

管满满

热爱生活，热爱编程，分享源于学习。

[博客园](#) [首页](#) [新随笔](#) [联系](#) [订阅](#) [管理](#)

Android NDK开发之从环境搭建到Demo级十步流

写在正文之前：

几个月没有更新博客，感觉有点生疏了，所以说不能断，一断人就懒。

其实这几个月也并不是什么事也没有做，俺可是时刻想着今年的任务呢，10本书，30篇博文...，这几个月间断性的也是在学习中，学H5，学设计模式，以及NDK JNI开发等等。

学习JNI主要是因为公司有一些COCOS游戏需要添加计费点，而又没有真正的游戏开发人员，这个重任就落到我身上了，然后就是各种虐，一虐到底，苦不堪言，这种虐并不是学习技术的虐，而是一款游戏用于N种计费点，不停的改改改，那个需求这个需要的，然后你就等着被玩吧。

正文

OK，言归正传，说到NDK，相信大家都不陌生，它是Google为便于Android开发提供的一种原生开发集：Native Development Kit，而且也是一个包含API、构建工具、交叉编译、调试器、文档示例等一系列的工具集，可以帮助开发者快速开发C（或C++）的动态库，并能自动将so和java应用一起打包成APK。

公告

昵称：管满满
园龄：1年6个月
粉丝：13
关注：4
[+加关注](#)

<	2017年12月						>
日	一	二	三	四	五	六	
26	27	28	29	30	1	2	
3	4	5	6	7	8	9	
10	11	12	13	14	15	16	
17	18	19	20	21	22	23	
24	25	26	27	28	29	30	
31	1	2	3	4	5	6	

与NDK密切相关的另一个词汇则是JNI，它是NDK开发中的枢纽，Java与底层交互绝大多数都是通过它来完成的，那么接下来看看什么是JNI？

JNI：Java Native Interface 也就是java本地接口，它是一个协议，这个协议用来沟通java代码和本地代码(c/c++)。通过这个协议，Java类的某些方法可以使用原生实现，同时让它们可以像普通的Java方法一样被调用和使用，而原生方法也可以使用Java对象，调用和使用Java方法。也就是说，使用JNI这种协议可以实现：java代码调用c/c++代码，而c/c++代码也可以调用java代码。

那为什么要使用NDK开发呢？

- 我们都知道，java是半解释型语言，很容易被反汇编后拿到源代码文件，在开发一些重要协议时，我们为了安全起见，使用C语言来编写这些重要的部分，来增大系统的安全性。
- 在一些复杂性的计算中，要求高性能的场景中，C/C++更加的有效率，代码也更便于复用。

当然还有其他的优点，这些都驱使我们选择相对来说高效和安全的DNK来开发我们的应用程序。

OK，说了那么多NDK，那到底怎么使用NDK来开发应用程序呢？

俗话说，工欲善其事必先利其器，想要使用NDK开发，必先打磨好工具。那下面首先来看看DNK的环境搭建吧。

NDK的环境搭建

1. 安装配置NDK

首先下载NDK，这里我使用的是android-ndk-r14b-windows-x86_64，可以自主选择。

- 1). 解压NDK的zip包，注意路径目录不要出现**空格和中文**，这里建议大家把包解压到SDK目录里面，并命名为ndk-bundle，好处是，启动AS的时候会检查它并直接添加到ndk.dir中，减少我们的配置工作；
- 2). 配置path：把解压好的路径添加到环境变量path中；
- 3).ndk-build：cd到解压后NDK的根目录，执行ndk-build命令。

2. 给AS配置关联NDK，这里我使用的是androidstudio，使用Eclipse的会有所不同，请自行查找资料来配置。

- 1). 在建立的工程中的local.properties中添加如下配置
ndk.dir=D:\guanmanman\androidStudio\sdk\ndk-bundle,这里注意下要使用转义字符“”来进行字符转义。如果ndk目录是存放在SDK

