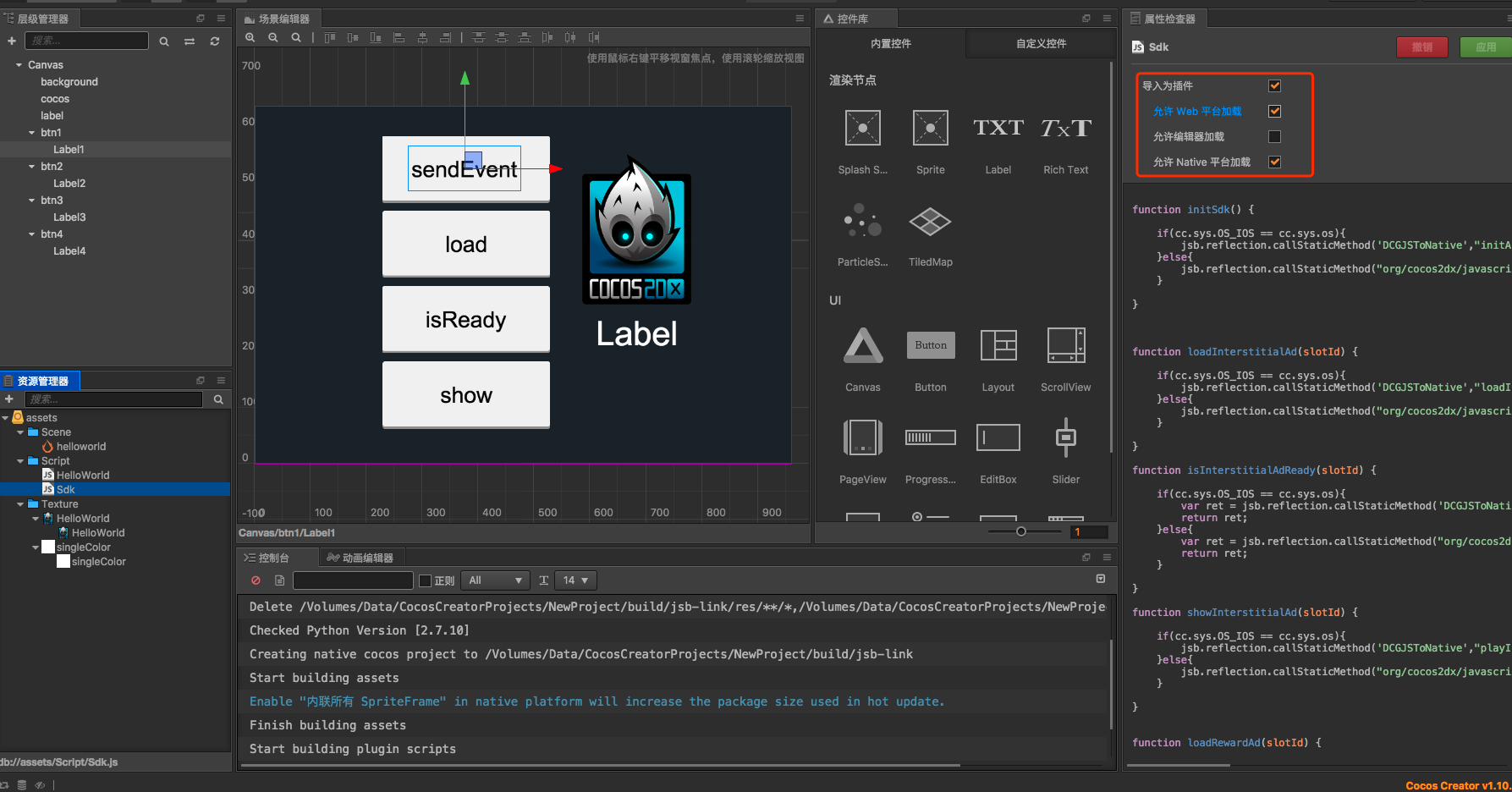
# CocosCreator Videox SDK接入文档

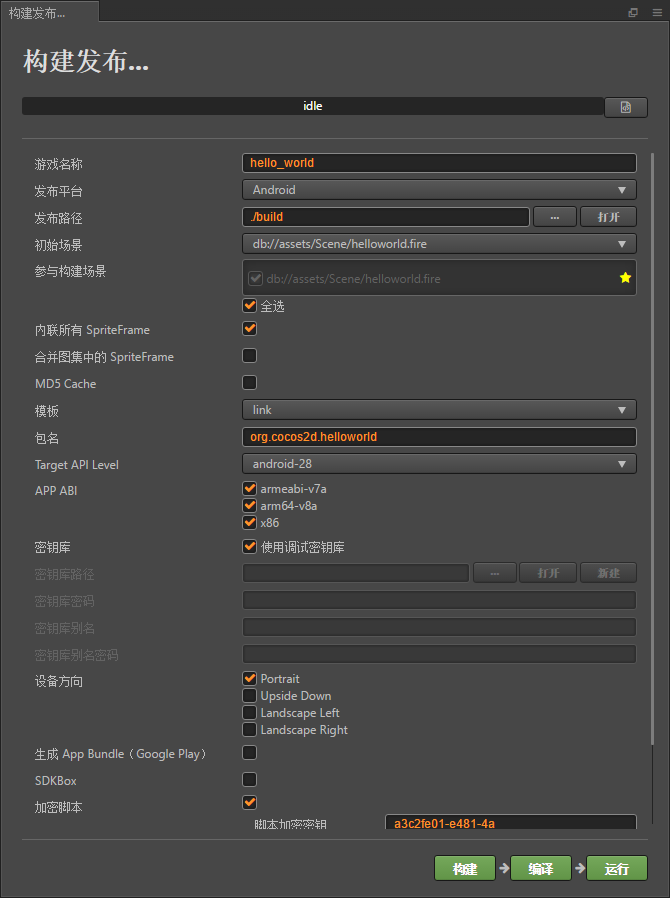
1. 在CocosCteator中正确配置Android平台的环境，以及SDK和NDK的路径。
2. 添加JS插件

解压压缩文件，将包内VideoxSdk.js文件放入CocosCreator工程文件Script目录下，如下图在属性检查器中设置为插件:



1. 构建发布CocosCreator项目

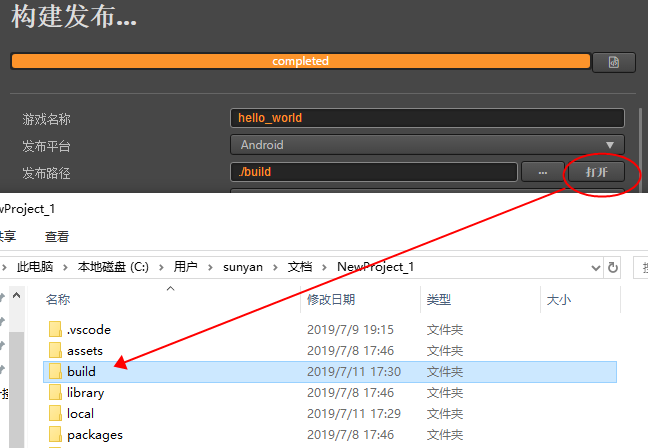
构建发布设置建议Target API level选择 android-28 ，并将APP ABI的情况都勾选（cpu32位和64位都需要支持）。



1. 生成android studio 项目

点击构建会自动生成android studio 项目，点击构建发布上的发布路径后边的打开，在build文件中找到对应的工程，用Android studio打开。

详细路径build\jsb-link\frameworks\runtime-src\\*\*.android-studio



1. 引入第三方仓库地址

为了可以引入广告平台的架包，如下图所示，在项目根目录下的build.gradle文件的 allprojects属性中添加以下代码。

代码：

repositories {  
 google()  
 jcenter()  
 maven {  
 url **"https://jitpack.io"** }  
 maven {  
 url **"https://dl.bintray.com/ironsource-mobile/android-sdk"** }  
 maven {  
 url **"https://adcolony.bintray.com/AdColony"** }  
 maven {  
 url **'http://maven.getvideox.cn/repository/videox/'** }  
}

