React-Native环境配置

window环境下搭建react-native

需要安装的工具包：

* Node.js
* React native 命令行工具
* Python2
* JDK
* Android Studio

注意 Node 的版本必须高于 8.3，Python 的版本必须为 2.x（不支持 3.x），而 JDK 的版本必须是 1.8（目前不支持 1.9 及更高版本）。安装完 Node 后建议设置 npm 镜像以加速后面的过程

# 安装Node

React Native开发需要用到Node.js环境，我们做React Native开发会经常性的和node.js进行打交道，例如：我们用npm start命令启动react native的启动 器，用npm install安装项目所依赖的第三方组件，用npm publish往npm上发布一些组件等，版本需要10以上。

## 下载地址：

<https://nodejs.org/zh-cn/>

## 安装过程

如不需要改变软件安装路径，一直点击下一步即可，此处省略具体安装过程。

## 验证安装是否成功

安装后，打开命令提示符窗口，输入npm -v命令查看node的版本



## 设置 npm镜像加速

npm config set registry https://registry.npm.taobao.org --global

npm config set disturl https://npm.taobao.org/dist --global

上面两条语句主要是切换成淘宝团队的镜像(感谢淘宝团队)

大大增加了在国内npm安装模块的速度

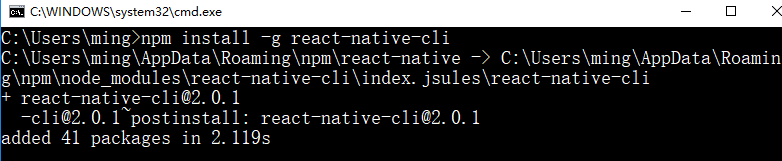
# 二、安装React Native命令行工具

Node.js安装成功之后，接下来我们就可以通过npm install 来安装React Native命令行工具

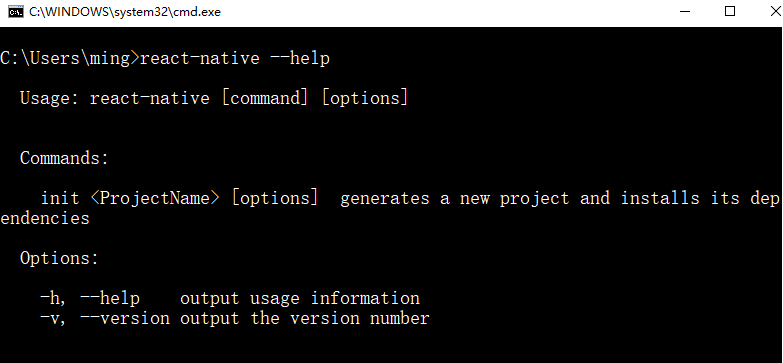
打开命令提示符窗口，输入以下命令：

npm install -g react-native-cli

安装成功提示：



React Native命令行工具安装成功之后，我们可以通过react-native --help来查看它所支持的所有命令。



如果报错的话，表示React Native命令行安装失败，那么则换成Yarn来安装React Native命令行。

npm install -g yarn react-native-cli

安装完 yarn 后同理也要设置镜像加速：

yarn config set registry https://registry.npm.taobao.org --global

yarn config set disturl https://npm.taobao.org/dist --global

安装完 yarn 之后就可以用 yarn 代替 npm 了，例如用yarn代替npm install命令，用yarn add 某第三方库名代替npm install 某第三方库名。

# 三、安装Python2

Python 的版本必须为 2.x（不支持 3.x）

# 四、安装JDK

JDK 的版本必须是 1.8（目前不支持 1.9 及更高版本）。

## 下载地址

<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

## 安装

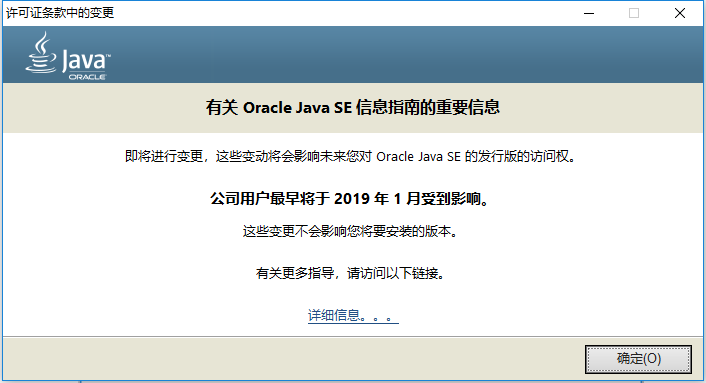
(1)点击下一步：



(2)选择安装路径，如无需变动，则默认即可，然后再点击下一步



1. 点击完下一步后，JDK就会进行安装，会出现下图提示：



点击确定，后会再弹出JRE的安装提示，注意要把JRE和JDK安装到同一目录下，即JDK和JRE是平级目录



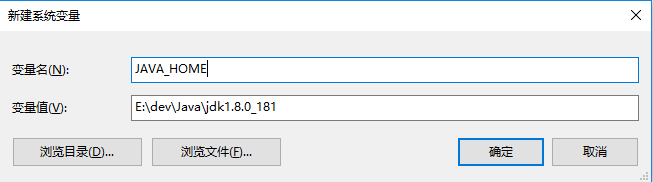
再点击下一步，即可完成安装。



## 配置JAVA环境变量

(1)鼠标右键点击计算机→属性→高级系统设置→高级→环境变量

(2)系统变量→新建 JAVA\_HOME 变量，变量值填写jdk的安装目录



(3)系统变量→寻找 Path 变量→编辑,在变量值输入

如果操作系统是win7:

在原来的环境变量基础上添加以下内容

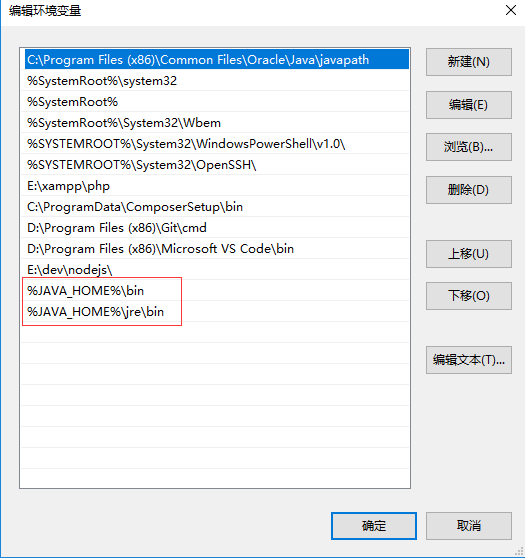
;%JAVA\_HOME%\bin;%JAVA\_HOME%\jre\bin

如果操作系统是win10

%JAVA\_HOME%\bin

%JAVA\_HOME%\jre\bin

（注意原来Path的变量值末尾有没有;号，如果没有，先输入；号再输入上面的代码）



1. 系统变量→新建 CLASSPATH 变量。

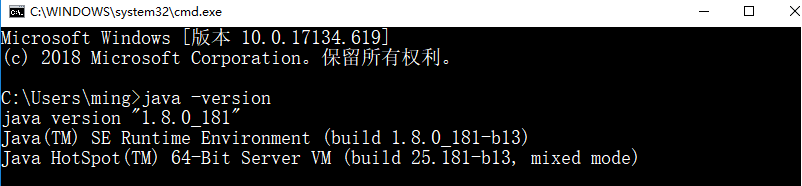
变量值填写以下内容：

.;%JAVA\_HOME%\lib;%JAVA\_HOME%\lib\tools.jar（注意最前面有一个点）

## 验证安装是否成功

检测java中的jdk环境是否安装正确，在命令提示符窗口中输入一下命令：

java -version



# 安装并配置Android Studio

## 下载地址

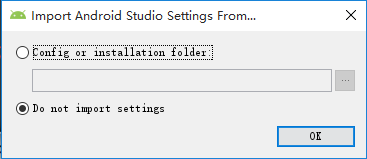
<http://www.android-studio.org/index.php>