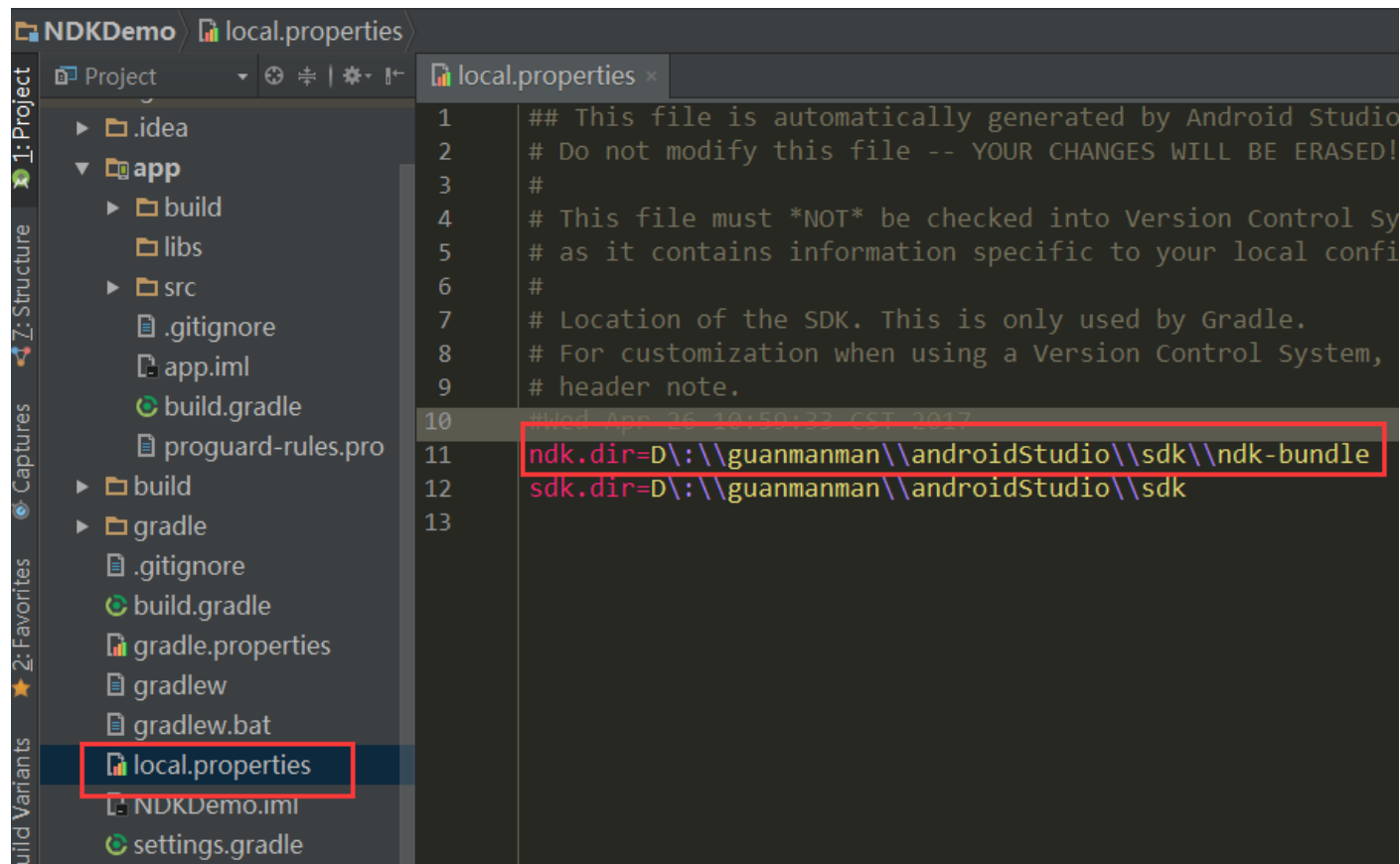
搜索

找找看

谷歌搜索

我的标签
Android(11)
5.0新特性(3)
RecyclerView(3)
源码分析(3)
自定义View(3)
view(2)
JNI(2)
NDK(2)
React Native(1)
6.0新特性(1)

中，并命名为ndk-bundle，这个配置会自动为添加上去。



2). 在工程中gradle.properties中添加对旧版本的NDK支持的配置

android.useDeprecatedNdk=true

更多

随笔档案

2017年6月 (3)

2017年5月 (3)

2017年4月 (1)

2016年12月 (4)

2016年11月 (7)

积分与排名

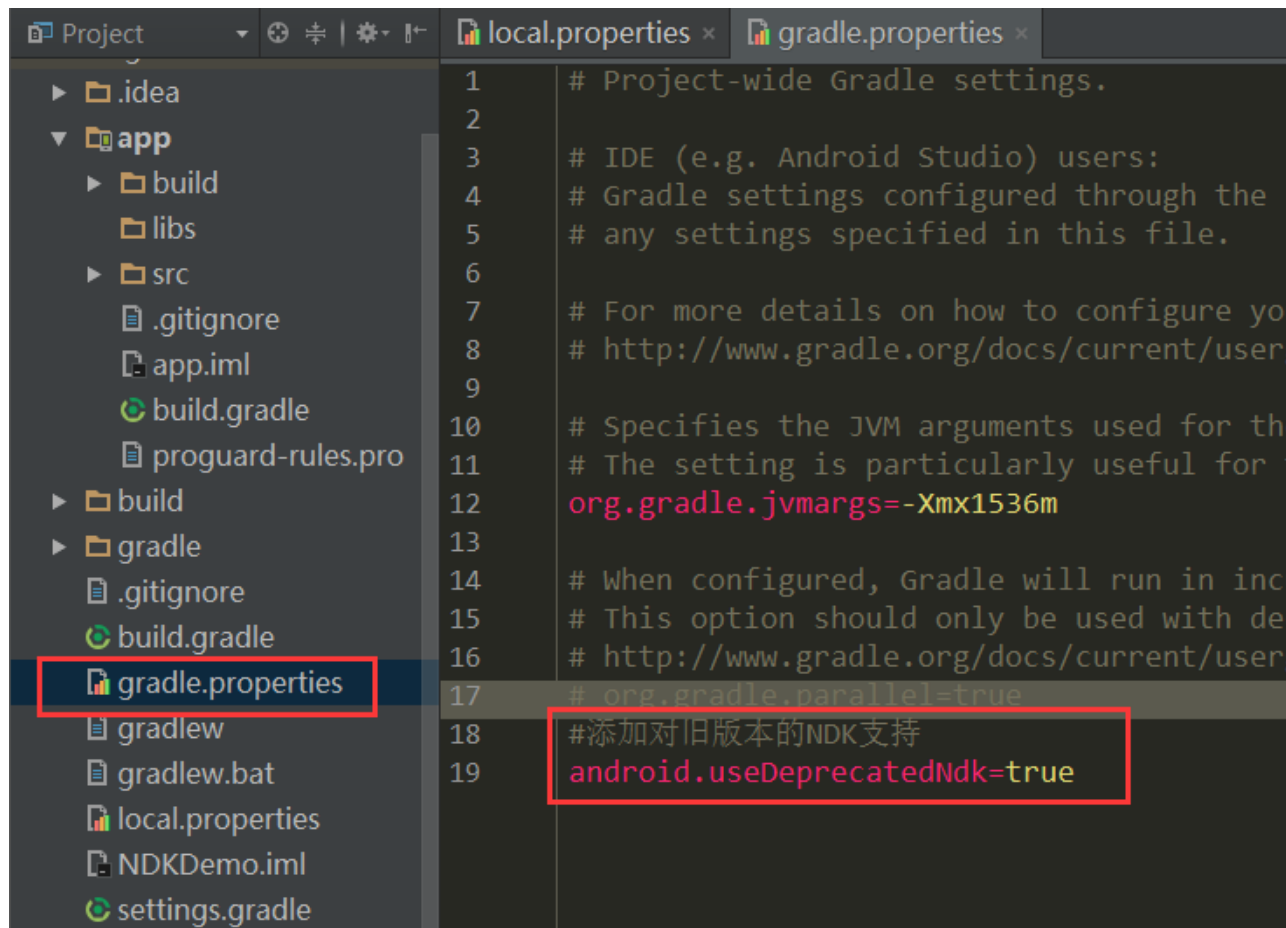
积分 - 22785

排名 - 15535

阅读排行榜

1. Android 图片加载框架Glide4.0源码完全解析（一）(4752)

2. Android6.0运行时权限管理(4088)



OK，到这里我们基本的NDK环境配置已基本完成，那接下来就开始我们的NDK开发旅程吧。

Demo实例之调用本地无参方法直接返回字符串

一 layout布局

直接在layout中添加一个按钮Button控件，用于点击调用本地方法：

3. Android 网络框架之Retrofit2使用详解及从源码中解析原理(2926)

4. Android 图片加载框架Glide4.0源码完全解析（二）(2448)

5. Android 5.X新特性之为RecyclerView添加下拉刷新和上拉加载及SwipeRefreshLayout实现原理(2195)

评论排行榜

1. 2016点滴生活：收获与展望(8)

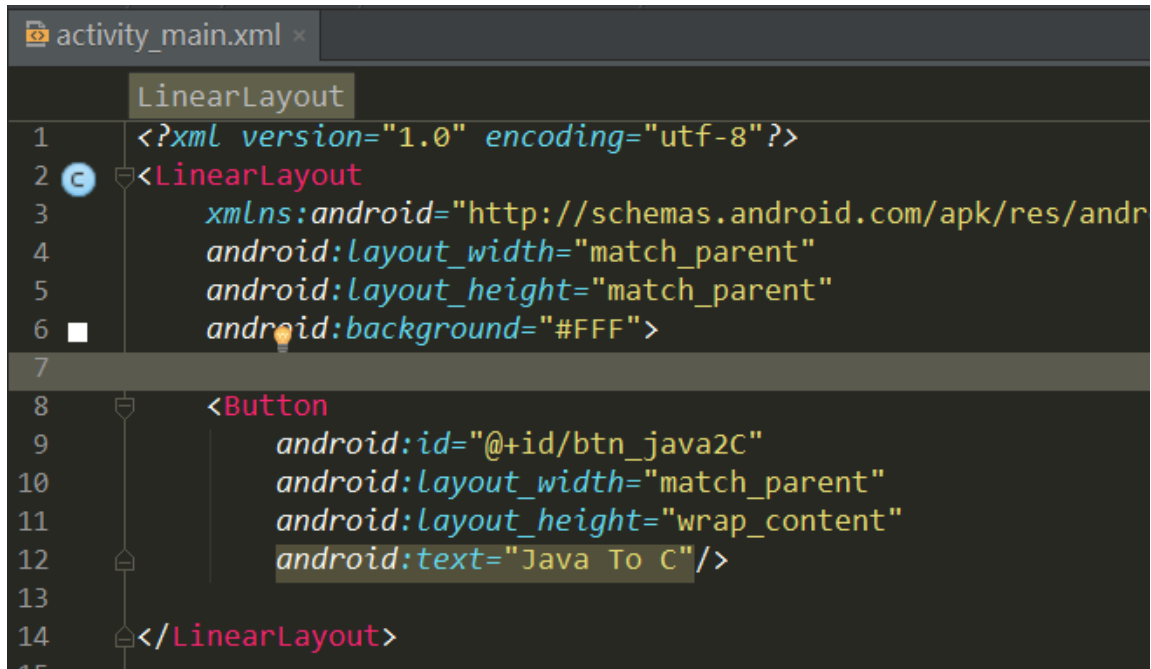
2. React Native环境配置之Windows版本搭建(1)