1. 引入Videox架包和广告平台架包（国内版本）



代码：

implementation **'com.google.android.gms:play-services-basement:15.0.1'**implementation **'com.google.code.gson:gson:2.8.2'**implementation **'com.videox.sdk:release\_vc2.0.4.07:1111111'**

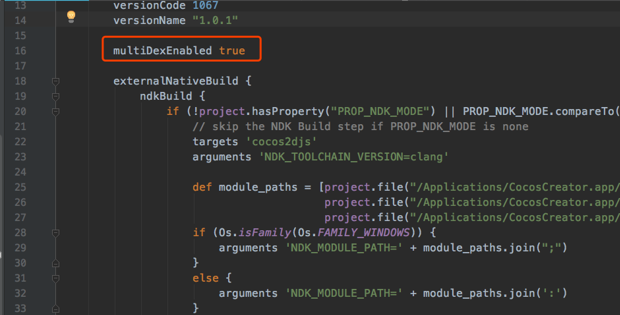
1. 引入Videox架包和广告平台架包（国际版本）



代码：

*// baselib*implementation **'com.google.code.gson:gson:2.8.2'**implementation **'com.google.android.gms:play-services-basement:15.0.1'**implementation **'com.google.android.gms:play-services-ads-identifier:15.0.1'***//Videox*implementation **'com.videox.sdk:release\_va2.0.1:11111111'**

1. 打开 /app目录下的build.gradle 需要注意以下设置

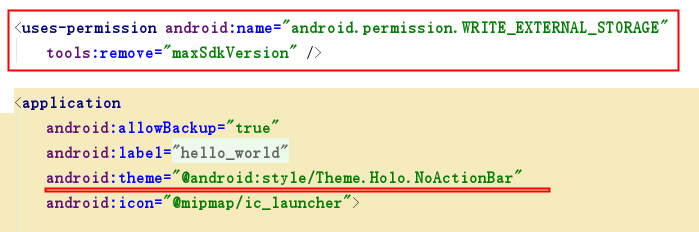


1. 如上图所示请加入 multiDexEnabled true 这行代码,对多个dex文件的支持。
2. Manifest.xml文件修改
3. 添加权限

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"  
 tools:remove="maxSdkVersion" />

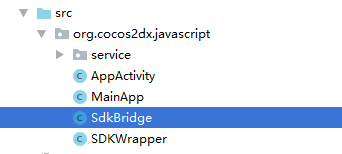
1. 修改Application的theme

android:theme="@android:style/Theme.Holo.NoActionBar"



1. 将包内 SdkBridge.java复制到指定目录

如图所示



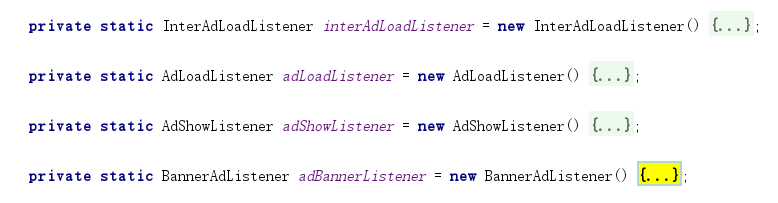
请打开AppActivity.java文件 并按图添加以下代码做广告SDK的初始化

SdkBridge.initAdSdk(this);



