

玉米移动广告 Android SDK

接入说明文档

V3.2.2

目录

1.概述.....	3
1.1.面向人群.....	3
1.2.开发环境.....	3
2.开发环境配置.....	3
2.1.添加 lib 文件.....	3
2.2.必要权限.....	4
2.3.注册组件.....	5
3.代码集成	5
3.1.横幅.....	5
3.2.插屏.....	6
3.3.激励视频.....	8
3.4.开屏.....	9
3.5.原生广告.....	10
4.高级功能	11
4.1.横幅.....	11
4.2.插屏.....	13
4.3.激励视频.....	14
4.4.开屏.....	15
4.5.原生广告展示说明.....	16
4.6.调试模式.....	17
4.7.Android6.0 以上系统权限处理.....	21
4.8.是否 GooglePlay 发布版本.....	22
4.9.混淆.....	22

1. 概述

1.1. 面向人群

本产品面向需要在 Android 产品中接入玉米移动广告 SDK 的开发人员。

1.2. 开发环境

OS : Windows , Mac , Linux

Android SDK : 2.3 (API level 9)

IDE : Eclipse with ADT (ADT version 23.0.4) , Android-studio

Java : jdk7

2. 开发环境配置

2.1. 添加 lib 文件

Android-studio :

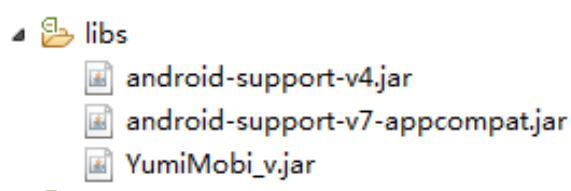
```
//确认 android studio 的 Project 根目录主 build.gradle 中配置了 jcenter 支持。
buildscript {
    repositories {
        jcenter()
    }
}
allprojects {
    repositories {
        jcenter()
    }
}
//在 module 的 build.gradle 中添加依赖
dependencies {
    compile 'com.yumimobi.ads:mediation:3.2.2.+'
}
```

Eclipse :

玉米移动广告需要的 lib 文件均放在 SDK 的 lib 文件夹下 :

- YumiMobi_vX.X.X.jar
- android-support-v4.jar
- android-support-v7-appcompat.jar
- google_play_service 的 lib 工程

使用时在工程的根目录下创建 libs 文件夹, 将 YumiMobi_vX.X.X.jar 添加到创建好的 libs 文件中。



视 需 求 添 加 android-support-v4.jar 、 android-support-v7-appcompat.jar 到 libs 文件中, 需要用到 V4 或 V7 包时必须使用我们提供的 jar。

关于 google_play_service 工程 :

google_play_service 工程非必加, 部分平台广告需要 google_play_service 支持, 玉米移动广告不需要添加。使用时需要将此工程作为 library 工程, 添加到您的工程中。并在 manifest.xml 文件的<application>标签内增加以下代码 :

```
<meta-data
    android:name="com.google.android.gms.version"
    class="kix-line-break"
    android:value="@integer/google_play_services_version" />
```

2.2. 必要权限

在工程中的 manifest.xml 中添加以下权限

必要权限：

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
```

可选权限：

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.DOWNLOAD_WITHOUT_NOTIFICATION"/>
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.WRITE_CALENDAR"/>
```

2.3. 注册组件

在工程中的 manifest.xml 文件中添加：

```
<receiver android:name="com.yumi.android.sdk.ads.self.module.receiver.ADReceiver" >
    <intent-filter>
        <action android:name="android.intent.action.DOWNLOAD_COMPLETE" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.PACKAGE_ADDED" />
        <data android:scheme="package" />
    </intent-filter>
</receiver>
<service android:name="com.yumi.android.sdk.ads.service.YumiAdsEventService" />
<activity android:name="com.yumi.android.sdk.ads.self.activity.YumiFullScreenActivity"
    android:configChanges="keyboardHidden|orientation|screenSize"
    android:theme="@android:style/Theme.NoTitleBar.Fullscreen" />
<!--Debugging Activity -->
<activity android:name="com.yumi.android.sdk.ads.mediation.activity.MediationTestActivity" ></activity>
```

3. 代码集成

3.1. 横幅

创建一个 ViewGroup 作为横幅广告容器，并添加到工程中 Activity 的适当位置。

之后调用如下代码：

```

//创建YumiBanner对象. activity是您要展示横幅的activity。 SlotID，您需要通过玉米移动平台创建一个广告位ID以
在应用中使用， auto代表使用模式是否是自动。
//auto==true 横幅广告自动轮换
//auto==false 横幅广告手动轮换，再次轮换需要重复调用banner.requestYumiBanner();
//若您在单独使用玉米广告，请开启玉米广告自动轮换，将字段置为 true。若您在通过其他聚合工具使用玉米广告，为
保证广告效果，请停止玉米广告自动轮换，将字段置为 false
banner = new YumiBanner(activity, "YOUR_SLOT_ID", auto);
//将您创建好的ViewGroup作为banner容器， 同尺寸一并设置
// bannerContainer 您的广告容器
// AdSize.BANNER_SIZE_AUTO SDK自动根据屏幕设置320*50或728*90
// isMatchWindowWidth ==true 时Banner的宽度为屏幕宽度
banner.setBannerContainer(bannerContainer, AdSize.BANNER_SIZE_AUTO, isMatchWindowWidth);
//根据您在平台的配置, 设置渠道, 您只需要设置一次渠道. 重复调用以最后一次为准.
banner.setChannelID(channelStr);
//根据您在平台的配置, 设置版本, 您只需要设置一次版本. 重复调用以最后一次为准.
banner.setVersionName(versionStr);
//开始请求广告, auto==true时此方法只需要调用一次
banner.requestYumiBanner();

```

注：ChannelID是指应用发布的渠道标示，填写后YUMI平台可根据渠道ID进行数据统计和效果分析；以Popstar!消灭星星官网正版为例，当游戏发布到三星渠道时，需要将setChannelID(channelStr)设置为setChannelID('SamSung')。

渠道标示为YUMI平台生成信息，不可随意修改；

渠道名称	ChannelID
三星	SamSung

在Activity生命周期方法中实现：

```

@Override
protected void onDestroy() {
    if (banner!= null) {
        banner.onDestroy(); //销毁banner对象
    }
    super.onDestroy();
}

```

3.2.插屏

在 Activity 的 onCreate()方法内添加以下代码：

```

//创建YumiInterstitial对象. activity是插屏展示的activity。 SlotID，您需要通过玉米移动平台创建一个广告位ID以在应
用中使用。 auto代表使用模式是否是自动。

```

```
//auto==true 插屏广告自动请求下一条，为保证广告效果建议设为自动请求
//auto==false 插屏广告不自动请求下一条，需要重复调用interstitial.requestYumiInterstitial()
//若您在使用玉米广告，为保证广告效果，请开启玉米广告自动轮换，将字段置为true。
interstitial = new YumiInterstitial(activity, "YOUR_SLOT_ID", auto);
//请根据平台的配置，设置渠道，您只需要设置一次渠道。重复调用取最后一次。
interstitial.setChannelID(channelStr);
//请根据平台的配置，设置版本，您只需要设置一次版本。重复调用取最后一次。
interstitial.setVersionName(versionStr);
//开始请求广告
interstitial.requestYumiInterstitial();
```

注：ChannelID是指应用发布的渠道标示，填写后YUMI平台可根据渠道ID进行数据统计和效果分析；以Popstar!消灭星星官网正版为例，当游戏发布到三星渠道时，需要将setChannelID(channelStr)设置为setChannelID('SamSung')。

渠道标示为YUMI平台生成信息，不可随意修改；

渠道名称	ChannelID
三星	SamSung

在需要展现插屏广告的时候，调用以下代码：

```
//立即展示 如果有预加载完成的插屏，立刻显示插屏弹窗。(弹出速度根据机型，可能有视觉延迟)，如果没有预加载完成的插屏，则不展示，并放弃此次展示机会。直到下次调用。
if (interstitial != null) {
    interstitial.showInterstitial(false);
}

//延迟展示 如果在您意图展示插屏时，预加载尚在进行，延迟展示可以允许 SDK 等待此次预加载完成，完成后自动展示插屏。等待的时间不定。
if (interstitial != null) {
    interstitial.showInterstitial(true);
}

//取消延迟展示 延迟展示时当您需要进行其他操作，而延迟展示还没进行，您需要调用以下方法取消此次延迟展示。取消后，预加载完成时不会再进行插屏展示。直到下次调用。
if (interstitial != null) {
    interstitial.cancelInterstitialDelayShown();
}
```

在Activity生命周期方法中实现：

```
@Override
protected void onDestroy() {
    if (interstitial != null) {
        interstitial.onDestroy();
    }
    super.onDestroy();
}
```

```
}
```

因为不同平台的插屏显示弹出不同. 需要在Activity的onBackPressed()方法中添加如下代码：

注意：在使用插屏时，必须增加该方法，避免back键逻辑混乱。

//在使用插屏时，必须增加此方法. 避免back键逻辑混乱.

```
if (interstitial.onBackPressed()) {
    return ;
}
super.onBackPressed();
```

3.3.激励视频

在 Activity 的 onCreate()方法内添加以下代码：

//创建YumiMedia对象. activity是您要展示插屏的activity。SlotID，您需要通过玉米移动平台创建一个广告位ID以在应用中使用。auto代表使用模式是否是自动。

```
media = new YumiMedia(activity, "YOUR_SLOT_ID");
//请根据平台的配置，设置渠道，您只需要设置一次渠道. 重复调用取最后一次.
media.setChannelID(channelStr);
//情根据平台的配置，设置版本，您只需要设置一次版本. 重复调用取最后一次.
media.setVersionName(versionStr);
//开始请求广告
media.requestYumiMedia();
```

注意：每个 YumiMedia 实例，开始请求广告的方法只需要调用一次。

ChannelID是指应用发布的渠道标示，填写后YUMI平台可根据渠道ID进行数据统计和效果分析；以Popstar!消灭星星官网正版为例，当游戏发布到三星渠道时，需要将**setChannelID(channelStr)**设置为**setChannelID('SamSung')**。

渠道标示为YUMI平台生成信息，不可随意修改；

渠道名称	ChannelID
三星	SamSung

检查视频是否准备完成，调用以下代码：

```
if (media!= null) {
    media. isMediaPrepared ();
}
```

注意：建议调用间隔 5 秒一次。

请在需要展现视频广告的时候，调用以下代码：

```
//立即展示
if (media!= null) {
```



```
media.showMedia();
}
```

注意：

1. 完成上述接入后，可以完成基本的视频接入，但是无法获取奖励回调。 请根据高级功能中状态监听部分，增加监听获取奖励回调。
2. 广告在关闭或请求失败后会自动请求下一条广告。
3. `media.requestYumiMedia()` 方法只在一开始调用一次即可。

在Activity生命周期方法中实现：

```
@Override
protected void onDestroy() {
    if (media != null) {
        media.onDestroy();
    }
    super.onDestroy();
}
```

3.4.开屏

在 Activity 的 onCreate()内添加以下代码：

```
//创建开屏对象.
//activity：是您要展示插屏的activity，
//SlotID：您需要通过玉米移动平台创建一个广告位ID以在应用中使用。
// container：广告容器
// width/height：广告容器宽高
// SplashADListener：广告回调监听
splashAD = new SplashAD(activity, SlotID, container, adwidth, adheight, SplashADListener);
```

请在 Activity 相应生命周期中调用相应方法：

```
@Override
protected void onDestroy() {
    if (splashAD != null) {
        splashAD.onDestroy();
    }
    super.onDestroy();
}
```

3.5.原生广告

在 Activity 的 onCreate()内添加以下代码：

```
// 创建一个原生广告对象，YOUR_SLOT_ID是在玉米后台申请的广告位ID
YumiNative nativeAd = new YumiNative(this, "YOUR_SLOT_ID" );
//请根据平台的配置，设置渠道，您只需要设置一次渠道。重复调用取最后一次。
nativeAd.setChannelID(channelStr);
//情根据平台的配置，设置版本，您只需要设置一次版本。重复调用取最后一次。
nativeAd.setVersionName(versionStr);
// 设置原生广告回调接口
nativeAd.setNativeEventListener(new IYumiNativeListener()
{
    @Override
    public void onLayerPrepared(int adCount)
    {
        // 请求成功的回调，其中adCount是返回的广告条数
    }
    @Override
    public void onLayerFailed(LayerErrorCode error)
    {
        // 请求失败的回调，其中error是请求失败的错误提示
    }
    @Override
    public void onLayerClick() {
        //广告点击回调
    }
});
// 请求广告，成功或失败的结果会在回调接口中返回
nativeAd.requestYumiNative();
```

