CSDN首页 (http://www.csdn.net?ref=toolhar)

学院 (http://edu.csdn.net?ref=toolbar)

下载 (http://download.csdn.net?ref=toolbar)

□ 下载 CSDN APP (http://www.csdn.net/app/?ref=toolbar)

✓ 写博客 (http://write.blog.csdn.net/postedit?ref=toolbar)

全部

ocx-\_- ▼ (http://my.csdn.net?ref=toolbar)

首页 (http://blog.csdn.net/) cson (http://www.csdn.net)

# Android移植NDK子题目--以android-gif-drawable为例

2016年06月21日 17:43:28

**2304** 

### 前言:

一些非常优秀的开源项目引入起来都是比较麻烦的,麻烦到都可以写一篇博客来引导后来人少踩坑的程 度。我在最近一次项目中想要做一个GIF的展示功能,光是引入著名的gifhub开源项目android-gifdrawable (https://github.com/koral--/android-gif-drawable)就花费了一番功夫,今天把我踩过的坑都 记录一下,提醒一下准备使用这个开源库的盆友。

GIF三方库引入方式:

方式一: 直接添加dependence

这种方式在Android studio中最简单的一种方式,只需打开build.gradle文件,加入这一句话即可。一般我 引入三方库都直接使用这样的方法,简便快捷,仅需一行就完成了依赖的部署,然后clean一下项目就可以 使用依赖库里的类和方法了。

```
[html]
1.
    dependencies {
2.
         compile 'pl.droidsonroids.gif:android-gif-drawable:1.1.+'
```

但是我这样引入以后再运行阶段出现一个crash 异常,如下:

```
AndroidRuntime: FATAL EXCEPTION: DBExecutor #1
 1.
 2.
              Process: com.demo.app, PID: 15616
 3.
              java.lang.UnsatisfiedLinkError: dalvik.system.PathClassLoader[DexPathList[[zip file "/s
      2/base.apk"], nativeLibraryDirectories=[/data/app/com.demo.app-
      2/lib/arm64, /vendor/lib64, /system/lib64]]] couldn't find "xxxx.so"
              at java.lang.Runtime.loadLibrary(Runtime.java:366)
 5.
              at java.lang.System.loadLibrary(System.java:988)
 6.
              at com.demo.utils.JConstantUtils.<clinit>(Unknown Source)
              at com.demo.handler.bk.<init>(Unknown Source)
 8.
              at com.demo.handler.bi.f(Unknown Source)
9.
              at com.demo.handler.aq.a(Unknown Source)
10.
              at com.demo.handler.aq.a(Unknown Source)
11.
              at com.demo.handler.io.a(Unknown Source)
12.
              at com.demo.handler.io.a(Unknown Source)
13.
              at com.demo.handler.run(Unknown Source)
14.
              at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1112)
15.
              \verb|at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:587)|\\
16
              at java.lang.Thread.run(Thread.java:818)
```

从字面意思上来看,是在执行System.load()方法的时候没有找到对应的.so文件,然后我解压apk安装包, 发现该so文件还在armeabi文件夹和x86文件夹下。



lvshaorong (http://blog....

Q

+关注

(http://blog.csdn.net/lvshaorong) 码云

原创 粉丝 喜欢 (https://g 131 utm\_sour

#### 他的最新文章

更多文章 (http://blog.csdn.net/lvshaorong)

Caddy轻松让HTTP端口实现TLS加密 (/lvshaorong/article/details/78065178)

Oracle Database 12c 数据库简单实例 的创建过程

(/lvshaorong/article/details/78048823)

NodeJS为邮件服务器打造远程反病毒 扫描服务

(/lvshaorong/article/details/78003936)

RedHat7 LVM分区方案部署和扩容方法 (/lvshaorong/article/details/77945125)

