }
    }
});
```

接口都是类方法，用类名 SDK 直接调用：

```
/**
 * 1. 登陆
 */
public void login() {
    MYPass.toLogin(new MYAccountInfo.MYLoginCallback() {
        @Override
        public void onCallBack(MYAccountInfo.MYLoginResultType result, String
resultMsg, MYAccountInfo accountInfo) {
            Log.d("MYPass", "----->>>>>> Login ResultMsg:" + resultMsg);
            switch (result) {
                case MY_LOGIN_RESULT_SUCCESS:
                    break;
                case MY_LOGIN_RESULT_CANCEL:
                    break;
                case MY_LOGIN_RESULT_FAIL:
                    break;
            }
        }
    });
}

/**
 * 2. 注销
 */
public void logout() {
    MYPass.toLogout();
}

/**
 * 3. 用户中心
 */
public void user() {
    MYPass.showUserCenter();
}

/**
 * 4. 下单
 */
public void pay() {
    MYOrderInfo orderInfo = new MYOrderInfo();
    orderInfo.cpUserID = "1"; // CP 用户 ID
    orderInfo.cpOrderID = "Test" + System.currentTimeMillis(); // CP 订单 ID
    orderInfo.cpProductID = "1"; // 产品 ID
    orderInfo.productName = "10 钻石"; // 产品名称
    orderInfo.quantity = 1; // 产品数量
    orderInfo.productPrice = 0.01f; // 产品总价格（元）
    orderInfo.extData = "1"; // 扩展信息（下单成功后原样通过回调地址回传）
    orderInfo.notifyUrl = "http://www.baidu.com"; // 回调地址
    MYPass.toOrder(orderInfo, new MYOrderInfo.MYOrderCallback() {
        @Override
        public void onCallBack(MYOrderInfo.MYOrderResultType result, String
resultMsg, MYOrderInfo orderInfo) {
            Log.d("MYPass", "----->>>>>> Order ResultMsg:" + resultMsg);
            switch (result) {
                case MY_ORDER_RESULT_SUCCESS:
                    break;
                case MY_ORDER_RESULT_CANCEL:
                    break;
                case MY_ORDER_RESULT_FAIL:
                    break;
            }
        }
    });
}
```



```

}

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

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

```

调用接口用到的参数模型和回调模型:

(1) MYPass.java

```

public class MYPass {

    // 初始化: 应用实例, APPID
    public static void initPass(Application application, String appID);

    // 设置注销回调: 初始化后务必设置
    public static void setLogoutCallback(MYAccountInfo.MYLogoutCallback
callback);

    // 登录: 登录回调
    public static void toLogin(MYAccountInfo.MYLoginCallback callback);

    // 注销
    public static void toLogout();

    // 显示用户中心
    public static void showUserCenter();

    // 下订单: 订单信息, 订单回调
    public static void toOrder(MYOrderInfo orderInfo, MYOrderInfo.MYOrderCallback
callback);

    // 主 Activity 恢复时调用
    public static void onResume();

    // 主 Activity 暂停时调用
    public static void onPause();

}

```

(2) MYOrderInfo.java

```

public class MYOrderInfo {

    // 订单结果码
    public enum MYOrderResultType {
        MY_ORDER_RESULT_SUCCESS, // 成功
        MY_ORDER_RESULT_FAIL, // 失败
        MY_ORDER_RESULT_CANCEL // 取消
    }

    public String cpUserID; // CP 用户 ID (由 CP 自行定义)
    public String cpOrderID; // CP 订单 ID (由 CP 自行定义)
    public String cpProductID; // CP 产品 ID (由 CP 自行定义)
    public String productName; // 产品名称
    public int quantity; // 产品数量
    public float productPrice; // 产品总价格 (元)
}

```

```

    public String extData;           // 扩展信息（下单成功后原样通过回调地址回传，长度不可超过 256）
    public String notifyUrl;         // 回调地址

    // 订单回调（结果码，结果信息，订单信息）
    public interface MYOrderCallback {
        void onCallBack(MYOrderResultType result, String resultMsg, MYOrderInfo orderInfo);
    }
}

```

(3) MYAccountInfo.java

```

public class MYAccountInfo {
    // 登陆结果码
    public enum MYLoginResultType {
        MY_LOGIN_RESULT_SUCCESS,    // 成功
        MY_LOGIN_RESULT_FAIL,       // 失败
        MY_LOGIN_RESULT_CANCEL      // 取消
    }

    // 注销结果码
    public enum MYLogoutResultType {
        MY_LOGOUT_RESULT_SUCCESS,   // 成功
        MY_LOGOUT_RESULT_FAIL,      // 失败
        MY_LOGOUT_RESULT_CANCEL     // 取消
    }

    public String userID;           // 用户 ID
    public String username;         // 用户名
    public String token;            // Token 用于服务器二次验证

    // 登陆回调（结果码，结果信息，账号信息）
    public interface MYLoginCallback {
        void onCallBack(MYLoginResultType result, String resultMsg, MYAccountInfo accountInfo);
    }


    // 注销回调（结果码，结果信息）
    public interface MYLogoutCallback {
        void onCallBack(MYLogoutResultType result, String resultMsg);
    }
}

```

三.其他必要设置

1. SDK 的初始化方法需在 Application 的 onCreate 中执行，如果没有创建或者注册过 Application 需要在 AndroidManifest.xml 中引入，如下图：（具体可参考 Demo）

```
<application
    android:name="com.moyegame.mypassdemo.MYPassApplication"
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="MYPassDemo"
    android:roundIcon="@mipmap/ic_launcher_round"
    android:supportsRtl="true"
    android:theme="@android:style/Theme.NoTitleBar.Fullscreen">
```



2. 如果 Eclipse 配置中 android-support-v4.jar 与应用原来的 jar 冲突，可以先使用应用原来的 jar，如果 SDK 正常运行可不替换，如果运行失败，可再尝试用 SDK 中的 jar 替换。

3. 如果使用 Android Studio 打包，请禁用 V2 签名，否则会影响后期分包推广效果，可在 Gradle 中添加如下配置：

```
android {
    //...
    signingConfigs {
        release {
            // 满足下面两个条件时需要此配置
            // 1. Gradle 版本 >= 2.14.1
            // 2. Android Gradle Plugin 版本 >= 2.2.0
            // 作用是只使用旧版签名，禁用 V2 版签名模式
            v2SigningEnabled false
        }
    }
}
```

PS:如遇到任何接入问题，可先参考 Demo，如还无法解决，讨论组中询问技术！