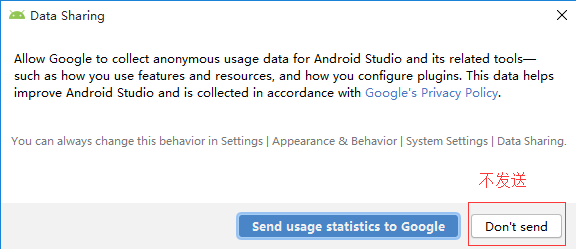
## 安装Android Studio

如无需改变安装路径，则一直下一步即可

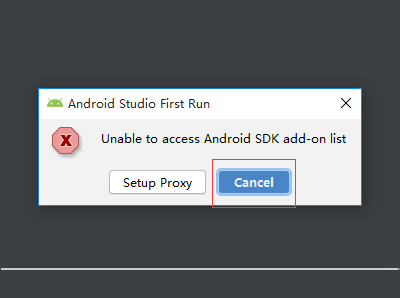
安装完成后

导入设置：

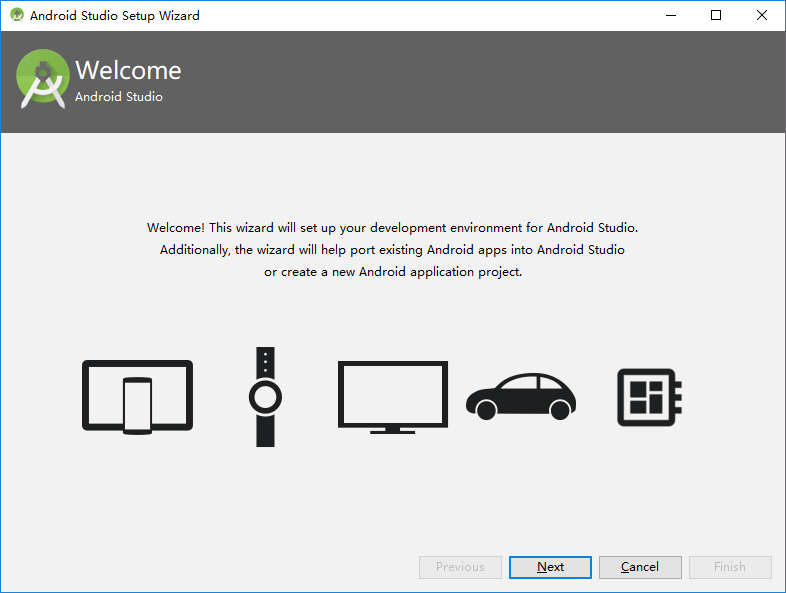




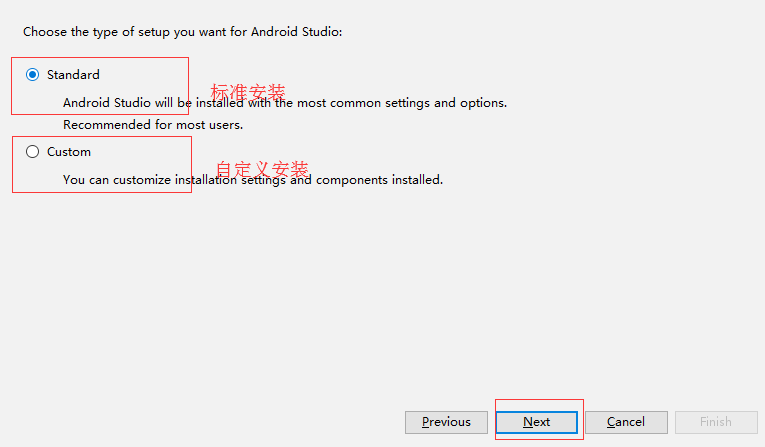
第一次运行：



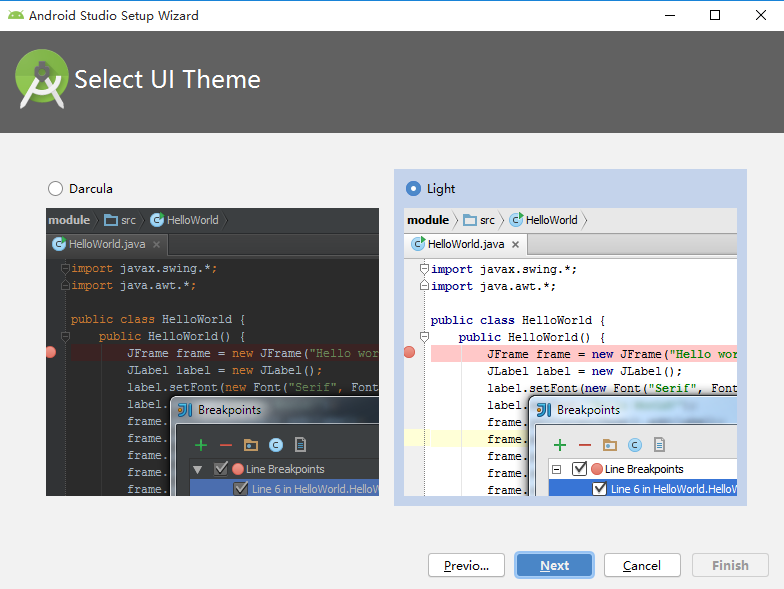
安装模拟器：



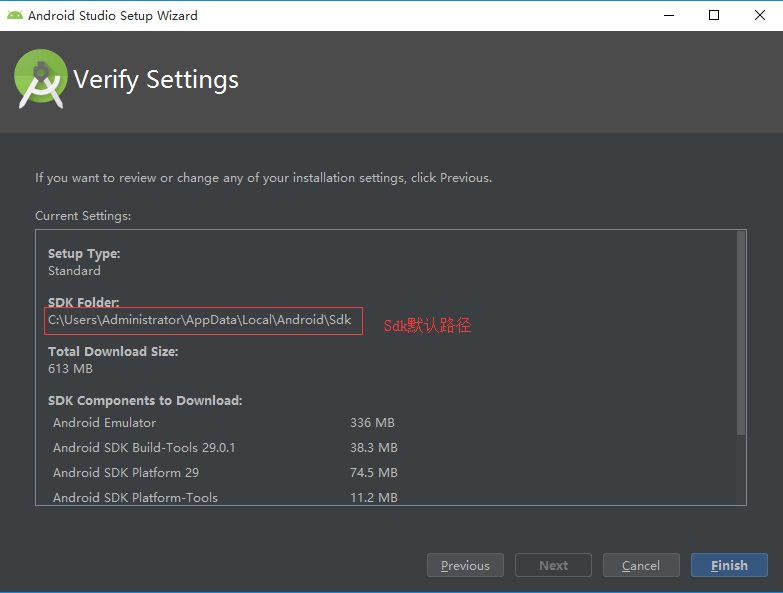
选择安装方式：



选择UI主题(看心情)：

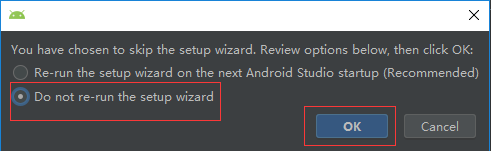


安装完成：

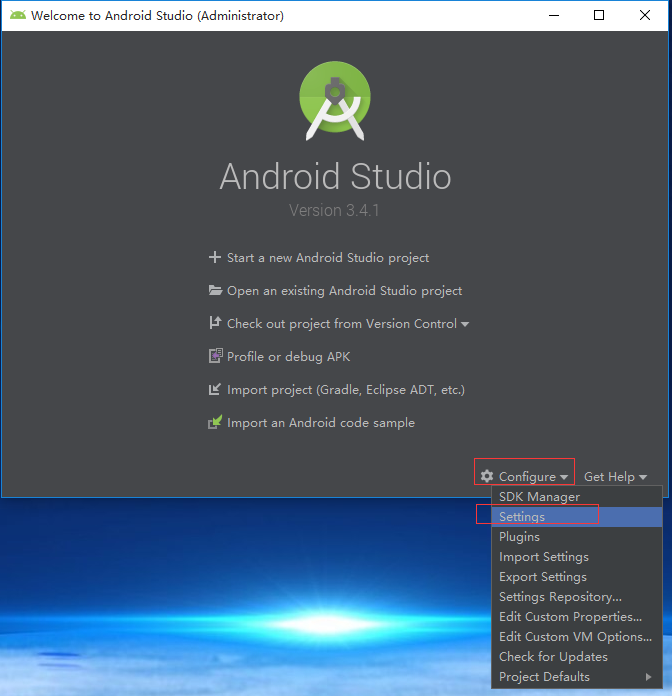


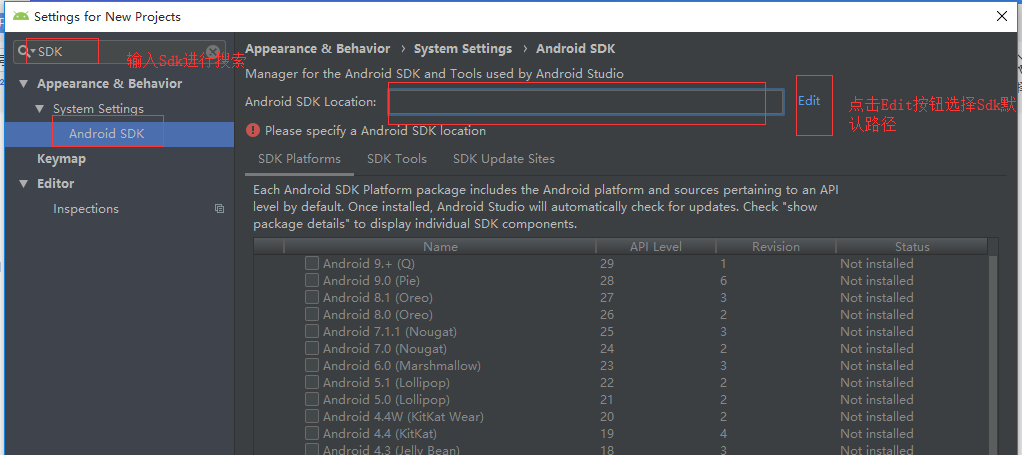
ReactNative需要通过环境变量来了解你的Android SDK装在什么路径，从而正常进行编译。

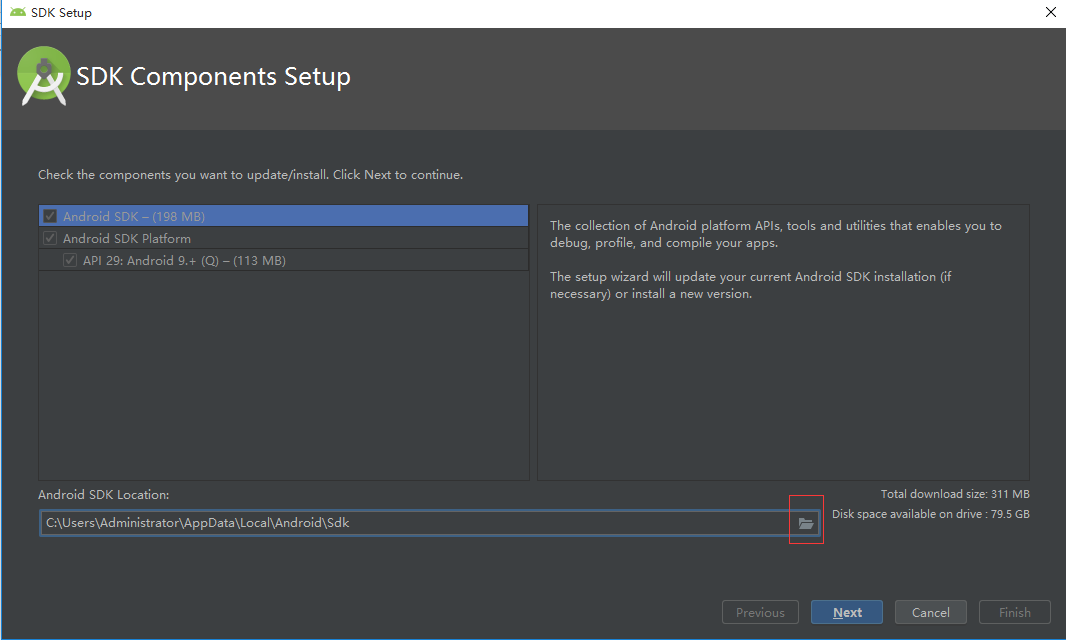
如果不想使用默认路径，则点击cancel按钮



点击OK按钮终止程序设置，然后会出现如下界面，在Configure里找到Settings选项来设置Sdk默认路径







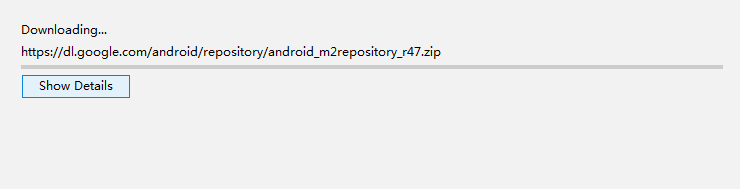
选择好Sdk默认路径后，在环境系统变量中再新建一个ANDROID\_HOME变量