### 女生丰胸的方法









#### 编辑推荐

最热专栏

两个gif图片处理库使用 (/yehuohan/ar…

SimpleGif,一款使用方法类似Glide但更…

记一次Android开发GIF功能踩过的坑 (/…

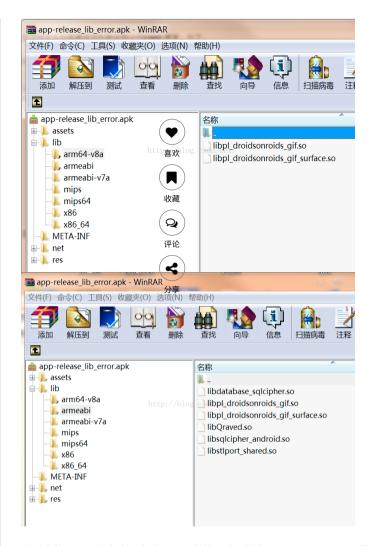
Android加载网络GIF完整解决方案 (/lv···

Android—行代码实现网络加载GIF闪图…

在线课程



返回顶部



但是我的项目里却多出了多个CPU平台的so库,比如arm64\_v8a,x86\_64等,根据上面的报错异常可以看出,安卓系统找不到arm64下的.so文件,因为arm64\_v8a文件夹全都是这个gif框架的so库,缺少本项目的so库。那么问题就显而易见了:

Android在加载时会根据当前cpu型号自动选择相应的so文件夹,然后加载里面的so库,如果我的apk里面有arm64\_v8a文件夹,我使用的又是Android5.0 64位CPU,但文件夹里面又没有我要加载的so库,即使armeabi文件夹里有也不管用,就会出现这个异常。

这个问题的坑人之处在于如果apk包里没有arm64\_v8a文件夹,系统会自动使用armeabi文件夹的so库顶上,也就是使用32位的方法兼容64位CPU,但是Android Studio在编译项目时将第三方库的64位so编译进了apk,而项目原本的so只有armeabi版本,所以导致加载so库失败。

要处理这个异常有两个途径,第一是使用ndk r11以上版本将本项目的so库编译出arm64\_v8a的so,misp的so,x86\_64等等so库,注意只有NDK r11以上版本才支持64位的编译,r10是在Android5.0发布之前发布的,安卓是在5.0之后才开始支持64位CPU,所以r10是不能编译arm和x86 64位版本的。

第一个途径有一个明显的弊端是so库使用的太多,会造成app的体积过大,而且如果原项目中依赖了其他的库,比如xUtils,它只提供了armeabi和x86的so,真是令人瞬间傻眼,所以就有了第二个解决途径。 方式二:使用aar包

第二类途径就是删掉一部分本项目没有而gif第三方库有的so,让两边的so库数量相同,这样就不会出现缺少某一CPU平台的问题了。该方案需要用到aar包。aar包可以看成是添加了相关图片资源文件的jar包,这个gif第三方库在github上也提供了它的arr包下载,如果你下载不成功的话,可以去github上下载该项目的源码,使用android studio打开,clean一下项目。然后到:项目根目录\build\outputs\aar文件夹下,取得android-gif-drawable-master-debug.aar,打开后如下图:







现在只需要在这个aar文件中将<del>ag</del>neabi,x86以外的文件夹删掉,就可以避免在编译时将不需要的so文件 打入apk了。

那么如何将arr导入自己的项目作为依赖,网上有很多方法,下面说一种:

这里演示的aar文件为: " genius.aar "

第一步:拷贝到: libs目录

```
第二步: build.gradle 配置文件中更改为
repositories {
    flatDir {
        dirs 'libs'
    }
}
dependencies {
    compile(name:'genius', ext:'aar')
```

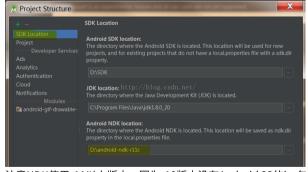
摘自: http://www.tuicool.com/articles/V7reai

这样就可以请可以轻松的将一个外部NDK项目导入到自己的项目中使用了。

但是如果我想把外部NDK项目的c/c++源码一并导入进来并进行修改,过程就比较复杂了

方式三:将第三方库的源码全部导入自己的项目 将外部NDK项目完全导入自己的项目步骤:

- 1、去github上下载项目zip源码
- 2、解压zip,并使用Android Studio打开该项目,就像打开普通项目一样,点击file--project stucture--设置NDK的路径,如下



