

测试账号：

imloniper 11111111

18551624814 /1

在虚拟机内启动项目

1. npm install

2. npm start -- --reset-cache

3. react-native run-android

如果写代码有添加图片的操作或者找不到某些文件，重新加载之后 需要再次执行下
npm start -- --reset-cache

注： 安卓测试机连接电脑时，启动虚拟机会报错。

一：虚拟机内访问开发菜单

MAC：

在iOS模拟器中运行，还可以按下Command⌘ + D 快捷键，Android模拟器对应的则是Command⌘ + M。

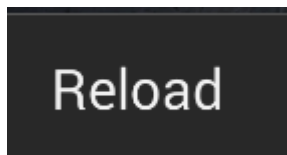
Windows：

按下F1 (/F2) ， 或者



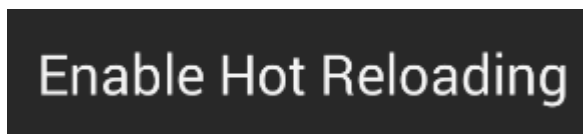
二：刷新JavaScript

1.手动刷新



在iOS模拟器中按下Command⌘ + R，Android模拟器上对应的则是按两下R

2.自动刷新



*****不能刷新JS，需要重新编译应用的情况*****

- 增加了新的资源(比如给iOS的Images.xcassets或是Andorid的res/drawable文件夹添加了图片)
- 更改了任何的原生代码 (objective-c/swift/java)

三：错误信息

1.红屏错误

应用内的报错会以全屏红色显示在应用中（调试模式下），我们称为红屏（red box）报错。你可以使用`console.error()`来手动触发红屏错误。

2.黄屏警告

应用内的警告会以全屏黄色显示在应用中（调试模式下），我们称为黄屏（yellow box）报错。点击警告可以查看详情或是忽略掉。和红屏报警类似，你可以使用`console.warn()`来手动触发黄屏警告。在默认情况下，开发模式中启用了黄屏警告。

