

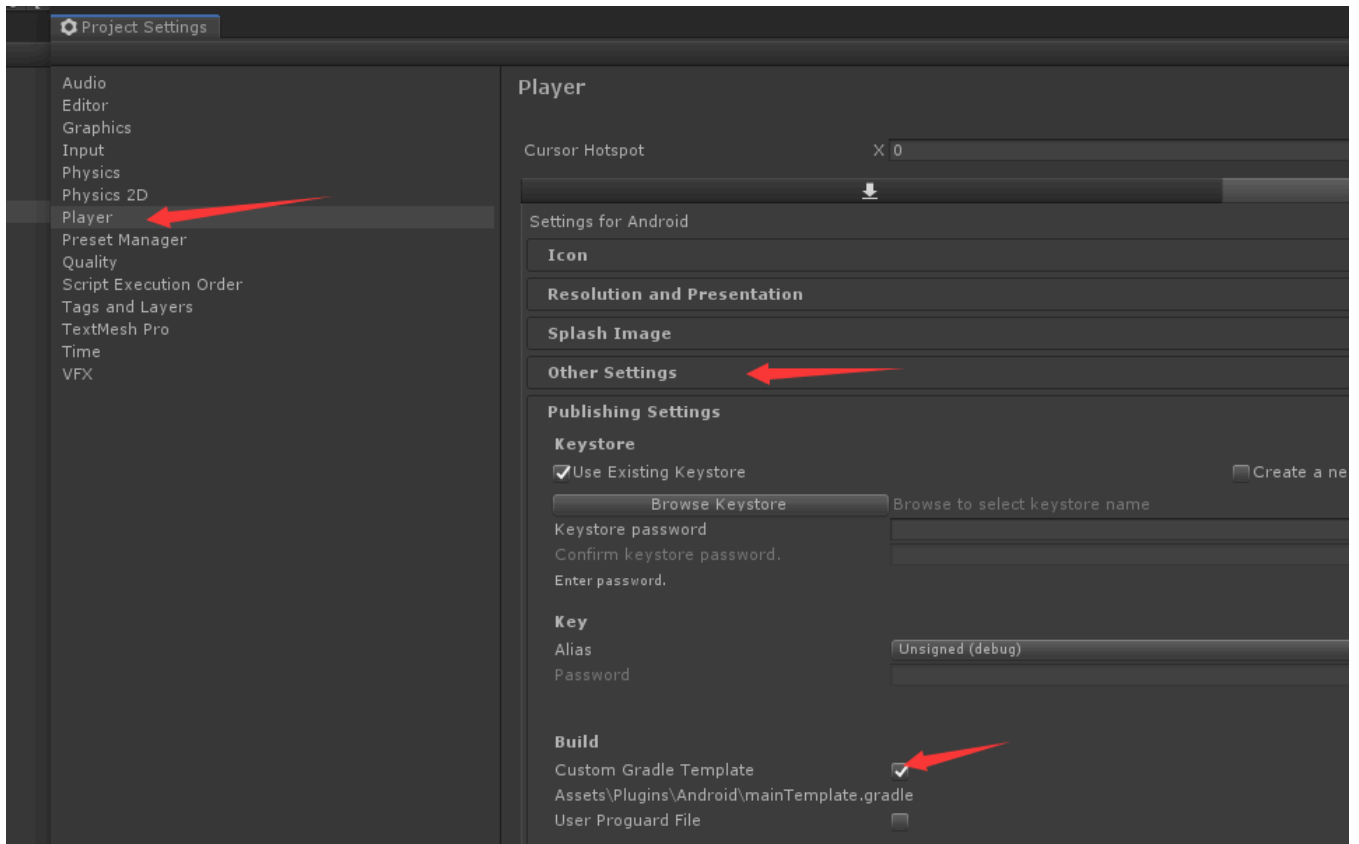
Unity 接入文档

一、接入配置

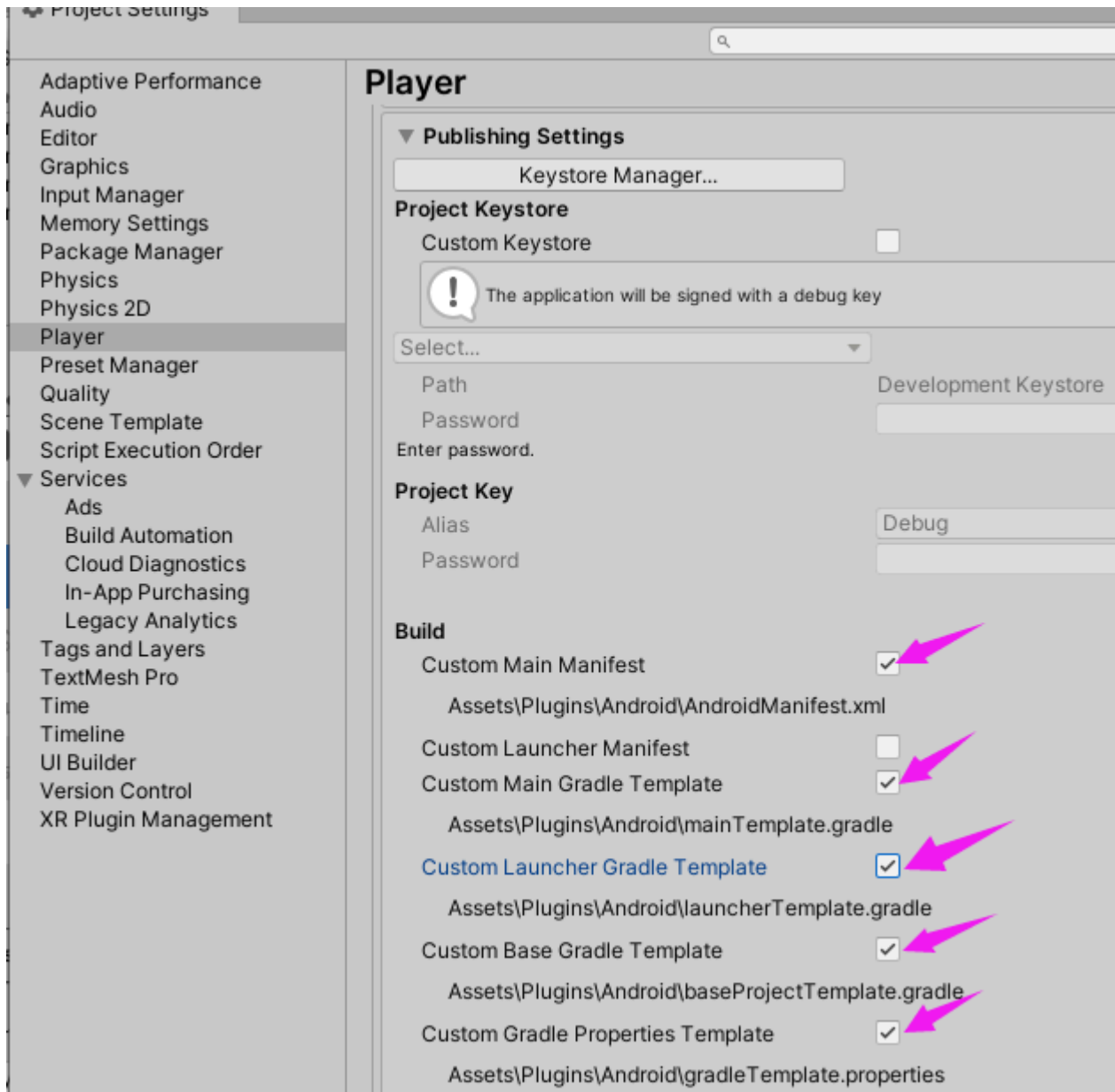
1. Unity 配置

勾选下图配置，在 `Assets/Plugins/Android` 目录下生成 `mainTemplate.gradle` 文件

unity2018 版本



unity2020 版本



2. 拷贝 SDK 资源

拷贝下列文件到 `Assets/Plugins/Android` 目录：

`Unity SDK Resources/quickgame_hw_x.x.x.aar` (x.x.x 为具体版本)

`Unity SDK Resources/unity_imp_libs-release.aar`

拷贝 `Unity SDK Resources/script` 中的文件到 Unity 项目的 `Assets/Plugins/script` 目录

导入 `Unity SDK Resources/JsonNet.9.0.1.unitypackage` 到 Unity 项目

3. 修改 `mainTemplate.gradle` 文件

在 `dependencies` 里面添加如下代码

```
dependencies {
    //必需依赖的库
    api 'androidx.appcompat:appcompat:1.6.0'
    api "org.jetbrains.kotlin:kotlin-stdlib:1.9.20"
    api 'com.android.billingclient:billing:7.0.0'

    //根据需求添加的依赖库
    //添加后sdk会获取adid
    api 'com.google.android.gms:play-services-ads-identifier:17.0.1'
    // 使用google登录需要添加
    api 'com.google.android.gms:play-services-auth:21.3.0'
}
```

4. AndroidManifest 配置

1. AndroidManifest 在 Assets/Plugins/Android 目录，在 AndroidManifest 的 application 节点下添加如下代码

```
<!--value值的格式为 QK+quick sdk后台的ProductCode,例如: productcode为123456, value值为QK123456-->
<meta-data
    android:name="QKproductCode"
    android:value="QK<这里替换为SDK的ProductCode"/>
```

2. 删掉 AndroidManifest 中配置的主 Activity，即 activity 的 <intent-filter> 中包含如下代码

```
<category android:name="android.intent.category.LAUNCHER" />
```