打开控制面板 -> 系统和安全 -> 系统 -> 高级系统设置 -> 高级 -> 环境变量 -> 新建，创建一个名为ANDROID\_HOME的环境变量（系统或用户变量均可），指向你的 Android SDK 所在的目录（具体的路径可能和下图不一致，请自行确认）

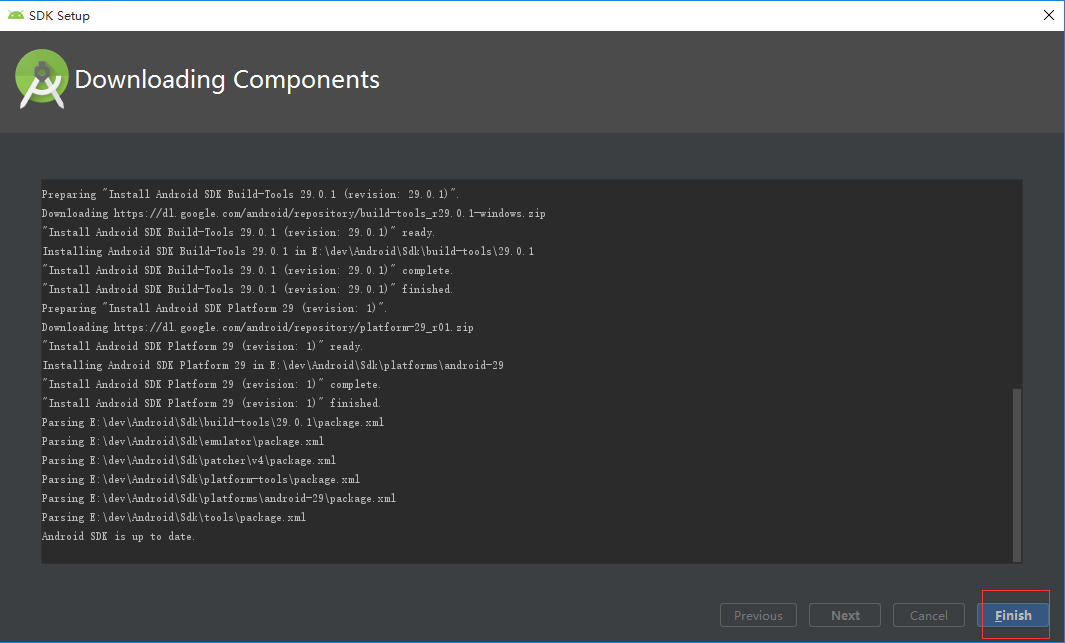


设置好Sdk路径后，点击next，然后会下载相关Sdk包

下载中，需要等待



等待下载直到完成(期间有失败的地方，重复)



点击finish

|

|

|

|

|

|

|

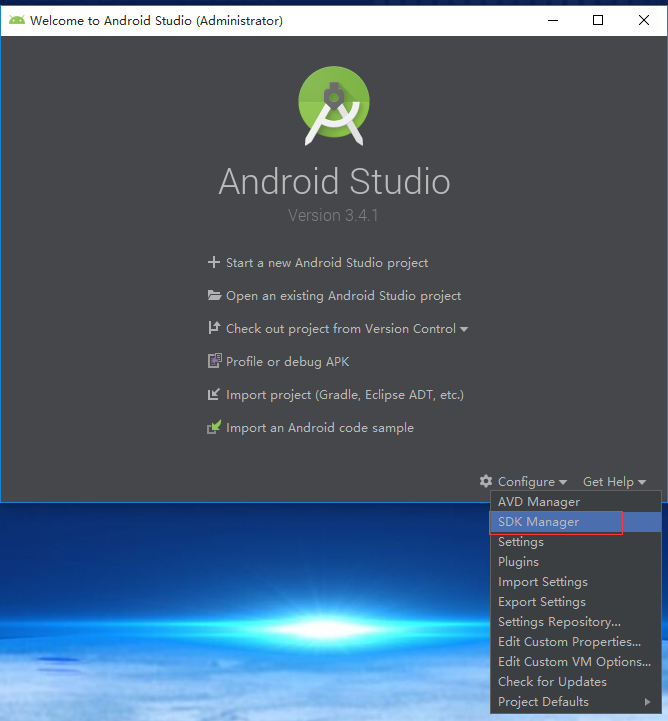
|

|

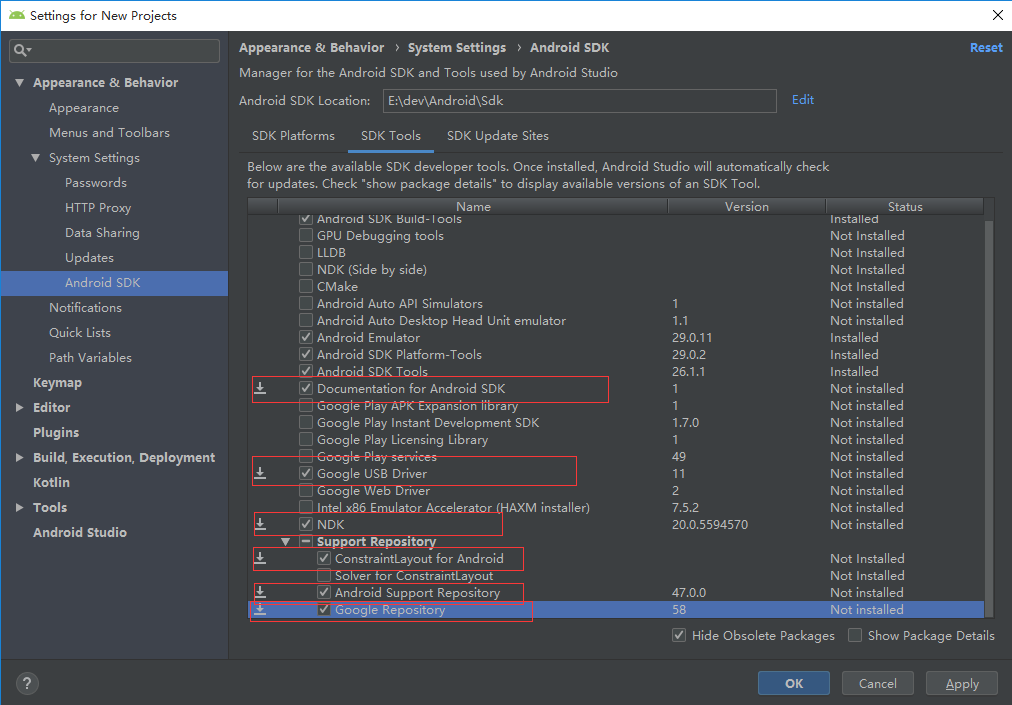
|

|

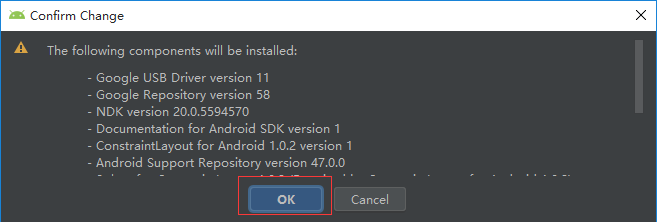
然后点击configure，选择 Sdk Manager



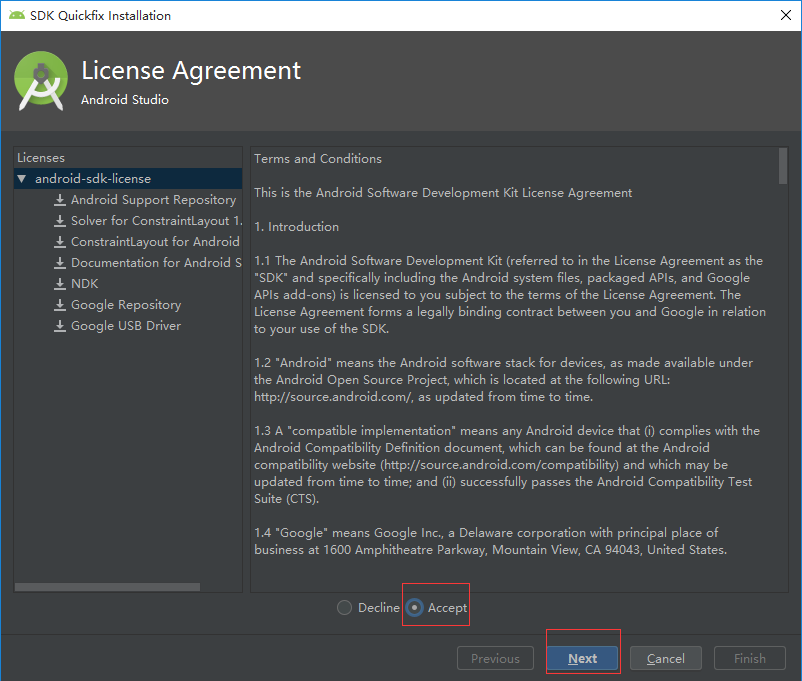
sdk tools



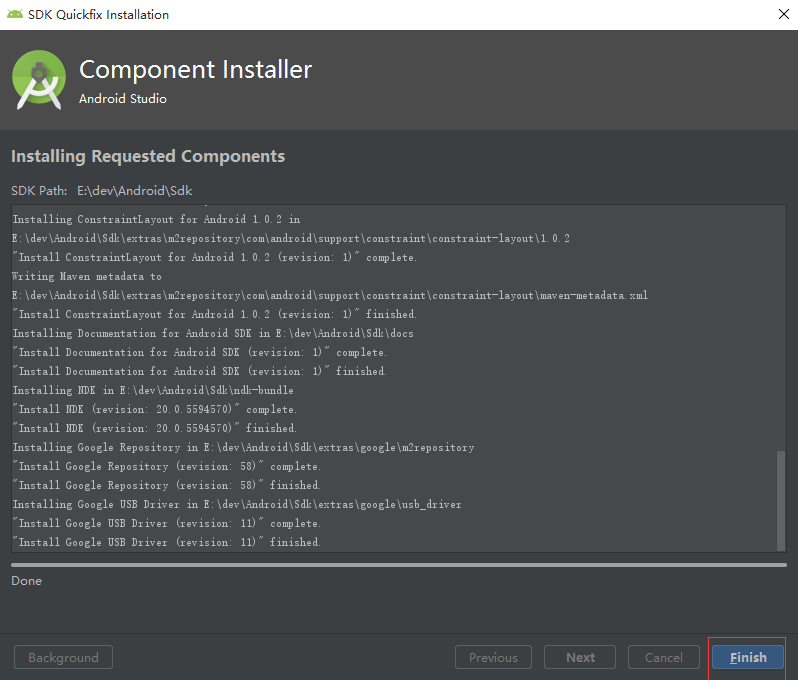
确认要下载的包



接受协议，点击next



等待下载



下载完成后下载Finish按钮完成Sdk tools的设置

## 创建项目

初始化一个React Native应用，我们可以通过React Native命令行工具来完成，打开命令提示符窗口并运行下面命令：

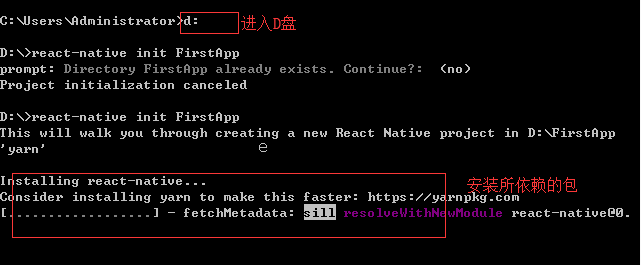
react-native init 项目名称

进入到项目的盘符：比如d盘

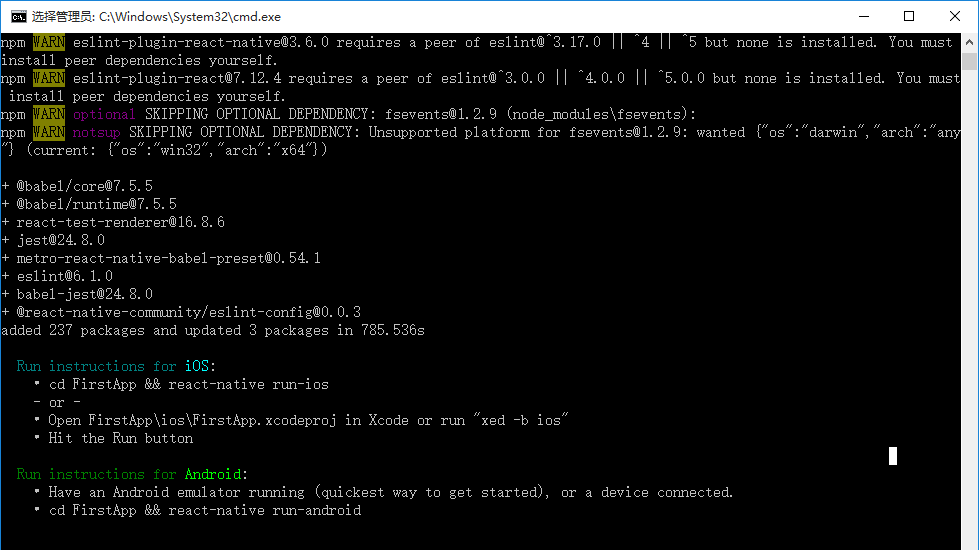
执行命令：

react-native init FirstApp --version 0.59.4

必须要指定版本，最新版不好用



运行此命令后，需要等待一段时间，React Native从npm上下载一些项目所依赖的包， 并完成项目的初始化，初始化完成之后你会看到下图的输出



然后我们打开创建的项目根目录会看到如下的项目结构：



## 项目目录结构

android：Android的原生开发目录，可以用Android Studio打开进行原生开发。

ios: Ios的原生开发目录，可以用Xcode打开进行原生开发。

node\_modules: 存放所有的项目依赖库，配置package.json之后执行“npm install”后自动创建的文件夹。

.buckconfig: buck的配置文件，buck是Facebook推出的一款高效率的App项目构建工具。

.flowconfig: Flow 是 Facebook 旗下一个为 JavaScript 进行静态类型检测的检测工具。它可以在 JavaScript 的项目中用来捕获常见的 bugs，比如隐式类型转换，空引用等等。

.gitattributes: git属性文件设定一些项目特殊的属性。比如要比较word文档的不同；对strings程序进行注册；合并冲突的时候不想合并某些文件等等。

.gitignore: 用来配置git提交需要忽略的文件。

.watchmanconfig: 用于监控bug文件和文件变化，并且可以出发指定的操作。

app.json: 配置了name和displayName。

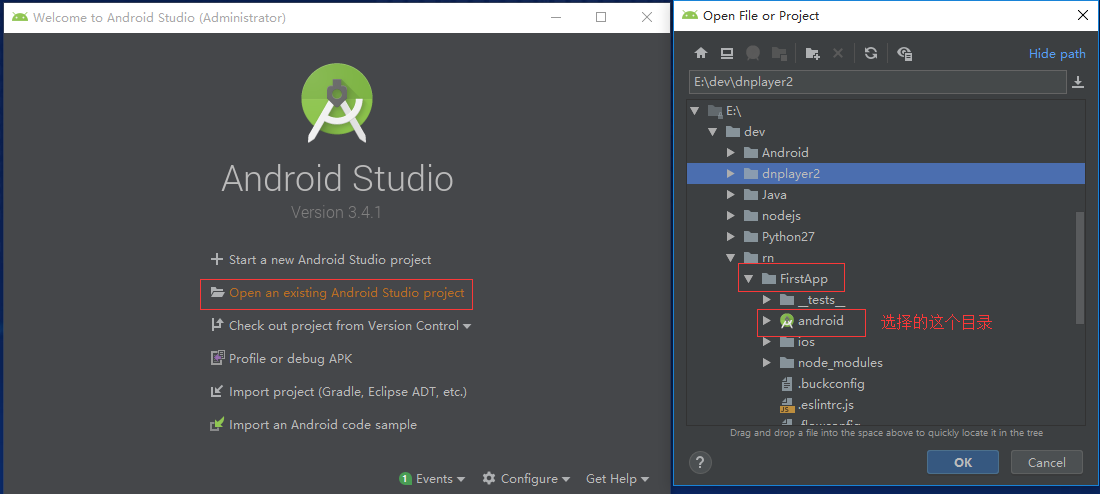
index.js : Android 和 Ios 入口文件，可以在android的MainApplication中的ReactNativeHost中重写getJSMainModuleName()方法更改; 在Ios的AppDelegate.m文件的

didFinishLaunchingWithOptions方法中通过jsBundleURLForBundleRoot可以更改入口文件。

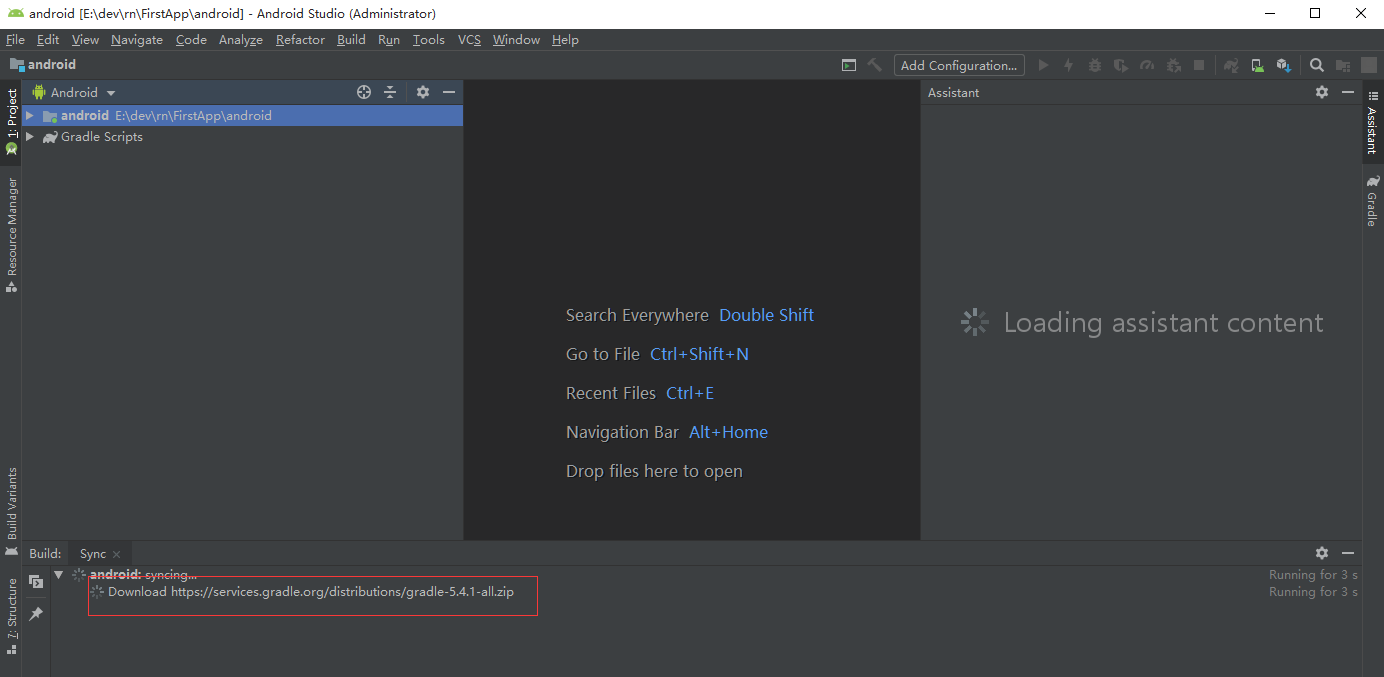
package.json: package.json定义了项目所需要的各种模块，以及项目的配置信息（比如名称、版本、许可证等元数据）。npm install命令根据这个配置文件，自动下载所需的模块，也就是配置项目所需的运行和开发环境。

