## Windows下Android打包教程:将Android 源码打包成.APK可 执行文件

## 原创

<u>飞翔的大黑壮(猫猫)</u> 
● 于 2020-01-20 16:41:52 发布 ● 5441 ☆ 收藏 11 分类专栏: <u>高级测试 测试基础部分 手机专项测试</u> 只做原创,告别转载。 本文链接: <u>https://blog.csdn.net/pingsha\_luoyan/article/details/104051411</u> 版权

订阅专栏



25 篇文章 3 订阅 订阅专栏



<u>手机专项测试</u> 5 篇文章 0 订阅 订阅专栏

最基出的环境:

Node.js, Android SDK, Java JDK(1.8)

nodejs安装包中文下载地址: http://nodejs.cn/download/

测试基础部分

Android SDK下载地址: https://www.androiddevtools.cn/

Java JDK下载地址:

## 打包环境搭建:

1.首先安装Java JDK, 配置Java JDK环境变量。

为什么安装java: android sdk是依赖java的,只有安装了java环境,才能保证android sdk的正常运行

2.安装Android SDK, 配置Android SDK的环境变量。

📕   🖓 📕 🖛		督	町	Android sdk								- 0	×
文件 主页 共享	查看	应用程	·~ 評工具	-									~ 🕐
← → v ↑ ] > 此	电脑 > 机	戒盘2 (E	:) > pro	gram files > And	droid sdk >				~ č	) 搜索"An	droid sdk"		٩
APP ^	名称		^	-	修改日期	类型	大小						
- 保存的图片					2020/1/20 10:52	÷-/⊬-==							
■ 图片		ons			2020/1/20 10:52	又件关							
「二」「二」	bulud	-tools			2020/1/19 17:31	又件关							
1,380	a docs	_			2020/1/19 18:59	又件夹							
OneDrive	extra	s			2020/1/19 17:36	文件夹							
∧ w/ps网盘		orms	-1-		2020/1/19 19:04	又件关	下井	<u>)</u>	· 立 地 た の	ルンド	北哥	landro	bid
C) WI SMIM		orm-too	OIS		2020/1/19 14:27	又件夹	1140	67671	الالكري		,北土	Janur	Jiu
🧢 此电脑	system-images				2020/1/19 19:10	文件夹	sdk	的安	装路径	然后	找到圈	出的5	ל ז
🧊 3D 对象	temp				2020/1/20 10:40	文件关	<i>1</i> #	JU +					ļ
📑 视频					2020/1/20 10:52	文件夹	14,	双击	l o				
▶ 图片	knownPackages				2019/6/18 22:57			1.170					,
圖 文档		ипраска	ages		2020/1/20 11:50	KNOWNPACKA	0.077.5	T KB					
	android-sdk.zip				2019/6/19 10:28	WINKAR ZIP 述缩	2,211,5	o/6					
		Managi	er exe		2015/10/14 9:51	应用程序	210	6 KB					
┛ 首乐	SDK	Manage	er.exe		2015/10/14 9:51	应用程序	210	6 KB					
■ 桌面	SDK	Readme	e.txt		2015/10/14 9:51	又本又档	-	2 KB					
💺 本地磁盘 (C:)	tin 🖏 🖏	tall.exe	9		2019/6/18 22:26	应用程序	/(	0 KB					
🥪 机械盘1 (D:)													
✓ 机械盘2 (E:)													
Android SDK Manager Packages Tools												-	• ×
SDK Path: E:\program files\Android_	sdk												
Packages													
🖗 Name	۵	PI Rev.	Status										
Toois (Preview Channel)													
API 28, Q preview													
Android 9 (API 28)													
Android 8.0.0 (API 26)				选择适合	自己的android	1版本,根据自	1己的需	一般下载	戈,				
Android 7.1.1 (API 25)				一还要讲To	ools里面适配的	文件下载下载	。有点	、, 1	「载				
> Android 6.0 (API 23)				时间长。	要有心理准备。								
Android 5.1.1 (API 22)     Android 5.0.1 (API 21)					хно								
> Android 4.4W.2 (API 20)													
Android 4.4.2 (API 19)													
> Android 4.2.2 (API 17)													
Android 4.1.2 (API 16)													
Android 4.0 (API 14)													
Android 3.2 (API 13)     Android 3.1 (API 12)													
Android 3.0 (API 11)													
Android 2.3.3 (API 10)													
Android 2.2 (API 8)													
> Extras													<sup>KUS</sup> 51%)
Show: Updates/New Installed	Select New (	or Update	s								E	• • 🙂 🖢	, 📟 🐁 👕 🔡
	Deselect All	<u>- pource</u>										Delet	e packages
												2.100	
Done loading packages.													bingsha_luoyar