3. Android 自定义控件之继承ViewGroup创建新容器(1)

4. Android 设计模式实战之关于封装计费代码库的策略模式详谈(1)

5. Android NDK开发之C调用Java及原生代码断点调试（二）(1)

推荐排行榜



```
activity_main.xml x
LinearLayout
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout
3      xmlns:android="http://schemas.android.com/apk/res/android"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:background="#FFF">
7
8      <Button
9          android:id="@+id/btn_java2C"
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:text="Java To C"/>
13
14 </LinearLayout>
15
```

二 在MainActivity中获取该控件并注册它的点击监听器

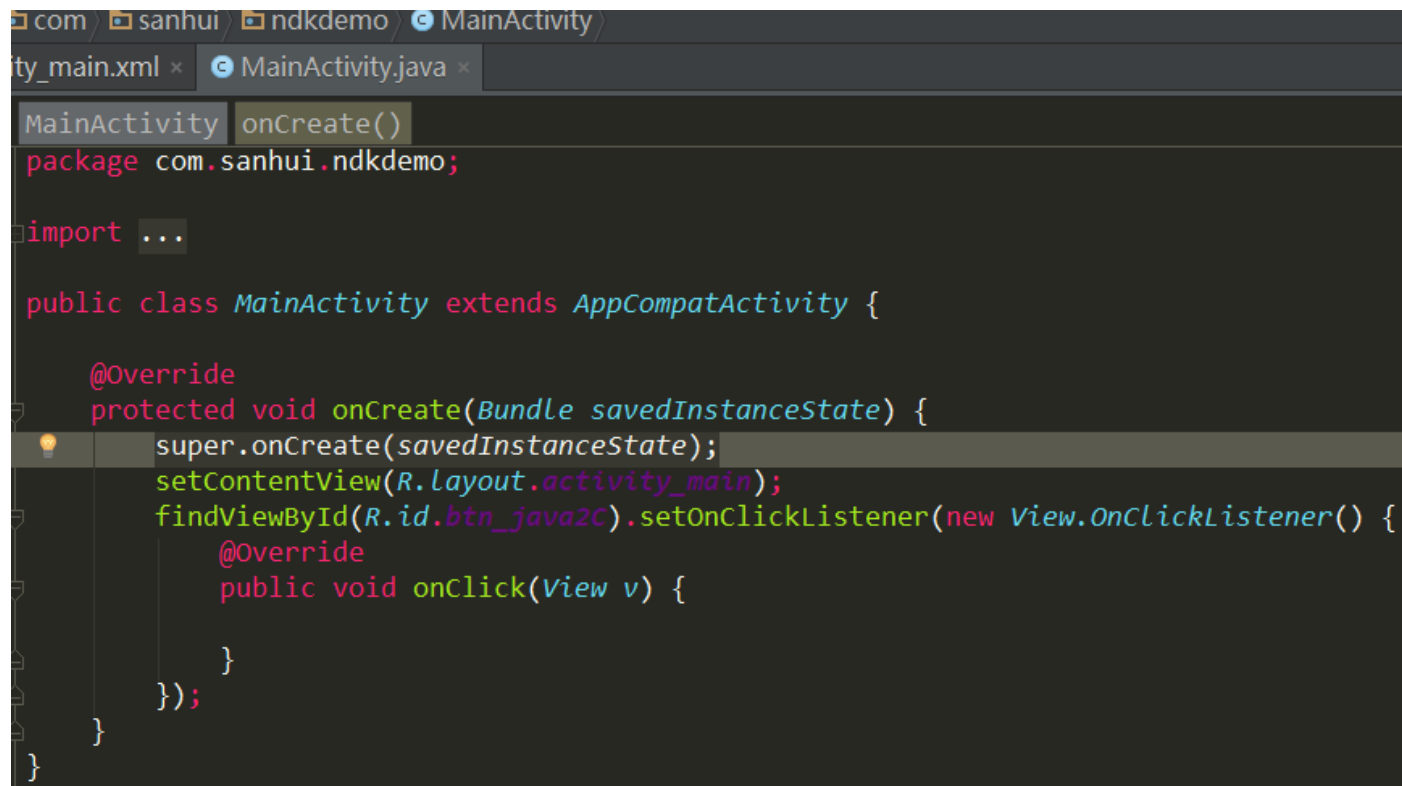
1. 2016点滴生活：收获与展望(7)