## 配置模拟器环境

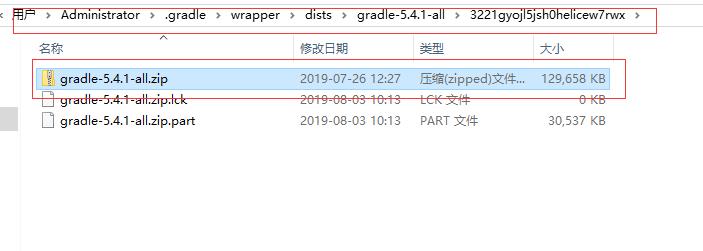
使用android studio打开项目，注意，打开的是项目中的Android目录



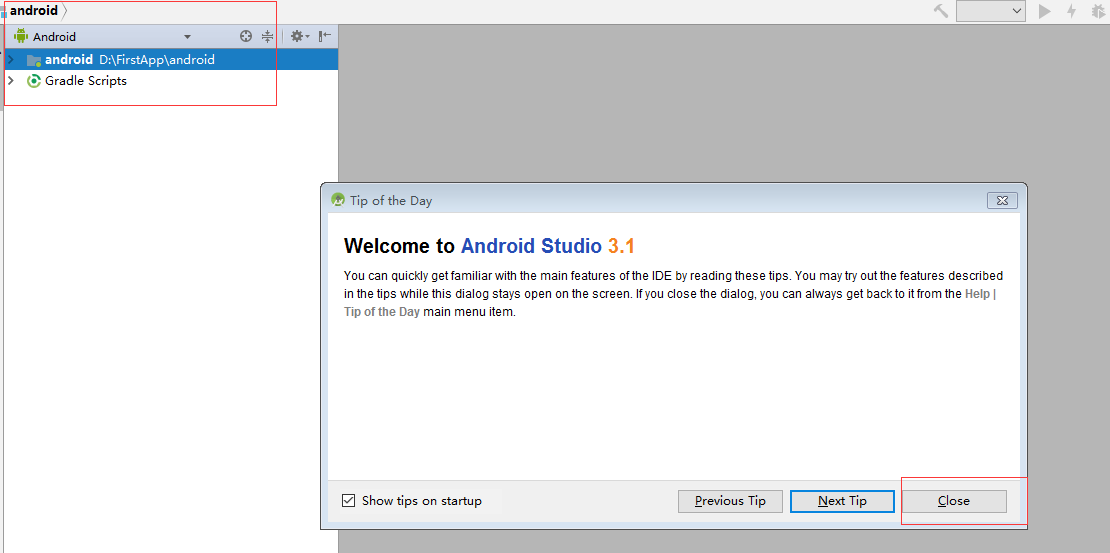
打开项目后，会自动下载缺少的工具包



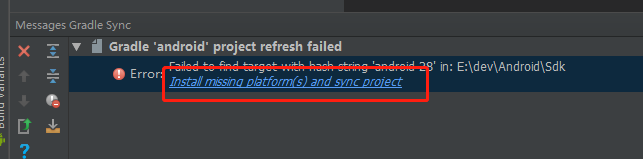
这个工具包下载比较慢，可以强行终止Android Studio的运行，然后把自行下载好的压缩包文件放到指定的目录中然后再打开Android Studio即可



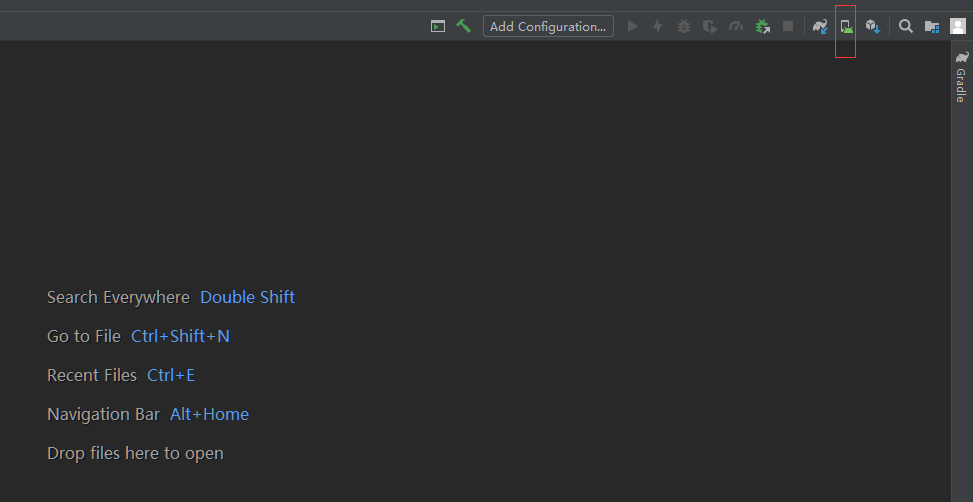
完成



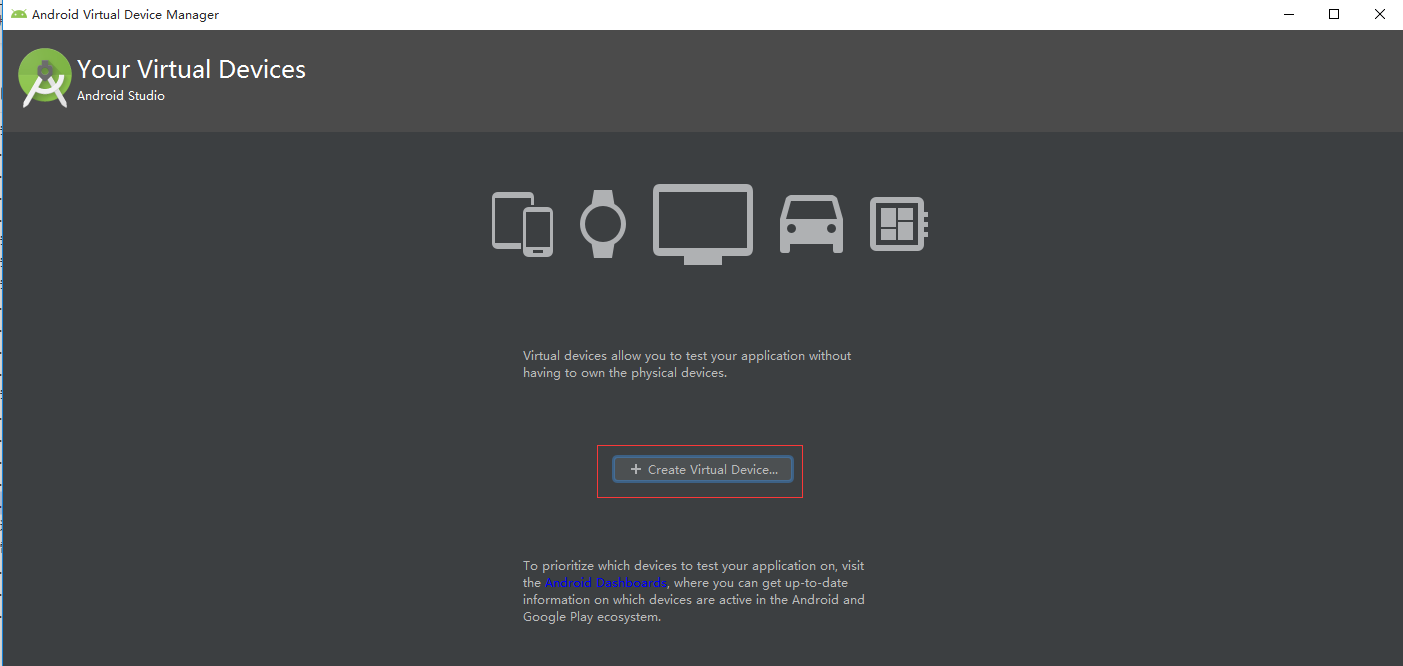
如果打开项目后出现类似这样的带有蓝色下划线的错误提示，那么就点击蓝色的链接进行安装确实的包：