3. 添加 Google 登录配置（按需配置）

```
<!--没有Google登录可以不加, value值直接写在这里, 不能用@string引用-->
<meta-data
    android:name="google-signin-client_id"
    android:value="替换为google后台申请的clientid"/>
```

5. gradle.properties 配置（Unity 2020）

gradle.properties 对应 Assets/Plugins/AndroidgradleTemplate.properties 文件，在 gradle.properties 里面添加

```
android.useAndroidX=true
android.enableJetifier=true
```

6. 修改闪屏方向

闪屏默认是横屏，如果要修改闪屏方向，在AndroidManifest里面添加下面的Activity配置修改

android:screenOrientation 的值

```
<activity
    android:name="com.qk.unity.QuickUnityPlayerProxyActivity"
```

```

    android:configChanges="fontScale|orientation|keyboardHidden|locale|navigation|screenSize|ui
Mode"
    tools:replace="android:screenOrientation,android:configChanges"
    android:screenOrientation="portrait"
    android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

    <meta-data
        android:name="unityplayer.UnityActivity"
        android:value="true" />
</activity>

```

二、接口调用

添加 quicksdk 的命名空间 `using quicksdk;`

如下方法获取 SDK 实例，SDK 接口都通过该实例调用

```
QuickSDK.GetInstance()
```

1. 设置监听器

用于接收初始化、登录结果的监听回调

调用方法

```
public void setListener(QuickSDKListener quickSDKListener )
```

示例

参考 `Unity Demo/Assets/demo/EventHandle.cs` 文件

注：onLoginFailed 回调包括登录取消和登录失败

可以通过ErrorMsg的代码来区分，code为 2 表示登录取消，为 3 是登录失败

2. 初始化

调用方法

```
public void init()
```

示例

```
QuickSDK.GetInstance().init();
```

以下接口都需要在初始化成功之后调用

3. 登录

调用方法

```
public void login()
```

示例

```
QuickSDK.getInstance().login();
```

静默登录可以调用下面的方法

```
public void freeLogin()
```

4 上报角色信息

调用方法

```
public void updateRole(GameRoleInfo gameRoleInfo)
```

GameRoleInfo

字段	类型	说明
serverID	String	服务器 ID
serverName	String	服务器名称
gameRoleName	String	角色名称
gameRoleID	String	角色 ID
vipLevel	String	角色 VIP 等级
gameRoleLevel	String	角色等级

5. 支付

```
public void pay (OrderInfo orderInfo, GameRoleInfo gameRoleInfo)
```

OrderInfo

字段	类型	说明
skuType	String	商品类型，默认消耗性（inapp），订阅请设置为 subs
goodsID	String	商品 ID，用来识别购买的产品
goodsDesc	String	商品名称

字段	类型	说明
amount	double	支付金额
currency	String	支付金额单位（如 USD、CNY）
cpOrderID	String	游戏方的订单号
extrasParams	String	透传参数 请勿传特殊符号，如果无法避免建议进行 base64 编码
callbackUrl	String	支付回调地址 可选参数，可配置在后台；优先读取后台配置的回调地址

6. 退出登录

```
public void logout()
```

7. 显示个人中心

```
public void userCenter()
```

8. 获取uid

可以在登录之后再次获取Uid，如果没有登录返回空字符串

```
public String getUserId()
```

9. 销毁账号

```
//销毁账号，SDK会显示警告提示
public void userTrash()

//销毁账号，SDK不显示警告提示，游戏有警告提示时使用
public void userTrashNoUI()
```

10. 检查是否有指定的权限

```
public boolean checkHasPermission(String permission)
```

permission 权限对应的字符串，详见[Android开发者文档](#)

11. 打开谷歌Play商店评价

在mainTemplate.gradle添加依赖：

```
implementation 'com.google.android.play:review:2.0.1'
```

在需要的时候调用下面的方法：

```
public bool openReview()
```

结果会通过onOpenReviewFinish返回，重写QuickSDKListener中下面的虚方法

```
public virtual void onOpenReviewFinish(){  
    //评价界面关闭，不会返回评价是成功、失败还是取消  
}
```

三、其他接口

1. 绑定账号接口（按需接入）

如果游戏内有按钮，需要绑定第三方账号可以调用此接口

若指定类型未绑定，调用会走绑定流程；若已经绑定用户，调用该方法则为解绑账。

接收绑定结果需要重写 QuickSDKListener 中的 onBindUnbindUserSuccess 和 onBindUnbindUserFailed

```
public void bindUser(string bindType)
```

bindType 要绑定的类型，详细见下表

绑定/解绑类型	类型参数（数字字符串）
facebook	6
Google	8
Naver	9
Line	11
Email	13
VK	14
PlayGame	17

[https://developer.android.com/reference/android/Manifest.permission#constants_1]: