

# SDK接入说明

- [一. SDK接入配置](#)
- [二. SDK api说明](#)
  - [1. 实例SDK接口IStarpy对象](#)
  - [2. Activity生命周期调用](#)
  - [3. 初始化sdk](#)
  - [4. 设置SDK语言版本](#)
  - [5. 设置角色信息](#)
  - [6. 登录接口](#)
  - [7. 充值接口](#)

---

## • SDK接入配置

说明：由于Google 2015年6月不再支持

Eclipse, <https://developer.android.com/studio/tools/sdk/eclipse-adt.html> , Android 24之后的版本SDK和Google play servcie库, google support v4 v7库, 还有很多第三方库都已经不支持Eclipse, 固本SDK不支持Eclipse, 请使用AndroidStudio接入。

- 在您的项目中, 打开 your\_app | Gradle Scripts | build.gradle (Project) 并添加以下位置添加下面的代码, 以便从maven仓库下载 SDK:

```
//添加以下配置
allprojects {
    repositories {
        jcenter()
        google()
        maven {
            url "http://maven.starpy168.com:8081/repository/maven-public/"
            credentials {
                username "cp_dev"
                password "cpdev1688"
            }
        }
    }
}
```

- 在您的项目中, 打开 your\_app | Gradle Scripts | build.gradle (Module: app) 并添加以下编译语句, 以便编译SDK:

```

configurations.all {
    // 采用动态版本声明的依赖缓存 时间
    resolutionStrategy.cacheDynamicVersionsFor 2*60, 'seconds'
}

dependencies {
    compile fileTree(include: ['*.jar'], dir: 'libs')

    //SDK    添加以下配置以依赖SDK,具体版本对接的时候与SDK人员沟通确定
    compile 'com.starpy.sdk:starpy-sdk:2.+'
}

```

- 添加游戏配置文件，在您的项目中，打开 your\_app | assets 下创建 starpy 目录并添加 starpy\_game\_config 配置文件（该文件 SDK 对接人员提供）
- 在您的项目中，打开 your\_app | Gradle Scripts | build.gradle 中 productFlavors 内添加以下配置，以动态设置游戏配置

```

productFlavors {

    xxx {
        minSdkVersion 18
        applicationId 'xxxx'    //值我方提供
        signingConfig signingConfigs.brmmdsign //签名我方提供
        targetSdkVersion 22    //值我方提供
        versionCode 1
        versionName '1.0'
        flavorDimensions "1"

        resValue "string", "facebook_app_name", "xxx"    //值我方提供
        resValue "string", "facebook_app_id", "xxx"    //值我方提供
        resValue "string", "facebook_authorities", "xxxx"    //值我方提供
        resValue "string", "google_app_id", "xxxx"    //值我方提供

    }

}

```

## SDK api说明

以下为 sdk api 使用示例,具体请查看 SDK demo

- 实例 SDK 接口 IStarpy 对象

```
iStarpy = StarpyFactory.create();
```

## • Activity生命周期调用

游戏Activity以下相应的声明周期方法（必须调用）：

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    iStarpy = StarpyFactory.create();
    //初始化sdk
    iStarpy.initSDK(this);
    //在游戏Activity的onCreate生命周期中调用
    iStarpy.onCreate(this);
}

@Override
protected void onResume() {
    super.onResume();
    PL.i("activity onResume");
    iStarpy.onResume(this);
}

@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);

    iStarpy.onActivityResult(this, requestCode, resultCode, data);
}

@Override
protected void onPause() {
    super.onPause();
    iStarpy.onPause(this);
    PL.i("activity onPause");
}

@Override
protected void onStop() {
    super.onStop();
    PL.i("activity onStop");
    iStarpy.onStop(this);
}

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

```

        PL.i("activity onDestroy");
        iStarpy.onDestroy(this);
    }

    @Override
    public void onWindowFocusChanged(boolean hasFocus) {
        super.onWindowFocusChanged(hasFocus);
        iStarpy.onWindowFocusChanged(this, hasFocus);
    }

    /**
     * 6.0以上系统权限授权回调
     * @param requestCode
     * @param permissions
     * @param grantResults
     */
    @Override
    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {
        super.onRequestPermissionsResult(requestCode, permissions, grantResults);
        PL.i("activity onRequestPermissionsResult");
        iStarpy.onRequestPermissionsResult(this, requestCode, permissions, grantResults);
    }
}

```

## • 初始化sdk

游戏启动的时候调用，通常在游戏activity onCreate中调用

```
iStarpy.initSDK(this);
```

## • 设置SDK语言版本

根据游戏的语言，设置相应的SDK语言，默认为繁体中文  
游戏activity onCreate中调用

```

/*
 * SGameLanguage.zh_TW 游戏为繁体语言时使用
 * SGameLanguage.en_US 游戏为英文语言时使用
 * */
iStarpy.setGameLanguage(this, SGameLanguage.zh_TW);

```

## • 设置角色信息

在游戏获得角色信息的时候调用，每次登陆，切换账号等角色变化时调用（必须调用）

```
/**
 * 在游戏获得角色信息的时候调用
 * roleId 角色id
 * roleName 角色名
 * rolelevel 角色等级
 * severCode 角色伺服器id
 * serverName 角色伺服器名称
 */
iStarpy.registerRoleInfo(this, "roleid_1", "roleName", "rolelevel", "s1001", "serverName");
```

## • 登录接口

```
//登陆接口 ILoginCallBack为登录成功后的回调
iStarpy.login(MainActivity.this, new ILoginCallBack() {
    @Override
    public void onLogin(SLoginResponse sLoginResponse) {
        if (sLoginResponse != null){
            String uid = sLoginResponse.getUserId();
            String accessToken = sLoginResponse.getAccessToken();
            String timestamp = sLoginResponse.getTimestamp();

            PL.i("uid:" + uid);

        }
    }
});
```

## • 充值接口

```
/*
充值接口
SPayType SPayType.OTHERS为第三方储值，SPayType.GOOGLE为Google储值
cpOrderId cp订单号，请保持每次的值都是不会重复的
productId 充值的商品id
roleLevel 角色等级
customize 自定义透传字段（从服务端回调到cp）
注意：设置SPayType.OTHERS为第三方储值时不需要传cpOrderId, productId, customize, 传了会被忽略掉
*/
iStarpy.pay(MainActivity.this, SPayType.GOOGLE, "cpOrderId", "productId", "roleLevel", "customize");
```