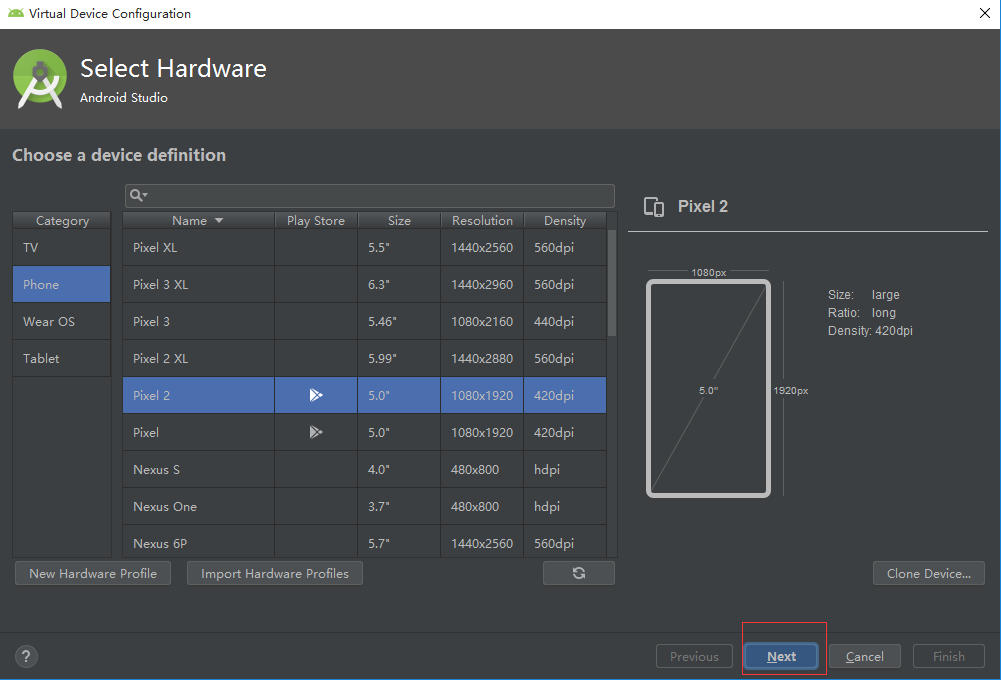
接下来开始配置Android Studio中的模拟器



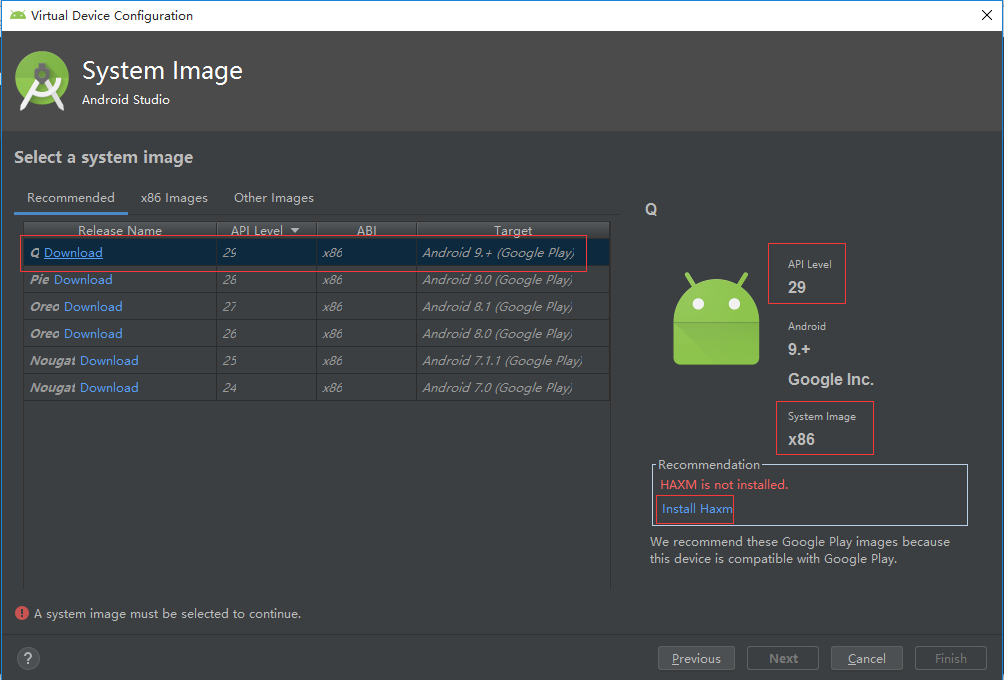
点击创建虚拟设备



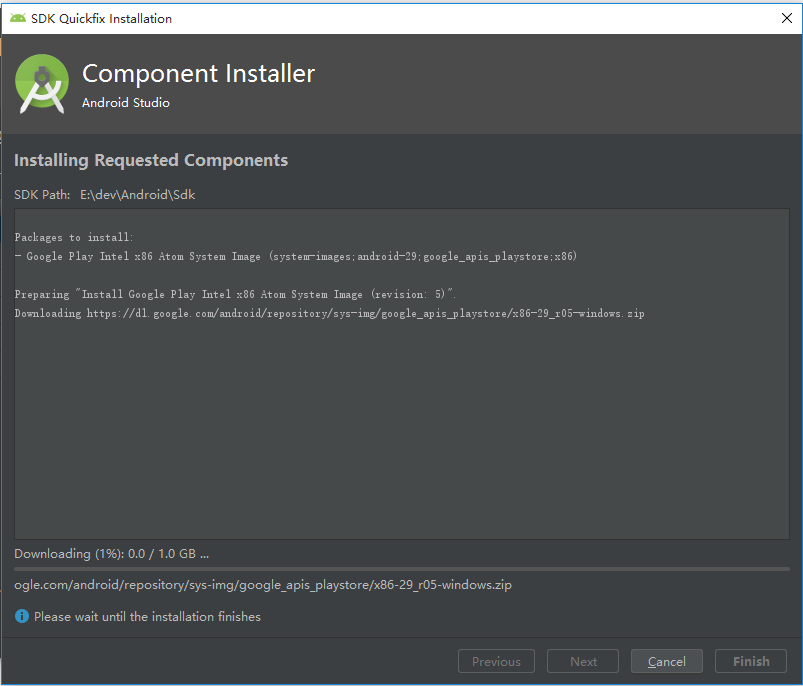
选择模拟设备，默认 next



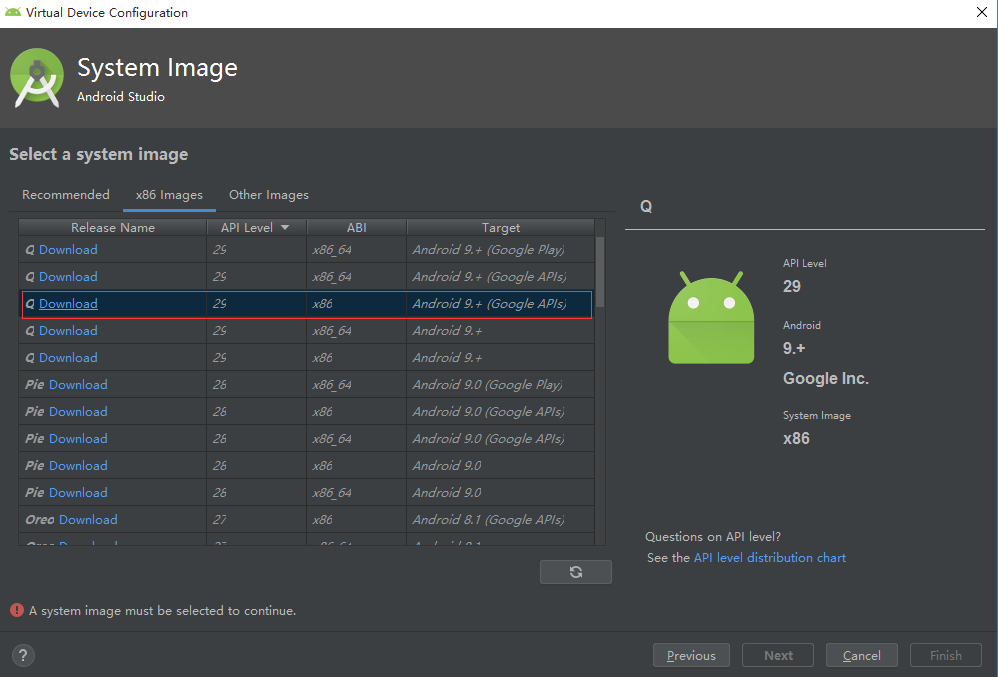
下载api29，并且安装Haxm



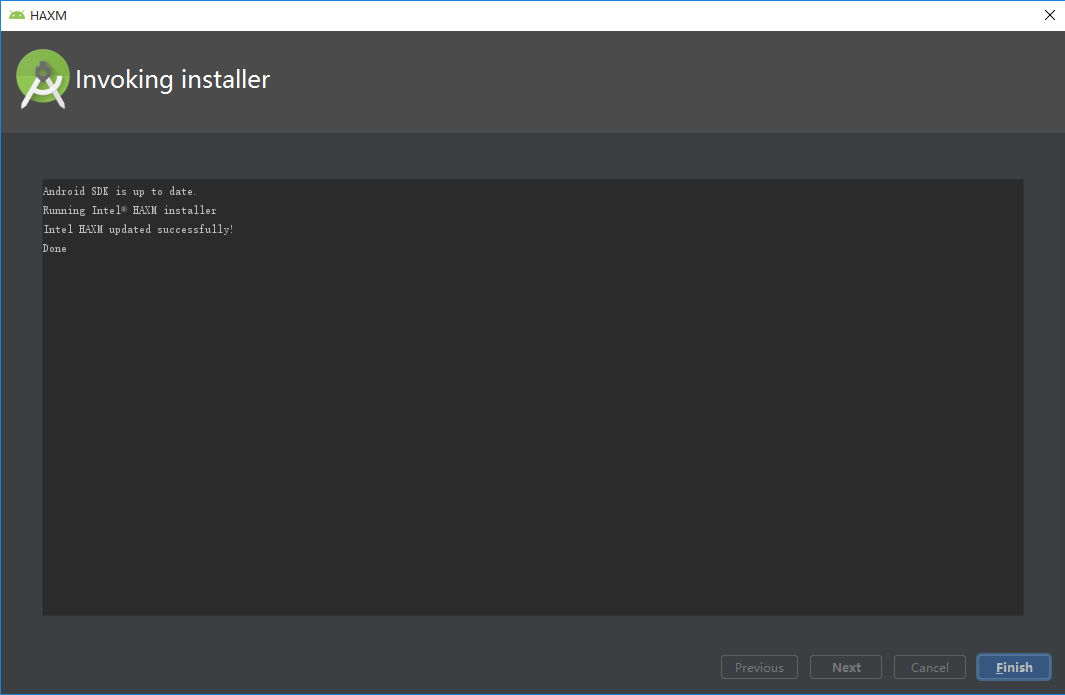
下载中



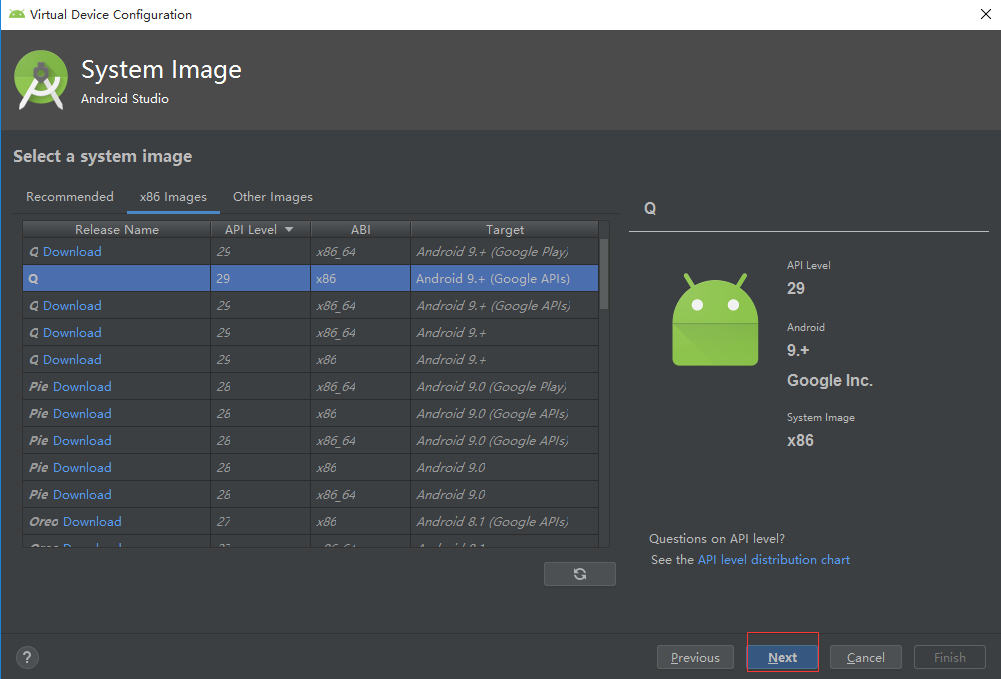
完成，下一步安装x86 images



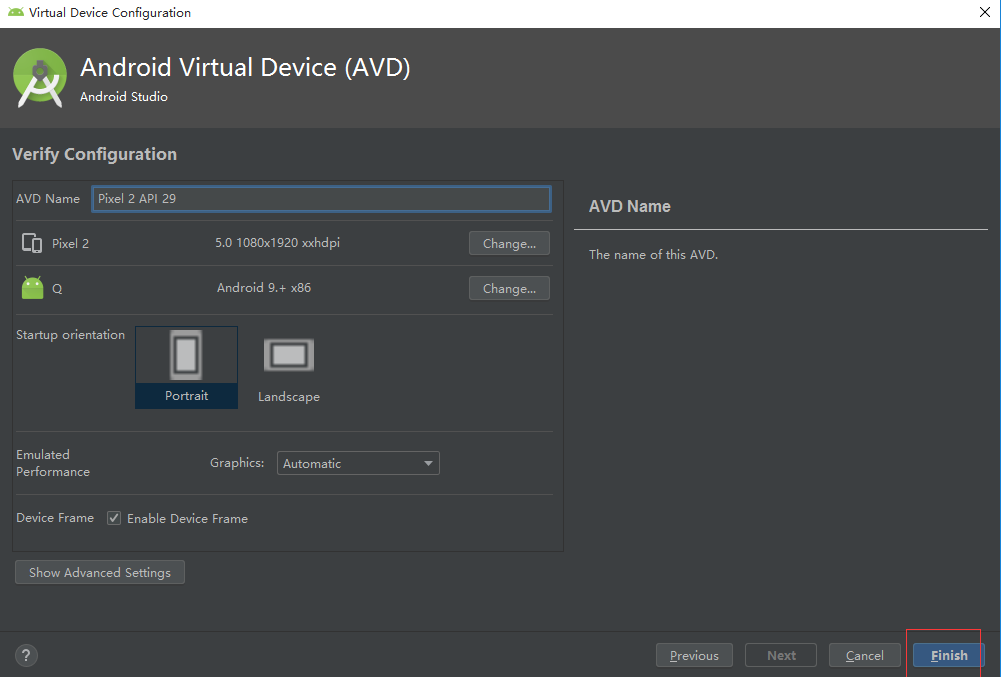
安装Haxm



点击next下一步