注：ChannelID是指应用发布的渠道标示，填写后YUMI平台可根据渠道ID进行数据统计和效果分析；以Popstar!消灭星星官网正版为例，当游戏发布到三星渠道时，需要将setChannelID(channelStr)设置为setChannelID('SamSung')。

渠道标示为YUMI平台生成信息，不可随意修改；

渠道名称	ChannelID
三星	SamSung

在 Activity 生命周期 onDestroy()方法中，增加对应的 SDK 生命周期方法:

```

@Override
protected void onDestroy()
{
    super.onDestroy();
    if (nativeAd!=null)
    {
        nativeAd.onDestroy();
    }
}

```

4.高级功能

4.1.横幅

4.1.1.设置广告状态监听

如果需要监听横幅广告生命周期，请在创建 YumiBanner 对象后，调用如下方法：

```

//设置banner状态监听.
banner.setBannerEventListener(bannerListener);

```

关于广告监听，您可以直接实例化一个 IYumiBannerListener，并根据回调增加您自己的逻辑。监听有以下回调方法：

onBannerPreparedFailed(LayerErrorCode errorCode)	当横幅加载失败时回调，可以通过 errorCode.getMsg()获得失败原因。
onBannerPrepared()	当横幅加载成功时回调
onBannerExposure()	当横幅展示成功时回调
onBannerClosed()	当横幅关闭时回调
onBannerClicked()	当横幅点击时回调

接入示例：

```

//创建banner状态监听
bannerListener = new IYumiBannerListener() {
    @Override

```

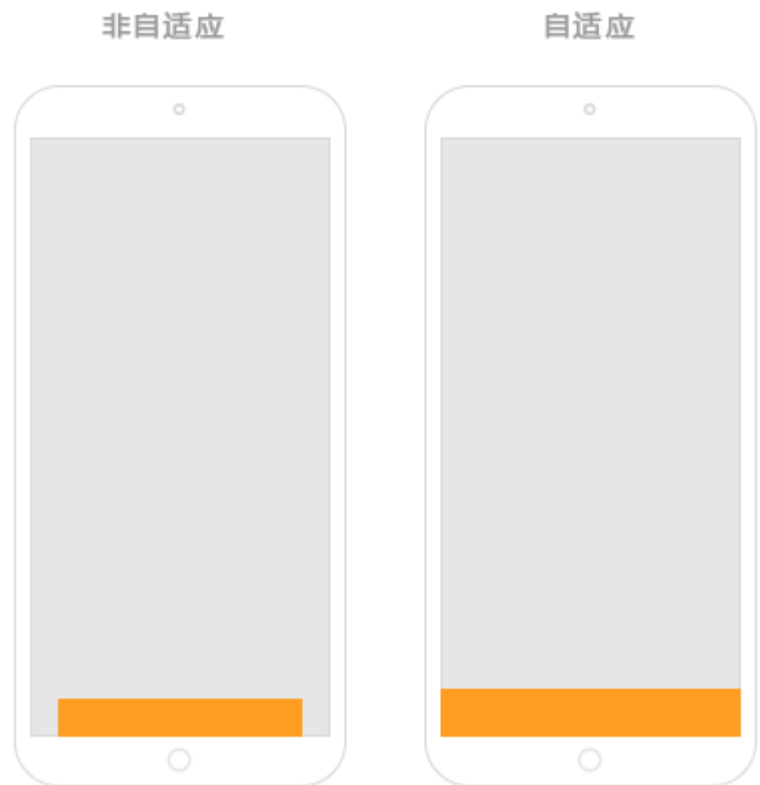
```
public void onBannerPreparedFailed(LayerErrorCode errorCode) {  
    //当横幅加载失败时回调. 可以通过errorCode.getMsg()获得失败原因.  
}  
@Override  
public void onBannerPrepared() {  
    //当横幅加载成功时回调.  
}  
@Override  
public void onBannerExposure() {  
    //当横幅展示成功时回调.  
}  
@Override  
public void onBannerClosed() {  
    //当横幅关闭时回调.  
}  
@Override  
public void onBannerClicked() {  
    //当横幅点击时回调.  
}  
};
```

4.1.2.横幅显示和隐藏

通过以下方法隐藏和恢复显示横幅：

```
//隐藏横幅， 同时暂停横幅轮换请求  
banner.dismissBanner();  
//恢复显示横幅， 同时恢复横幅轮换请求  
banner.resumeBanner();
```

4.1.3. 横幅自适应屏幕宽度



横幅设置广告容器的同时，玉米SDK提供了 `boolean` 类型的参数：`isMatchWindowWidth`。该参数为横幅宽度是否充满全屏，当该参数为`true`时横幅的宽度为屏幕宽度。

4.2. 插屏

设置广告状态监听

如果您需要监听插屏广告生命周期，请在创建 `YumiInterstitial` 对象后，调用如下方法：

```
//设置插屏状态监听。  
interstitial.setInterstitialEventListener(interstitialListener);
```

关于广告监听，您可以直接实例化一个 `IYumiInterstitialListener`，并根据回调增加您自己的逻辑。监听有以下回调方法：

<code>onInterstitialPreparedFailed(LayerErrorCode error)</code>	当插屏加载失败时回调，可以通过 <code>errorCode.getMsg()</code> 获得失败原因。
<code>onInterstitialPrepared()</code>	当插屏加载成功时回调。

	注意：请不要直接在此回调中调用插屏的展示方法。
<code>onInterstitialExposure()</code>	当插屏展示成功时回调
<code>onInterstitialClosed()</code>	当插屏关闭时回调
<code>onInterstitialClicked()</code>	当插屏点击时回调

接入示例：

```
//创建插屏状态监听
interstitialListener = new IYumiInterstitialListener() {
    @Override
    public void onInterstitialPreparedFailed(LayerErrorCode error) {
        //当插屏加载失败时回调. 可以通过errorCode.getMsg()获得失败原因.
    }
    @Override
    public void onInterstitialPrepared() {
        //当插屏加载成功时回调.
    }
    @Override
    public void onInterstitialExposure() {
        //当插屏展示成功时回调.
    }
    @Override
    public void onInterstitialClosed() {
        //当插屏关闭时回调.
    }
    @Override
    public void onInterstitialClicked() {
        //当插屏点击时回调.
    }
};
```

4.3.激励视频

设置广告状态监听

如果您需要监听视频广告生命周期，请在创建 YumiMedia 对象后，调用如下方法：

```
//设置视频状态监听.
media.setMediaEventListener(mediaListener);
```

关于广告监听，您可以直接实例化一个 `IYumiMediaListener`，并根据回调增加您自己的逻辑。监听有以下回调方法：

<code>onMediaExposure()</code>	当激励视频展示成功时回调
<code>onMediaClosed()</code>	当激励视频关闭时回调
<code>onMediaClicked()</code>	当激励视频点击时回调
<code>onMediaIncentived()</code>	当激励视频播放完成， 可以获取奖励时回调。 注意：如果未完成播放，不会回调次方法。另外，此方法永远在 <code>onMediaClosed()</code> 之后触发。

接入示例：

```
//创建视频状态监听
mediaListener = new IYumiMediaListener() {
    @Override
    public void onMediaIncentived() {
        //当激励视频获得奖励时回调
    }
    @Override
    public void onMediaExposure() {
        //当激励视频展示成功时回调.
    }
    @Override
    public void onMediaClosed() {
        //当激励视频关闭时回调.
    }
    @Override
    public void onMediaClicked() {
        //当激励视频点击时回调.
    }
};
```

4.4.开屏

设置广告状态监听

关于广告监听，您可以直接实例化一个 `SplashADListener`，并根据回调增加您自己的逻辑。监听有以下回调方法：

onSplashShow()	当开屏展示时回调
onSplashClose()	当开屏关闭时回调
onSplashClick()	当开屏点击时回调
onSplashFailed()	当开屏加载失败时回调

接入示例：

```
//创建视频状态监听
splashListener = new SplashADListener () {
    @Override
    public void onSplashShow () {
        //当开屏展示时回调
    }
    @Override
    public void onSplashFailed () {
        //当开屏加载失败时回调
    }
    @Override
    public void onSplashClose () {
        //当开屏关闭时回调
    }
    @Override
    public void onSplashClick () {
        //当开屏点击时回调
    }
};
```

