了解移动端应用目前开发现状

一、移动App开发基本介绍

智能手机应用开发在最初开发时采用的是原生开发，如iOS平台使用OC或现在的swift开发，android使用android SDK（java）等技术开发，在市场上确实取得了良好的效果，也为用户带来了很多精美上的手机应用，但是随着时间的发展。原生手机应用开发出现了以下问题：

· 开发者学习成本高

· 应用版本迭代效率极低

· 开发效率慢

· 同一段逻辑,需要使用不同语言实现

市场发现，原生应用开发虽然给用户提供了非常良好的应用体验和精美的应用，但是对于市场来说，公司为原生开发也将会提供非常大的成本，于是后来相继出现对于手机应用的Web App、 Hybrid App(Hybrid App（混合模式移动应用）是指介于web-app、native-app这两者之间的app，兼具“Native App良好用户交互体验的优势”和“Web App跨平台开发的优势”)等开发方式，大大提高了开发者开发手机应用的效率，并降低了对手机应用开发的成本，但是遗憾的是，初时这类开发的手机App就手机体验来说，其实非常糟糕。应该如何既能保证手机应用开发效率的同时，又能让用户对手机应用又会有良好的的体验是开发社区一直在讨论的问题。

Facebook公司在2015年开源了一套框架叫做React Native，为当前的手机应用开发市场带来了不一样的开发方式和体验，提出“既拥有Native的用户体验，又保留React的开发方式和效率”。React Native使你只使用JavaScript也能编写原生移动应用。 它在设计原理上和React一致，通过声明式的组件机制来搭建丰富多彩的用户界面。Facebook承认现阶段Native开发仍然是市场必须的，但是也提出“Learn once，write anywhere”的概念，大大降低了开发者入手手机开发的学习成本，经过这么久的市场考验，React Naitve确实很大程度上实现了当初提供的概念，React Naitve由于其以JS库调用底层UI和API的原理确实比其他方式开发手机应用更加优秀，而来自React框架的开发效率也是有目共睹的。

二、环境搭建

1. 安装 Android Studio

首先下载和安装 Android Studio，国内用户可能无法打开官方链接，请自行使用搜索引擎搜索可用的下载链接。安装界面中选择"Custom"选项，确保选中了以下几项：

```
Android SDK
Android SDK Platform
Performance (Intel ® HAXM) (AMD 处理器看这里)
Android Virtual Device
```

然后点击"Next"来安装选中的组件。

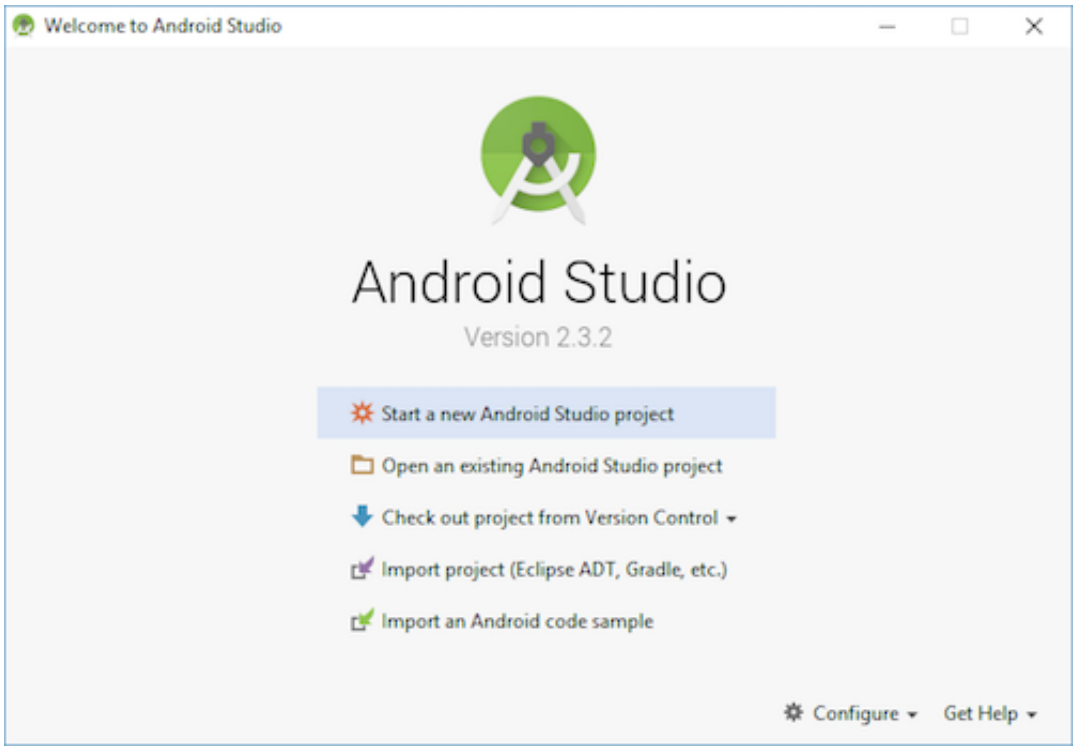
> 如果选择框是灰的，你也可以先跳过，稍后再来安装这些组件。

安装完成后，看到欢迎界面时，就可以进行下面的操作了。

2. 安装 Android SDK

Android Studio 默认会安装最新版本的 Android SDK。目前编译 React Native 应用需要的是`Android 9 (Pie)`版本的 SDK（注意 SDK 版本不等于终端系统版本，RN 目前支持 android4.1 以上设备）。你可以在 Android Studio 的 SDK Manager 中选择安装各版本的 SDK。