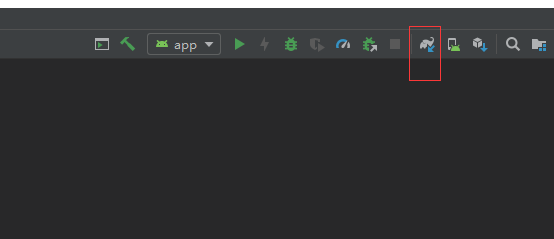
点击Finish完成



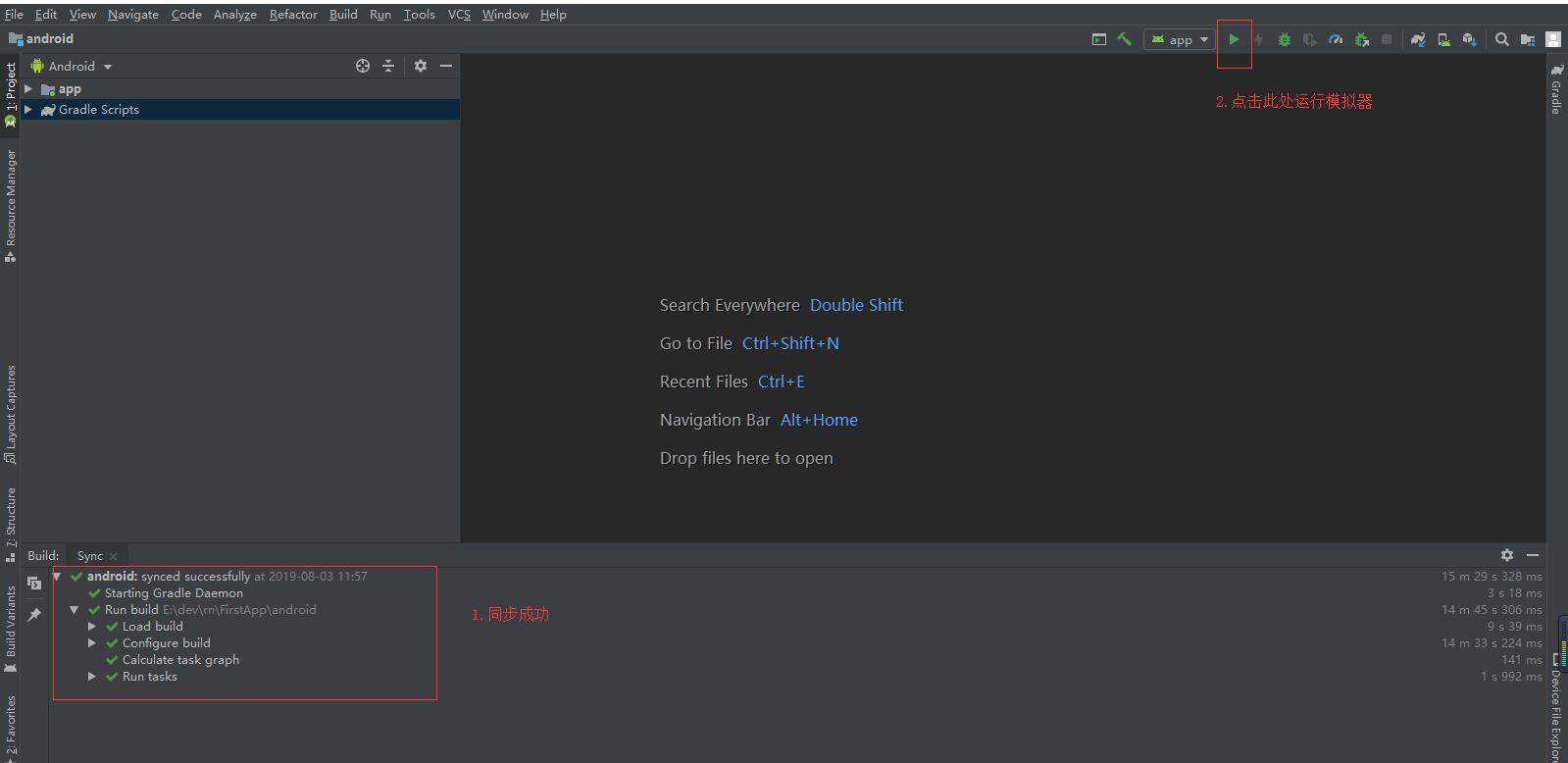
# 运行项目

1. 运行模拟器

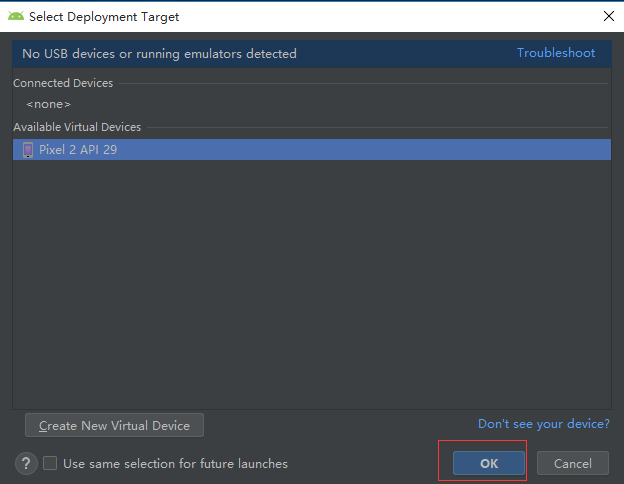
安装好模拟器后，在Android Studio界面上点击sync按钮来同步项目



等待同步完成后，界面上会显示绿色三角按钮，点击此按钮即可运行一个手机模拟器



选择模拟器



点击OK

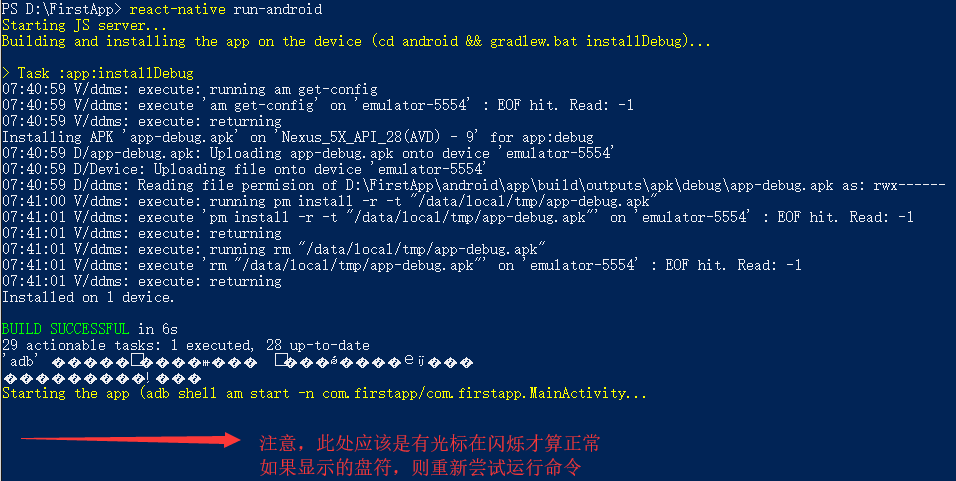
如果环境安装没有问题，则会在电脑屏幕上显示一个手机



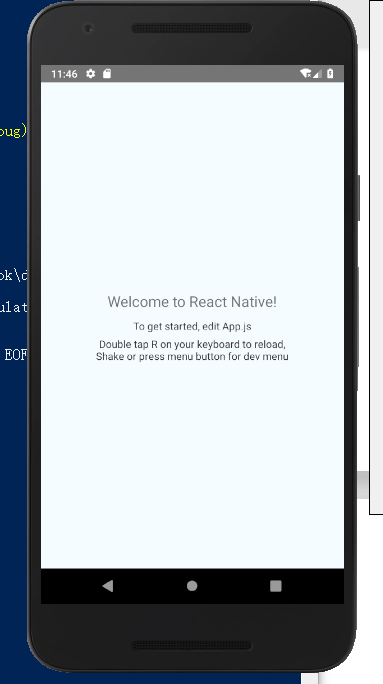
然后在项目根目录运行以下命令

react-native run-android

初次运行时会比较慢，在命令行中效果如下：



模拟器中显示一个默认的欢迎界面