2. Android NDK开发之C调用Java及原生代码断点调试（二）(3)

3. Android NDK开发之从环境搭建到Demo级十步流(2)

4. Android 网络框架之Retrofit2使用详解及从源码中解析原理(1)

5. Android 设计模式实战之关于封装计费代码库的策略模式详谈(1)



```
com> sanhui> ndkdemo> MainActivity>
ty_main.xml x MainActivity.java x
MainActivity onCreate()
package com.sanhui.ndkdemo;

import ...

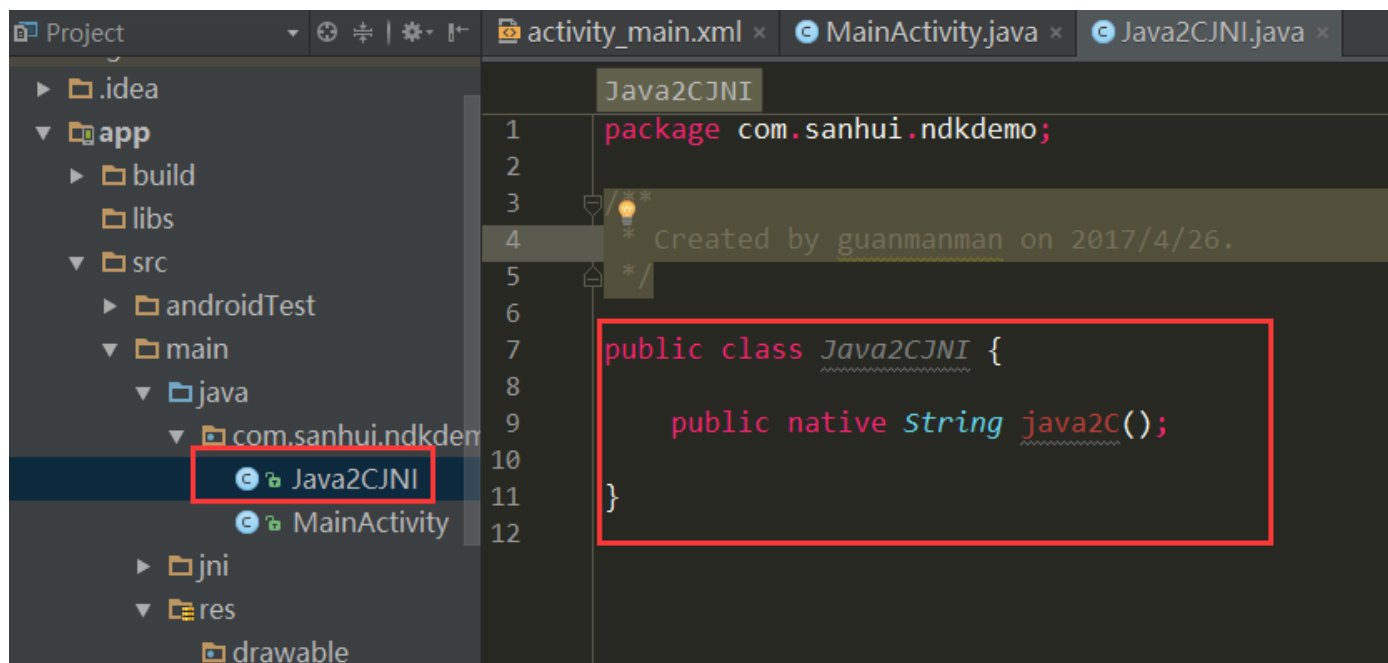
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        findViewById(R.id.btn_java2C).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {

            }
        });
    }
}
```