3.安装Node.js,Node.js在安装时,自动配置好环境变量,但是存在没有配置的情况,需要自己确认一下。

例如:这是我自己电脑上的配置环境,环境变量path下必须有java,node.js,Android的环境变量,为了防止环境变量路径输入错误,需要进入到cmd中查看,不要嫌弃繁琐,认为确认是没有必要的,但是请明白,环境如果搭建不好,一切都是镜花水月。(现身说法:刚开始学习java的时候,那时候使用的windo 7系统,配置环境变量时,拉掉了一个;导致最基出的HelloWorld都实现不了。)





基础环境已经搭建好。你认为这样就结束了吗,然而并没有,



下面是硬菜了:

说明: 打包分为两种, 第一张秘钥打包, 另外一种非秘钥打包, 本次介绍的是秘钥打包, 其实秘钥打包和非秘 钥打包基本上相似, 知识秘钥打包多了几个小步骤而已。

1.没有秘钥生成秘钥,若是已经有秘钥,则将秘钥放在android\app目录下。

生成秘钥的命令:

\$ keytool -genkeypair -v -keystore my-release-key.keystore -alias my-key-alias -keyalg RSA -keysize 2048 -v

解释秘钥生成命令:

1.在 Windows, keytool命令在Java JDK 的 bin 目录中。

2.alias后面跟着的是秘钥的别名, my-key-alias, 可以是任意字符, 只要不提示错误即可。因一个证书库 中可以存放多个证书, 通过别名标识证书

3.keyalg后跟的是加密的算法: RSA,秘钥算法有三种: RSA、DSA、EC

4.keysize 后跟的是密钥长度,2048。(keyalg要注意与keysize 相对应)

RSA、DSA、EC 分别对应 2048、1024、256

5.validity:证书保质期,在多少天内有效

2.设置gradle环境:

将生成的秘钥放置到android\app目录下,之后修改android/gradle.properties文件。

```
MYAPP_RELEASE_STORE_FILE=my-release-key.keystore
MYAPP_RELEASE_KEY_ALIAS=my-key-alias
MYAPP_RELEASE_STORE_PASSWORD=*****
MYAPP_RELEASE_KEY_PASSWORD=*****
```

3.将签名配置加到gradle配置中。

android/app/build.gradle中添加。

```
android {
    . . .
    defaultConfig { ... }
    signingConfigs {
        release {
            if (project.hasProperty('MYAPP_RELEASE_STORE_FILE')) {
                storeFile file(MYAPP_RELEASE_STORE_FILE)
                storePassword MYAPP_RELEASE_STORE_PASSWORD
                keyAlias MYAPP_RELEASE_KEY_ALIAS
                keyPassword MYAPP_RELEASE_KEY_PASSWORD
            }
        }
    }
    buildTypes {
        release {
            . . .
            signingConfig signingConfigs.release
        }
    }
}
```

## 4.生成apk包。

进入android目录下,执行:

gradlew assembleRelease

成功后就可以在目录android/app/build/outputs/apk/release/下找到app-release.apk包了。