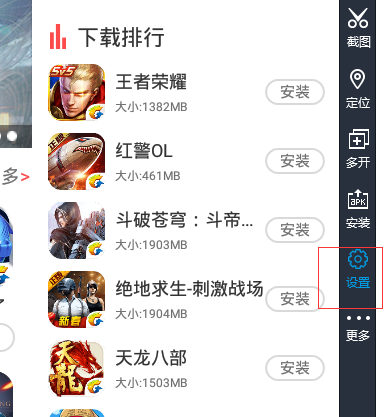
# 雷电模拟器的使用

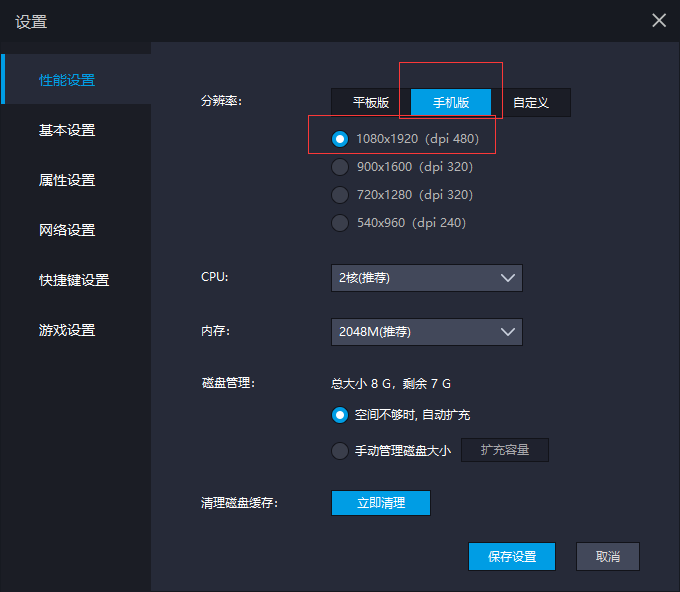
Android Studio自带的模拟器会非常占用cpu资源，造成电脑卡顿。所以，一般不建议使用。

百度搜索”雷电模拟器”，下载安装，此处省略安装过程。

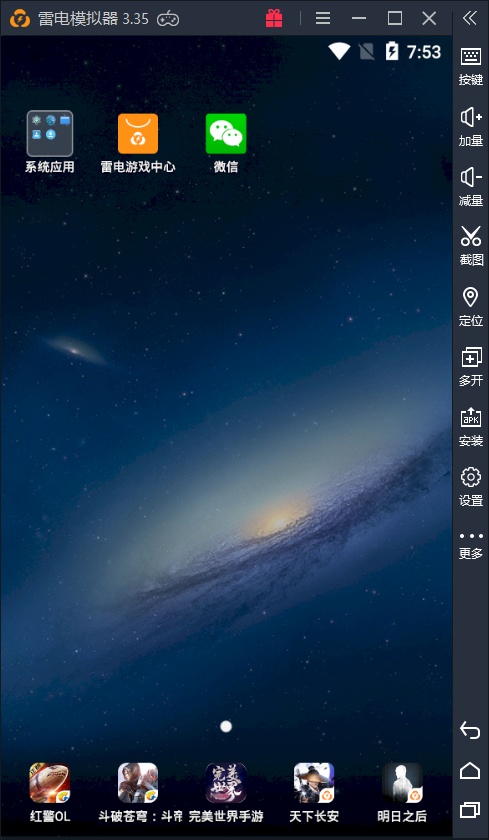
## 设置

设置成手机模式：





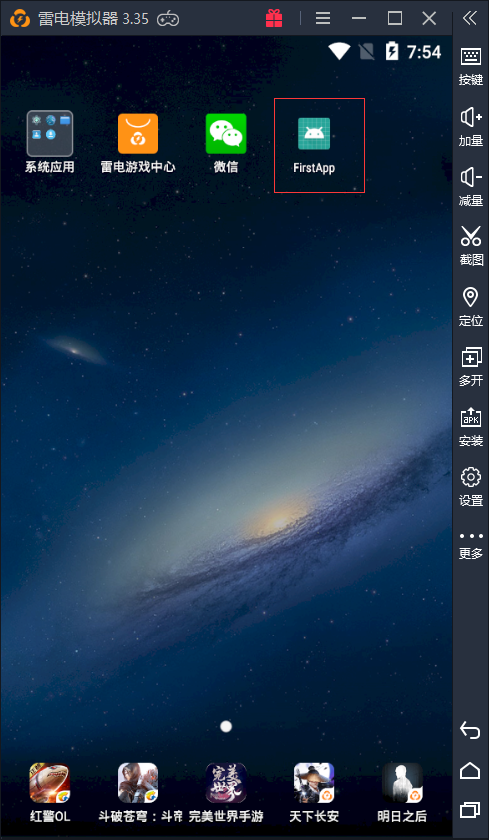
重启生效。



## 运行项目

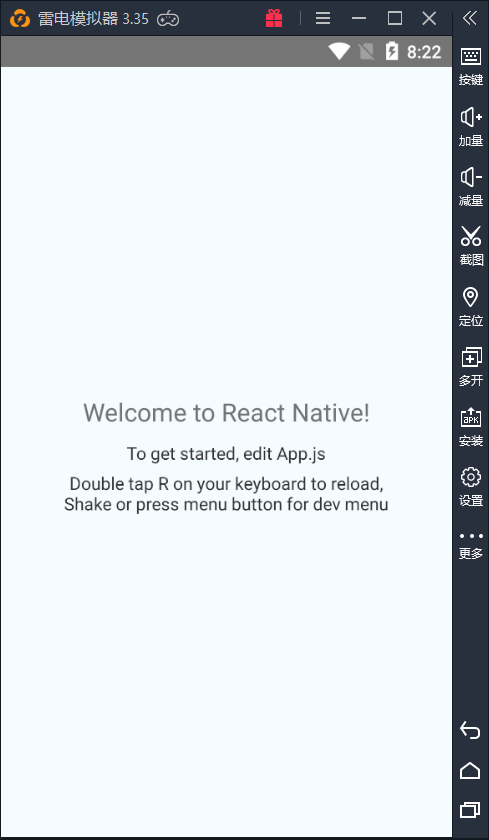
在项目根目录运行react-native run-android命令

会自动在雷电模拟器中添加一个App项目



双击打开。

双击r键即可刷新，出现如下界面



到此时，React-Native开发环境就配置好了！！！