4.5.原生广告展示说明

如果您需要使用原生广告，请参考如下方法：

```
if (nativeAd.getADCount() > 0) //通过剩余广告条数判断是否存在下一条广告
{
    final NativeContent content = nativeAd.nextContent(); //提取下一条广告
    content.getDesc(); //获取广告详情文字
    content.getIcon_url(); //获取icon的url
    content.getImg_url(); //获取广告图片的url
    content.getImg_height(); //获取广告图片的宽,获取不到时为0
    content.getImg_width(); //获取广告图片的高,获取不到时为0
    content.getJumpUrl(); //获取广告点击跳转的url
}
```



```

content.getTitle();//获取广告的标题文字
content.getButtonText();// 行动语(查看详情/下载)
content.getPrice();//价格(免费/$6.3)
content.getOther();//其他(2017-09-18 更新)
}

```

为了保证您的收益，请您在相应的位置调用以下相应的广告上报方法（重要）

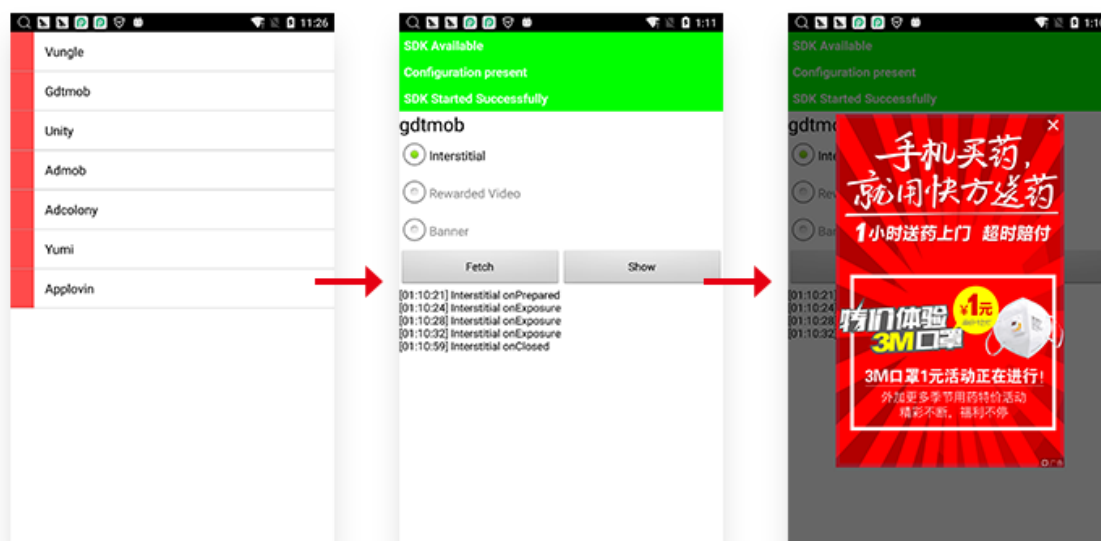
```

content.reportShow(container,content); //广告展示时上报(container为广告父布局)
content.reportClick(container,content); //广告展示时上报(container为广告父布局)

```

4.6.调试模式

玉米广告 SDK 为开发者提供了一个检测三方平台集成状态的调试模式，如图：



使用步骤：

1、调用方法打开调试页面：

```
YumiSettings.startDebugging(Activity,BannerSlotID,InterstitialSlotID,MediaSlotID);
```

根据您在平台的配置需要设置渠道、版本：

```
YumiSettings.startDebugging(Activity, BannerSlotID,InterstitialSlotID,MediaSlotID, channelId, versionName);
```

2、玉米 SDK 将获取配置并展示三方平台列表，进入 debug 页面：

1) 页面展示为 Searching for third party ADnetwork adapters：表示未进行配置，请检查应用的不同广告形式配置情况，如果问题仍未解决，请通过邮件联系我们：support@yumimobi.com