注意NDK使用r11以上版本,因为r10版本没有Android 23的jar包

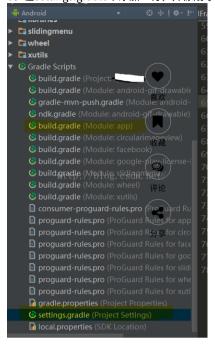
- 3、使用AndroidStudio 把这个项目clean一下
- 4、将第三方项目拷贝到自己项目的libraries文件夹下
- 5、使用Android Studio打开自己的项目,在build.gradle中添加如下内容(android-gif-drawable 是 libraries 里面的文件夹的名字)

```
[html]
1. dependencies {
2.    compile fileTree(include: ['*.jar'], dir: 'libs')
3.    compile project(':libraries:android-gif-drawable')
4.
```



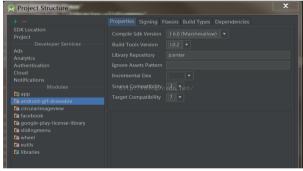
5. }

6、在settings.gradle中添加一行,并像刚才子项目一样添加NDK的依赖



#### [html]

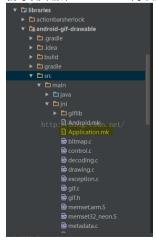
- . include ':libraries:android-gif-drawable'
- 7、然后clean一下当前项目。然后进入点击file--project structure查看外部项目是否已经导入进了项目目录,如下



如果导入成功,那么就可以看下面的部分,如果导入不成功,一般是缺少build-tools等sdk的依赖,一般翻墙下载就能解决,或者在build.gradle修改min sdk version, build tools version等.

8、修改子项目的so库编译选项

进入gif子项目的jni目录,如下,修改Application.mk中的APP\_ABI:all为APP\_ABI := armeabi x86,这样就可以只编译armeabi x86两个版本的so库,就可以和项目里其他的so库匹配起来了。





#### 9、去掉子项目编译前的单元测试

我在使用Android Studio编译整个项目时,并由执行gif的单元测试,但是我在使用命令行gradle build命令时,会首先执行gif子项目的单元测试,一共21个测试项目,如果失败任何一个编译过程就会终止。所以需要关掉单元测试,方法是修改build.gradle文件并删除相关测试文件,下面gif依赖库的build.gradle注释掉的内容即为和单元测试有关的内容:

```
[html]
 1.
     buildscript {
 2.
         repositories {
 3.
             jcenter()
 5.
 6.
         dependencies {
                          收藏
 7.
             classpath 'com_android.tools.build:gradle:2.1.0'
             //classpath 'com dicedmelon.gradle:jacoco-android:0.1.1'//不进行代码覆盖率的测试
 8.
 9.
                          评论
10.
     }
11.
     apply plugin: 'com.and d.library'
12.
     //apply plugin: 'jacoco-android'//不进行代码覆盖率的测试
13.
     apply from: 'gradle-mv种事ush.gradle'
14.
     apply from: 'ndk.gradle'
15.
16.
17.
     repositories {
18.
         jcenter()
19.
20.
21.
     project.version = VERSION NAME
22.
     project.group = GROUP
23.
24.
     android {
         compileSdkVersion 23
25.
26.
         buildToolsVersion '23.0.2'//如果编译器说build tools version本机sdk没有,就选择一个本机sdk里有的版
27.
         compileOptions {
             sourceCompatibility JavaVersion.VERSION 1 7
28.
29.
             targetCompatibility JavaVersion.VERSION_1_7
30.
         defaultConfig {
32.
             versionName project.version
33.
             minSdkVersion 8
34.
             targetSdkVersion 23
35.
             consumerProguardFiles 'consumer-proguard-rules.pro'
36.
         }
37.
     }
38.
39.
     dependencies {
         provided 'com.android.support:support-annotations:23.3.0'//如果这里报错请修改版本号并翻墙,下面
40.
     // testCompile 'junit:junit:4.12'//在编译时会进行单元测试,大约21个,如果有任何一个测试没有通过就会终止
41.
     编译,所以将所有单元测试去掉
42.
     // testCompile 'org.mockito:mockito-core:1.10.19'
     // testCompile 'org.robolectric:robolectric:3.0'
     // testCompile 'org.assertj:assertj-core:1.7.1'
44.
     // testCompile 'net.jodah:concurrentunit:0.4.2'
45.
46.
47.
     //jacocoAndroidUnitTestReport {//不进行代码覆盖率的测试
48.
49.
     // html.enabled true
     // xml.enabled true
50.
51.
     //}
```

