

有赞原生SDK5.1.0 原生UI接入教程

1. 简述

Android端的 **原生SDK** 相当于 **基础SDK** 的扩展包, 在Html5浏览页面的基础上可以实现部分页面的原生化展示, 现已支持商品页的原生化, 并会在后续更新升级中不断优化体验.

2. 快速体验

开发者通过demo来快速了解SDK的使用, 点击 [【下载demo】](#).

3. 运行环境

3.1 依赖

- 图片加载框架

SDK内需要使用图片加载框架, 现SDK内已自动适配的框架如下:

图片加载框架	兼容版本	适配器
Picasso	2.5.2	PicassoFactory
UniversalImageLoader	1.9.5	UILFactory
Glide	3.3.0+	GlideFactory
Fresco	0.8.0+	FrescoFactory
Volley	1.0.0+	VolleyFactory

如果项目之前没有接入过图片加载框架的话, 推荐使用[Picasso](#)

开发者只需要选择上面的一种图片加载框架在项目中导入即可, SDK内部会自动适配(也可以主动设置具体看 **常见问题6.2**), 无需另外编写代码.

- Android兼容包
[Android support v4](#)
- 网络请求框架
[okhttp3+](#), [okio](#)

4. 接入

在 项目的根build.gradle中声明maven仓库.
如下:

```
allprojects {  
    repositories {  
        maven {url 'https://dl.bintray.com/youzanyun/maven/'}  
    }  
}
```

在子项目build.gradle的dependencies中声明包:

```
compile 'com.youzanyun.open.mobile:basic:5.1.0'
```

有赞 原生SDK 依托于SDK中提供的 YouzanHybrid 来进行页面渲染.

```
public class YouzanActivity extends Activity{  
    private YouzanHybrid view;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        YouzanHybrid view = new YouzanHybrid(context);  
        setContentView(view);  
        view.loadUrl("入口链接");  
    }  
}
```

`YouzanHybrid` 需要使用单独的一个Activity来展示.

其他关于原生SDK的API和配置, 与基础SDK相类似, 这里就不赘述. 了解更多请查看[DEMO](#).

5. 其他

5.1 独立显示商品购买弹框



如果想对单个商品进行快速操作(加入购物车, 立即购买), 可以使用SDK内提供的 `GoodsCardDialog` 类. 构建Dialog代码如下:

```
new GoodsCardDialog.Builder(this, YouzanActivity.class)
    .setAlias(alias)
    .build(new Callback() {
        @Override
        public void onCreate(Dialog dialog) {
            //获取商品购买弹框依附的dialog, 开发者可以基于此进行操作.
            dialog.show();
        }
    });
```

其中 `YouzanActivity.class` 为 `YouzanHybrid` 所依附的Activity的类对象, 并在回调的 `onCreated` 方法中获取到Dialog实例.

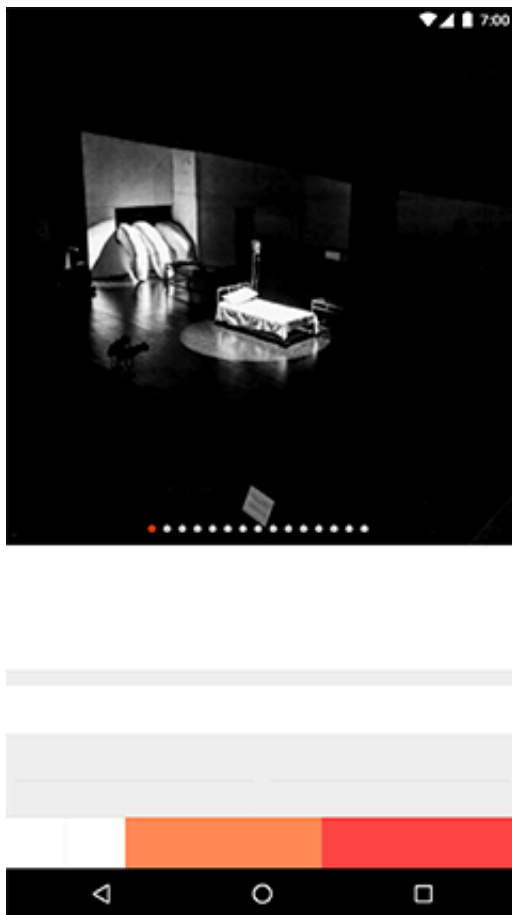
方法	说明	是否必需
setAlias(String)	商品数字编号	
setNumlid(long)	商品别名	调用时,参数 numlid 和 alias 必须选其一

6. 常见问题

6.1 原生SDK找不到本地资源文件

错误描述:

开发在项目中集成SDK后有时会出现找不到本地字符串和图标资源, 比如商品详情效果如图:



解决方法:

确认已经将相应的资源文件拷贝到项目中, 项目开启了代码混淆的话需要保持 `R` 类不被混淆.

另外, 如果项目在 `AndroidManifest.xml` 中定义的包名和 `build.gradle` 中定义的 `applicationId` 不一致的话, 也会造成同样的问题, 需要将 `AndroidManifest.xml` 中定义的包名传给SDK:

```
YouzanHybrid.setManifestPackageName("com.abc.sample");
```

6.2 无法展示图片

错误描述:

在确定无网络问题且确定项目中已经集成相应的图片加载框架的情况下, 无法显示页面内的图片, 效果如图:



解决方法:

这个问题一般是由于代码混淆造成的SDK无法找到图片加载框架, 解决方案有两种.

一种是修改项目的代码混淆策略, 如下:

```
# Resource class R, suggest to keep class R in package of {applicationId} only.
-keep class **.R$* {*; }
-keep class **.R{*; }

# Youzan SDK
-dontwarn com.youzan.sdk.***
-keep class com.youzan.sdk.**{*; }

# OkHttp
-dontwarn okio.**
-dontwarn com.squareup.okhttp.**
-keep class okio.**{*; }
-keep class com.squareup.okhttp.** { *; }
-keep interface com.squareup.okhttp.** { *; }

-dontwarn java.nio.file.*
-dontwarn javax.annotation.**
-dontwarn org.codehaus.mojo.animal_sniffer.IgnoreJRERequirement

# Image Loader
-keep class com.squareup.picasso.Picasso
-keep class com.android.volley.toolbox.Volley
-keep class com.bumptech.glide.Glide
-keep class com.nostra13.universalimageloader.core.ImageLoader
-keep class com.facebook.drawee.backends.pipeline.Fresco
```

另一种是主动为SDK设定加载框架的适配器, 需要在展示 `YouzanHybrid` 之前调用:

```
YouzanHybrid.setImageLoaderFactory(new PicassoFactory());
```

其中 `PicassoFactory` 为Picasso的适配器, 其他适配器还有: `FrescoFactory`, `GlideFactory`, `UILFactory`, `VolleyFactory` 等.

6.3 找不到constraint layout相关资源

错误类似:

```
AAPT: No resource identifier found for attribute
'layout_constraintLeft_toLeftOf'
```

请在子项目中build.gradle中, 引入:

```
repositories {  
    maven {  
        url 'https://maven.google.com'  
    }  
}
```