三 创建Java2CJNI类及本地方法

在我们的包下直接创建一个Java2CJNI类，并在类里创建一个java2C的本地方法：



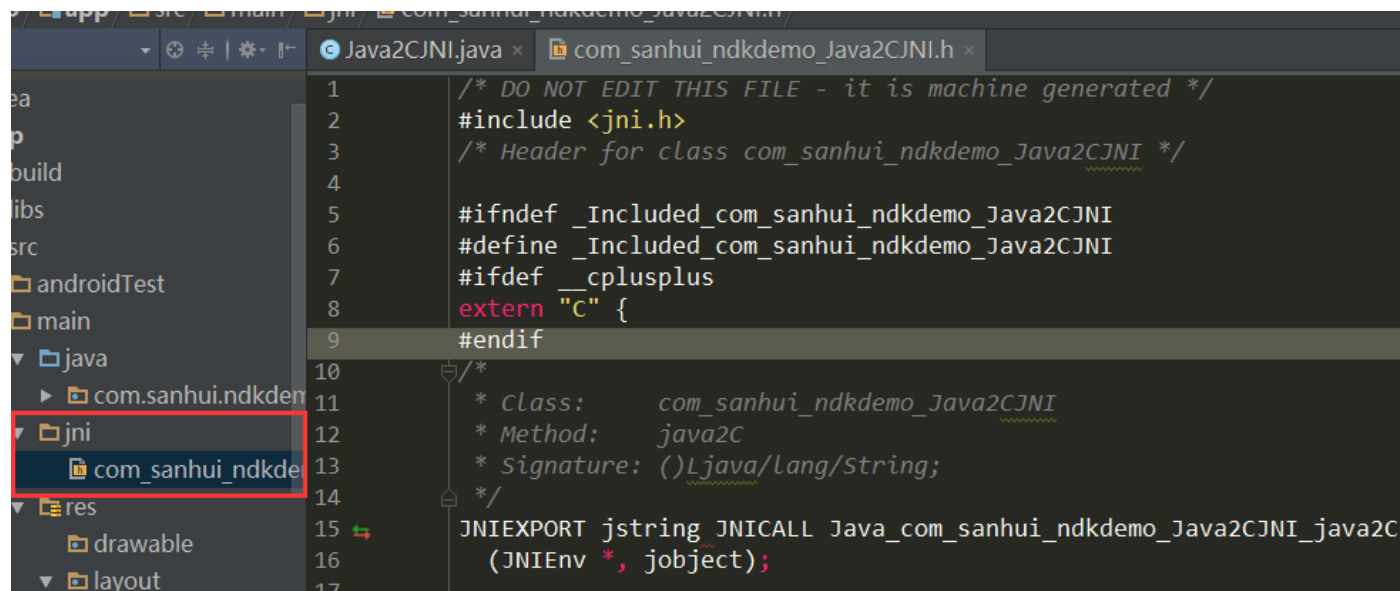
四 通过javah命令获取到本地头文件

在项目根目录下，进入main->java目录，全选文件目录栏，直接输入cmd命令并按回车键进入docs命令，在命令中执行javah com.sanhui.ndkdemo.Java2CJNI命令：



执行完javah命令后，会在java当前目录下创建一个.h的头文件

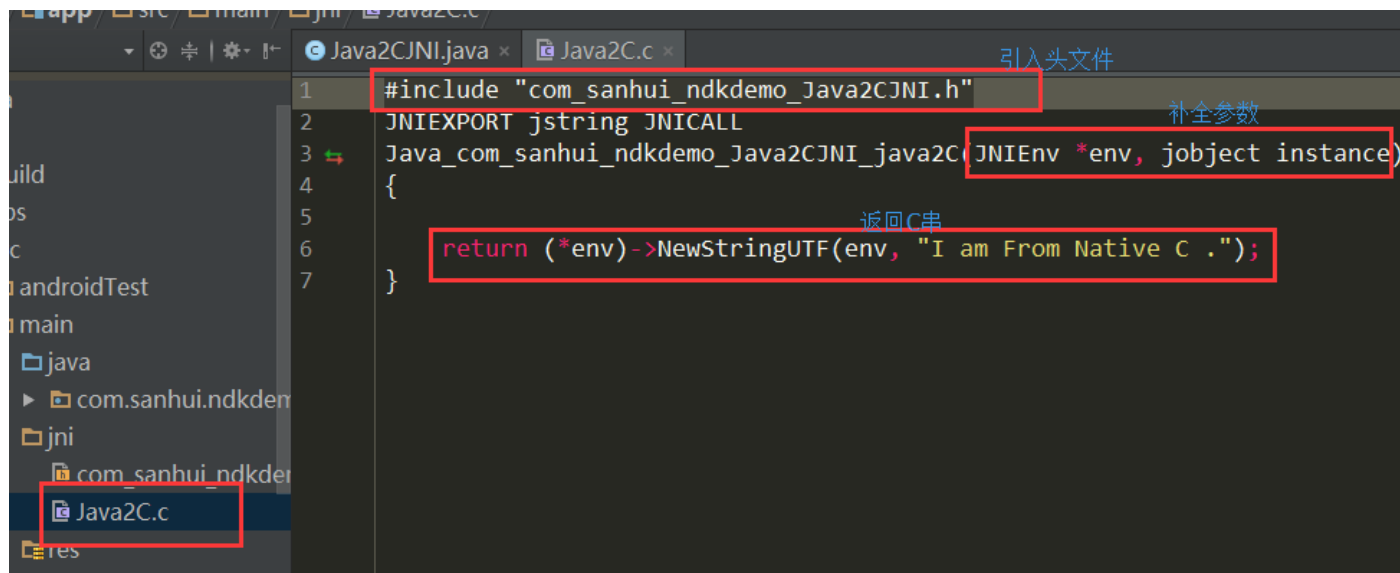
五 在main目录下创建一个jni文件夹，并把（四）中的头文件转移到该文件夹下



打开该文件夹可以看到系统为我们创建好的本地方法头文件。

六 创建实现头文件的.C源文件

在jni目录下创建一个Java2C.c的源文件，通过#include引入我们的头文件com_sanhui_ndkdemo_Java2CJNI.h，并把在头文件下的声明方法JNIEXPORT jstring JNICALL Java_com_sanhui_ndkdemo_Java2CJNI_java2C(JNIEnv *, jobject);复制到我们的Java2C.c中，补全方法参数，并实现一个C字符串“I am From Native C.”的返回：