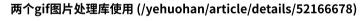
删除相关单元测试的java文件,不然会报编译错误,如下:







### 相关文章推荐



giflibgiflib是一个gif图片的处理库,可用于gif格式图片的生成与解码,在http://giflib.sourceforge.net/有详细的介绍。 在ht tp://download.c...

yehuohan (http://blog.csdn.net/yehuohan) 2016-08-09 22:24

**1458** 

## SimpleGif,一款使用方法类似Glide但更省cpu、memory的Android ndk gif解码库,cmake方 式, android studio项目 (/qq372848728/article/details/59104104)

Gif文件是由多张静态图片组合而成,每张图片都有自己的显示时间,屏幕通过连续显示这些图片,以达到动态图的目的,由 于这些特性,导致如果用java去解码,可能效率不是那么高,因为java能运行速度始终无法...

gq372848728 (http://blog.csdn.net/qq372848728) 2017-03-01 14:00 QQ 415



## 我是如何成为一名python大咖的?(http://edu.csdn.net/topic/python1?ut m\_source=blog10)

人生苦短,都说必须python,那么我分享下我是如何从小白成为Python资深开发者的吧。2014年我

记一次Android开发GIF功能踩过的坑 (/hys19920320/article/details/44904451)



相信不少人都知道listview中view的重用的,也会用holder和tag提高list的效率,但是我如果问你view什么时候销毁你还知道 吗?

nys19920320 (http://blog.csdn.net/hys19920320) 2015-04-06 19:05 🚨 1828

## Android加载网络GIF完整解决方案 (/lvshaorong/article/details/51732520)

本文主要介绍了在Android插入GIF动画的思路,主要包括文件缓存的方案,防止重复下载的方法,下载进度显示方案,防止 内存泄漏的方法,与ViewPager适配**商**务法等。



lvshaorong (http://blog.csdn/net/lvshaorong) 2016-06-23 14:10 \$\square\$ 8179

## Android一行代码实现网络加载GIF闪图(附源码) (/qq\_24515359/article/details/51900125)

最近项目有个需求是要从网络加载(Ji Mal Bi,但是Android原生的ImageView并不支持Gif。。。于是从网上看了些Dome,发 现总是有些这样那样的问题,譬如: 4年没有缓存,还要自...



🌔 qq\_24515359 (http://blog.csdn.net/qq\_24515359) 2016-07-13 18:02 🚨 1555

## 关于图片加载神器--Glide与Picasso的使用与比较 (/u011200604/article/details/53021685)

什么是Glide? Glide是一个加载图片的库,作者是bumptech,它是在泰国举行的google 开发者论坛上google为我们介绍 的,这个库被广泛的运用在google的开源项目中。...



形 u011200604 (http://blog.csdn.net/u011200604) 2016-11-03 15:07 🛚 🕮 2501

### Android图片加载框架比较----Glide, Picasso, Fresco (/hong geek/article/details/49849339)

Glide 添加依赖: dependencies { compile 'com.github.bumptech.glide:glide:3.5.2' compile 'com.and...

## Glide 加载Gif 导致cpu居高不下的解决办法 (/lhj\_android/article/details/60139216)

glide 加载gif cpu占用高的解决办法

lhi\_android (http://blog.csdn.net/lhi\_android) 2017-03-03 14:01

## Android加载Gif和ImageView的通用解决方案: android-gif-drawable (/u010983763/article/details/64133968)

Android加载Gif和ImageView的通用解决方案: android-gif-drawable Android自己的ImageView或者View不能直接加载运 行Gif图片,如果要在一个And...



**1** u010983763 (http://blog.csdn.net/u010983763) 2017-03-20 20:36

## 开源库android-gif-drawable (/tel\_15701574585/article/details/53081573)

开源库 android-gif-drawable



Tel\_15701574585 (http://blog.csdn.net/Tel\_15701574585) 2016-11-08 14:02 🕮 156

返回顶部