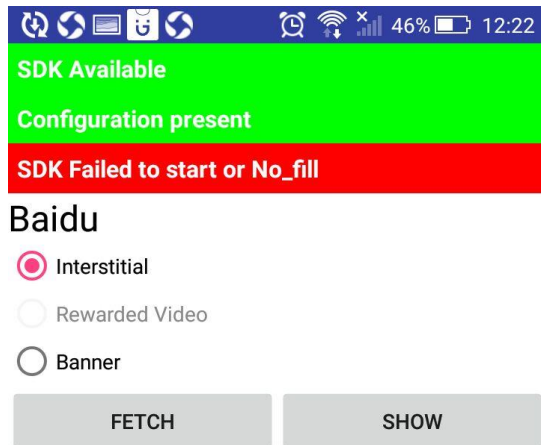


2) 广告配置后，正常展示配置平台，首次进入左侧所有平台均为红色，当某个平台正确接入并展示成功后，左侧状态为绿色。

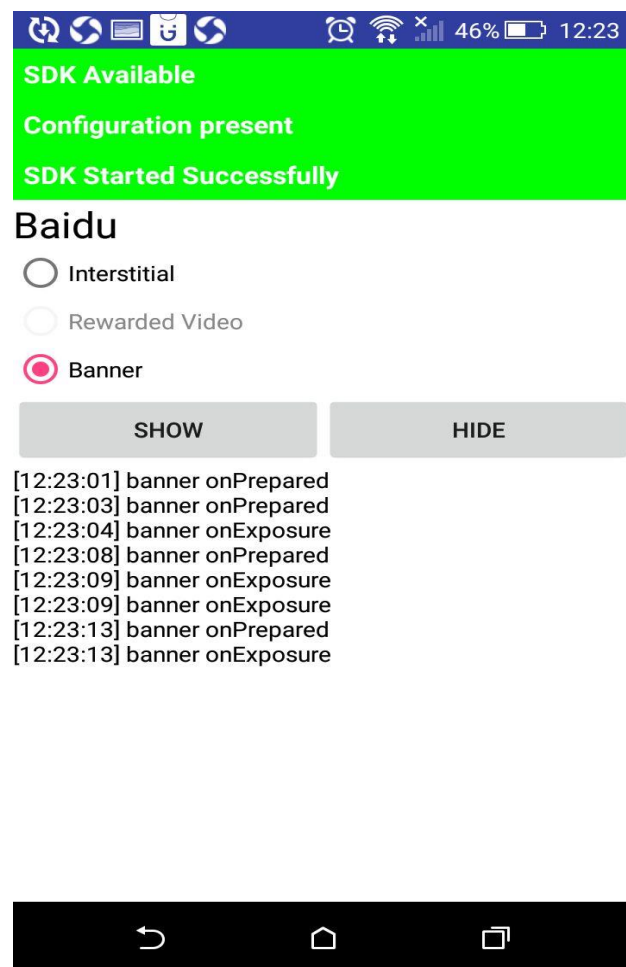


3、无论左侧状态栏颜色为何状态，均可选择一家平台点击进入：

- 1) SDK Available 为绿色时表示三方平台适配器已添加；当为红色时表示三方平台适配器未添加，回到 3.1-2 检查该平台 adapter 是否添加
- 2) Configuration present 为绿色时表示三方平台适配器组件 Manifest 已注册；当为红色时表示三方平台适配器组件 Manifest 未注册，可回到 3.3 检查该平台适配器组件是否添加
- 3) SDK Failed to start or No_fill 为绿色表示广告曾经展示成功过；当为红色时表示还未展示成功过广告，可继续进行下一个步骤，如果所有步骤完成后仍未红色，请邮件联系我们 support@yumimobi.com



- 4、点击 Fetch 开始请求广告，点击 Show 展示广告
- 5、广告展示成功后检查项全部变为绿色，表示该平台接入成功



6、应用发布前需要将调试模式注释掉。

4.7.Android6.0 以上系统权限处理

当您的应用 targetSdkVersion 为 23 及以上时，可选择以下方法进行权限检查并且弹窗提示用户授权。

注：该方法默认为 false，不会对用户进行权限提示并且不会导致崩溃。设为 true，会进行权限检查并且弹窗提示用户授权。该方法在实例化广告之前调用，并且需要添加 android-support-v4.jar。

```
YumiSettings.runInCheckPermission(true);
```

4.8.是否 GooglePlay 发布版本

注：如果您的 APP 是在 GooglePlay 发布，请设置以下方法。

```
YumiSettings. setApplsGooglePlayVersions (true);
```

4.9.混淆

如果您的工程需要混淆编译，请在混淆文件内增加以下内容。

```
-keepattributes Exceptions , InnerClasses , Signature , Deprecated ,  
SourceFile ,LineNumberTable ,*Annotation* ,Synthetic ,EnclosingMethod  
-keep class com.yumi.android.sdk.ads.** { *;}  
-keep class com.yumi.android.sdk.ads.self.**{*;}  
-keep class com.yumi.android.sdk.ads.selfmedia.**{*;}
```