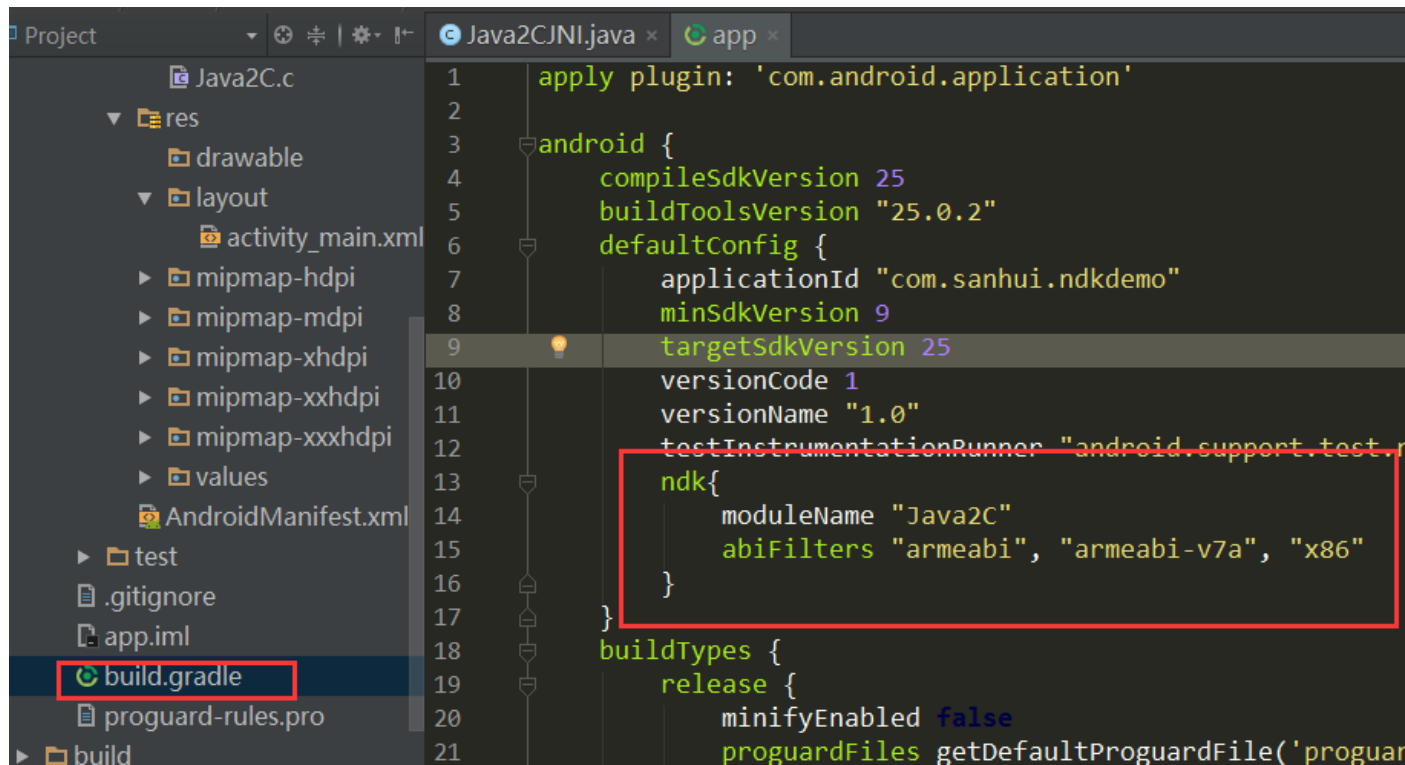


OK，至此我们的代码已编写完毕，接下来我们需要配置编译后的so名称和支持的cpu类型。

七 在该项目下的build.gradle配置生成的so名称和支持的cpu类型

在android->defaultConfig下添加如下代码：

```
ndk{
    moduleName "Java2C" //so文件名
    abiFilters "armeabi", "armeabi-v7a", "x86" //CPU类型
}
```



当然在这里不配置也是可以的，系统会用默认的项目名称作为so文件的名称，并且cpu也将会支持全部类型，只是当我们的项目名称改变的时候，在我们引用加载so文件的地方也需要改变，不改变的话就出现找不到so库的异常，所以，这里配置只是为了便利系统生成我们制定的so文件名，而不是根据项目名称生成。

八 加载so库

在我们创建的Java2CJNI类中加载so库，主要是为了在我们调用本地方法之前先编译本地源码。



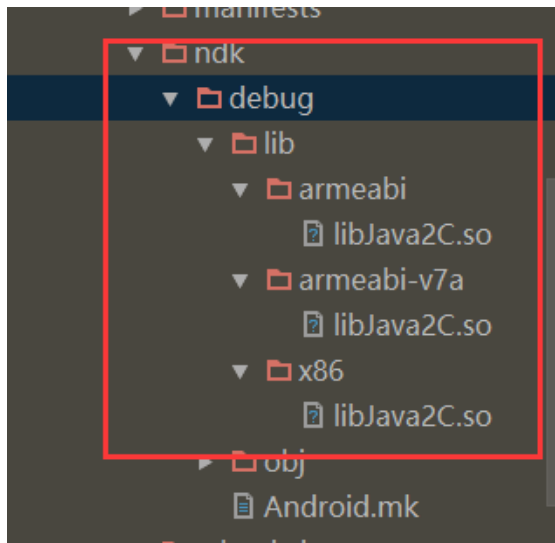
```
Java2CJNI.java x app x
Java2CJNI
1 package com.sanhui.ndkdemo;
2
3 /**
4  * Created by guanmanman on 2017/4/26.
5  */
6
7 public class Java2CJNI {
8
9     static {
10         System.loadLibrary("Java2C");
11     }
12
13     public native String java2C();
14
15 }
16
```

在使用 `System.loadLibrary("Java2C");`加载库时，库名一定要与在build.gradle中配置的moduleName 名称一致，否则将找不到库。