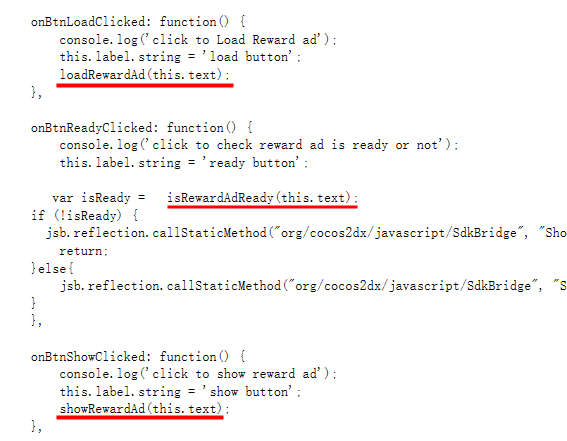
1. 广告的加载请求直接请求VideoxSDK.js插件中的方法。

具体可以参照压缩文件内的HelloWorld.js中代码。广告请求事件的回调在SDKBridge.java文件的监听接口中。



1. 激励视频广告

HelloWrold.js中的示例代码：

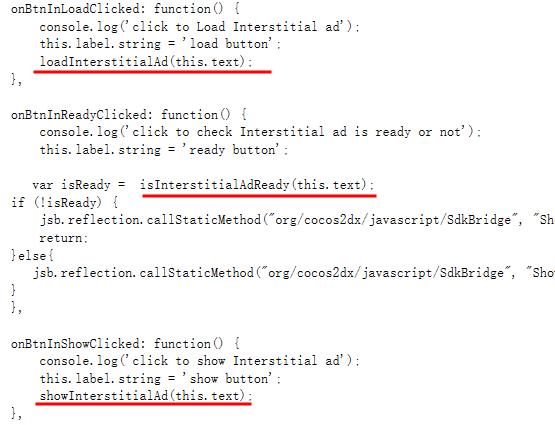


如上图所示直接调用VideoxSDK.js插件中的请求方法即可。

注意事项：

1. 每次展示之前需要做一次isRewardAdReady（”slotid”）方法的判断,只有ready好的广告才可以正常展示。
2. 为了更好的展示广告，最好在展示之前的将激励视频广告加载，在使用的时候直接判断是否ready好，如果ready好，可直接展示。
3. 插屏广告

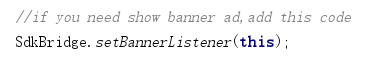
HelloWrold.js中的示例代码：

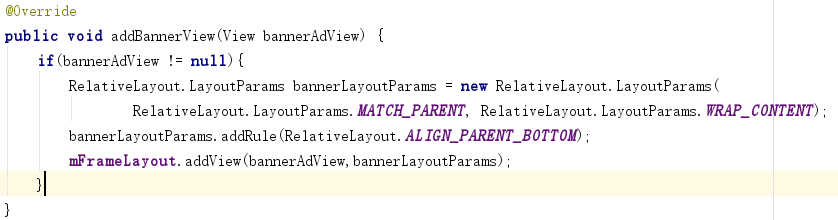


使用方式同激励视频，但是不需要提前场景加载，实时加载即可。

1. 横幅广告

因为横幅广告需要将返回的banner广告的View添加到界面中去，现在提供加载Android界面的底部功能。

1. 参照AppActivity.java文件中的方式AppActivity **implements** SdkBridge.AddBannerInterface主要用来监听banner广告是否加载成功。
2. 将当前的AddBannerInterface传递到VideoxSDK.java中，用来回调。
3. 实现接口SdkBridge.AddBannerInterface中的addBannerView（bannerView）的回调方法；参照下图实现将Banner广告置于屏幕底部



具体代码见AppActivity.java文件

1. 注意事项

（1）在激励视频调用的时候注意：

不建议在 AdLoadListener.onError 或 AdShowListener.onFinish 回调中加载广告不建议在 isReady 方法返回值为 false 的时候加载广告，SDK 内部会在 isReady 内判断广告没有准备好的情况下主动加载广告   
onFinish 回调参数 isReward ，true 表示有奖励(用户完整观看了视频)，false 表示没有奖励(用户没有完整观看视频)。是否视频时间以及可否中间退出请联系运营人员调整。

(2)如果是国内的版本出现广告无法填充，需要进行权限申请，可以参照AppActivity.java文件中的checkAndRequestPermission方法。（国际版忽略）



1. 添加混淆文件。

如果开启混淆，需要将文件夹中的[proguard.txt](https://github.com/VideoX-RewardVideoAds/videox_android_cococreator/blob/master/proguard.txt)文件下面的混淆语句，添加到app工程的Prguardproguard-rules.pro文件中。

1. 如有出现任何问题可以直接联系我们进行及时沟通。