你可以在 Android Studio 的欢迎界面中找到 SDK Manager。点击"Configure"，然后就能看到"SDK Manager"。



> SDK Manager 还可以在 Android Studio 的"Preferences"菜单中找到。具体路径是**Appearance & Behavior → System Settings → Android SDK**。

在 SDK Manager 中选择"SDK Platforms"选项卡，然后在右下角勾选"Show Package Details"。展开`Android 9 (Pie)`选项，确保勾选了下面这些组件（重申你必须使用稳定的翻墙工具，否则可能都看不到这个界面）：

- `Android SDK Platform 28`
- `Intel x86 Atom_64 System Image`（官方模拟器镜像文件，使用非官方模拟器不需要安装此组件）

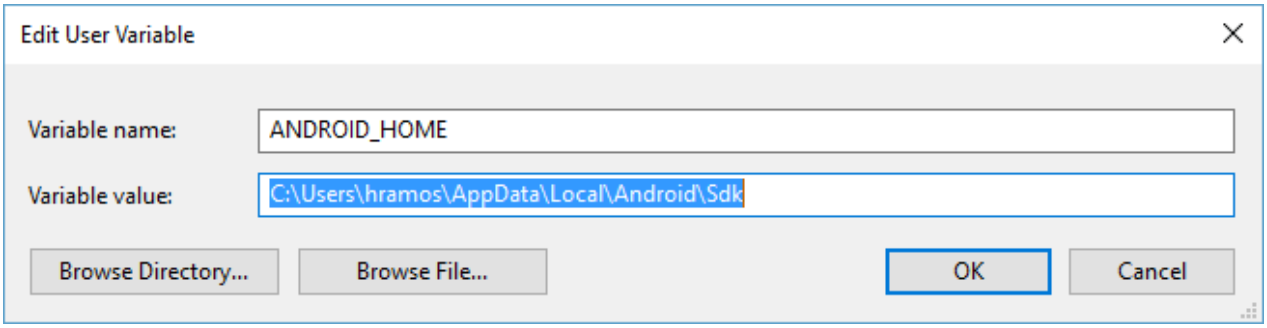
然后点击"SDK Tools"选项卡，同样勾中右下角的"Show Package Details"。展开"Android SDK Build-Tools"选项，确保选中了 React Native 所必须的`28.0.3`版本。你可以同时安装多个其他版本。

最后点击"Apply"来下载和安装这些组件。

3. 配置 ANDROID_HOME 环境变量

React Native 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。

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）：



SDK 默认是安装在下面的目录：

```
c:\Users\你的用户名\AppData\Local\Android\Sdk
```

你可以在 Android Studio 的"Preferences"菜单中查看 SDK 的真实路径，具体是**Appearance & Behavior → System Settings → Android SDK**。

你需要关闭现有的命令符提示窗口然后重新打开，这样新的环境变量才能生效。

4. 把一些工具目录添加到环境变量 Path 中

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，选中**Path**变量，然后点击**编辑**。点击**新建**然后把这些工具目录路径添加进去：platform-tools、emulator、tools、tools/bin

```
%ANDROID_HOME%\platform-tools
%ANDROID_HOME%\emulator
%ANDROID_HOME%\tools
%ANDROID_HOME%\tools\bin
```

三、创建新项目

```
npx react-native init AwesomeProject
```

Windows 用户请注意，请不要在某些权限敏感的目录例如 System32 目录中 init 项目！会有各种权限限制导致不能运行！

四、准备 Android 设备

你需要准备一台 Android 设备来运行 React Native Android 应用。这里所指的设备既可以是真机，也可以是模拟器。

使用 Android 真机

你也可以使用 Android 真机来代替模拟器进行开发，只需用 usb 数据线连接到电脑。

使用 Android 模拟器

你可以使用 Android Studio 打开项目下的"android"目录，然后可以使用"AVD Manager"来查看可用的虚拟设备，它的图标看起来像下面这样：

如果你刚刚才安装 Android Studio，那么可能需要先创建一个虚拟设备。点击"Create Virtual Device..."，然后选择所需的设备类型并点击"Next"，然后选择**Pie** API Level 28 image.

然后点击"Next"和"Finish"来完成虚拟设备的创建。现在你应该可以点击虚拟设备旁的绿色三角按钮来启动它了，启动完后我们可以尝试运行应用。

五、编译并运行 React Native 应用

确保你先运行了模拟器或者连接了真机，然后在你的项目目录中运行`yarn android`或者`yarn react-native run-android`：

```
cd AwesomeProject
yarn android
# 或者
yarn react-native run-android
```

修改项目

现在你已经成功运行了项目，我们可以开始尝试动手改一改了：

- 使用你喜欢的文本编辑器打开`App.js`并随便改上几行
- 按两下 R 键，或是在开发者菜单中选择 *Reload JS*，就可以看到你的最新修改。

react-native中文网网址参考

六、项目开发和组件演示

(新创建的项目/fls项目/官网演示代码项目)

通过在新创建的项目中进行添加组件,进行开发功能点

```
  render() {
    const { dataList } = this.state;
    return (
      <View style={{ flex: 1, backgroundColor: '#FFF', paddingHorizontal: 15
}}>
        <View style={{ flex: 1 }}>
```

```
          <View
            style={{ justifyContent: 'center', height: 40, borderBottomWidth:
0.5, borderColor: '#eae6e4' }}
          >
            <Text>最近下载</Text>
          </View>
          <FlatList
            keyExtractor={this._keyExtractor}
            style={{ paddingLeft: 5, flex: 1 }}
            data={dataList}
            showsVerticalScrollIndicator={false}
            renderItem={this._renderItem}
            ListEmptyComponent={<BlankPrompt onPress={this._refresh} loading=
{this.state.refreshing} />}
          />
        </View>
      </View>
    );
  }
```