九 生成so文件

在项目的工具类中选择Build->Rebuild Project，进行重新编译工程，然后AS会为我们生成so文件，so文件所在目录为：

NDKDemo\app\build\intermediates\ndk\debug\lib下



注意：so文件命名方式是：lib+moduleName+.so

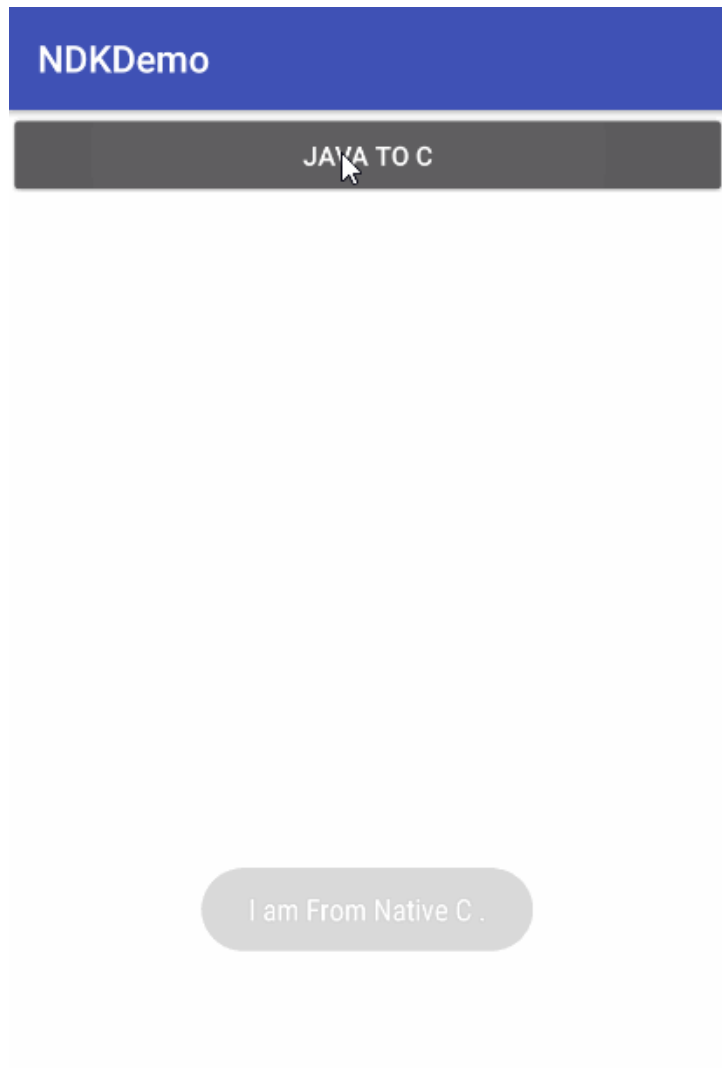
十 执行调用本地方法

在MainActivity中点击Button按钮调用本地方法。并通过Toast打印出来。

```
public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        findViewById(R.id.btn_java2C).setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { // 调用本地方法
                String result = new Java2CJNI().java2C();
                Toast.makeText(MainActivity.this, result, Toast.LENGTH_LONG).show();
            }
        });
    }
}
```

OK，到这里已经完成了一个DEMO级别的NDK应用开发了，那么来看看我们的执行结果：



到这里一个DEMO级别的NDK开发已经完成了，通过上面的十步已经淋漓尽致的展现了NDK开发的所有步骤流程，相信你已经完全的学会啦。

下面章节将会讲述Java调用本地C方法和C回调Java方法的实例，也会穿插的讲解下JNI的更多使用，敬请期待。

更多资讯请关注微信平台，有博客更新会及时通知。爱学习爱技术。



标签: Android, NDK, JNI

好文要顶

关注我

收藏该文



管满满

关注 - 4

粉丝 - 13

+加关注

2

0

« 上一篇: 2016点滴生活: 收获与展望

» 下一篇: Android NDK开发之从Java与C互调中详解JNI使用 (一)

posted @ 2017-04-26 16:40 管满满 阅读(1322) 评论(0) 编辑 收藏

刷新评论 刷新页面 返回顶部

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问](#)网站首页。

【推荐】50万行VC++源码: 大型组态工控、电力仿真CAD与GIS源码库

【推荐】腾讯云免费实验室，1小时搭建人工智能应用

【新闻】H3 BPM体验平台全面上线



最新IT新闻:

- 铁路12306 App迎3.0版更新：全新界面
 - 若福克斯资产出售 可能会让Hulu成Netflix强大对手
 - 三星260亿美元的豪赌：想垄断DRAM和NAND闪存市场
 - 苹果还是没有放弃？屏下指纹识别专利被曝光
 - Google员工和Google.org在2017年向非营利组织捐赠了2.6亿美元
- » 更多新闻...



告别高昂运维费用 云计算全面助力

40+款核心产品免费半年 再+8000津贴任意采购

立即申请

最新知识库文章:

- 以操作系统的角度述说线程与进程
- 软件测试转型之路
- 门内门外看招聘

- 大道至简，职场上做人做事做管理
- 关于编程，你的练习是不是有效的？
- » 更多知识库文章...

Copyright ©2017 管满满