四：Chrome开发者工具

Debug JS Remotely

Chrome中并不能直接看到App的用户界面，而只能提供console的输出，以及在sources项中断点调试js脚本。

五：React Developer Tools（支持React Native 0.43及以上版本）

安装命令：`npm install -g react-devtools`

启动命令：`react-devtools`

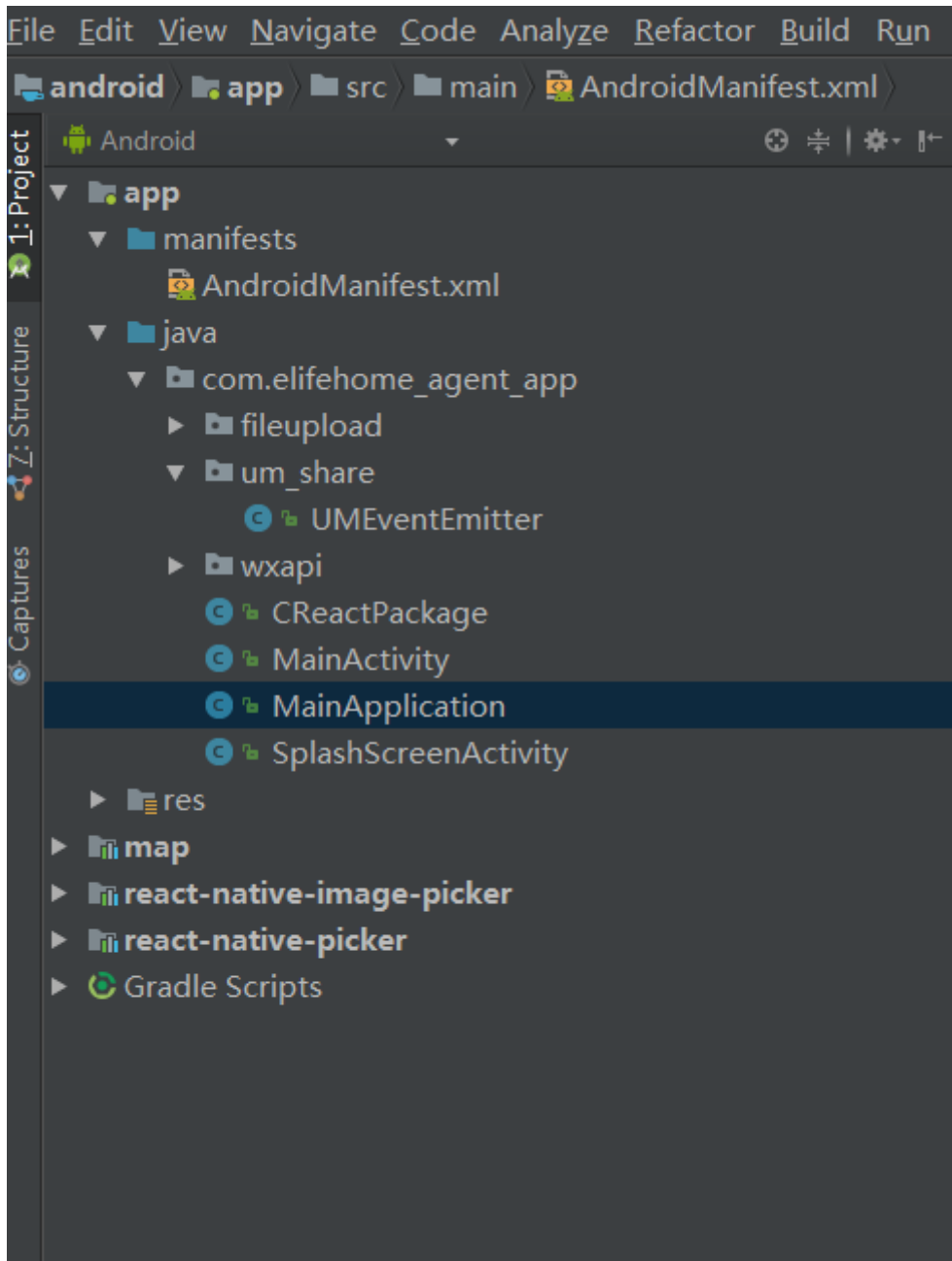
使用 Devtools 进行调试时，可以查看应用程序的 React 组件分层结构，而不是更加神秘的浏览器 DOM 表示。

如果在一个组件的渲染阶段有一个断点，将在“React”选项卡中自动选择。这允许你在渲染树中一步一步看到一个组件如何影响另一个。通过打破错误，你可以很容易地找到哪些组件在渲染过程中抛出一个错误，和什么样的道具导致它。

六：调试原生代码

在和原生代码打交道时（比如编写原生模块），可以直接从Android Studio或是Xcode中启动应用，并利用这些IDE的内置功能来调试（比如设置断点）。

七：在真机上进行调试



AndroidManifest.xml中的一些配置：

```
24      <application
25          android:name=".MainApplication"
26          android:allowBackup="true"
27          android:icon="@mipmap/icon"
28          android:label="斑马家政系统"
29          android:theme="@style/AppTheme">
```

.MainApplication：项目入口，相当于main函数

在Android Studio启动项目时，通过index.android.js进入

真机调试步骤：

1.启动本地服务器 `npm start -- --reset-cache (intellij)`

2.Android Studio打开android文件夹，配置容器（类似于tomcat）

3.连接手机（开发者选项权限！！！全部放开）

--注意事项1.打开USB调试模式 2.选择可以调试的软件-类似于APP信任该设备

4.Studio启动debug模式

4.调试时直接晃动手机就可以开启调试选项

过程中遇到错误：

真机红屏：（需要配置adb，查看adb:命令提示板adb）

adb reverse tcp:8081 tcp:8081

adb:给手机服务设置监听

ADB的全称为Android Debug Bridge，就是起到调试桥的作用。ADB是android sdk（Android专属的软件开发工具包）里的一个工具，用这个工具可以直接操作管理android模拟器或者真实的android设备(如DEFY)。它的主要功能有：

- 运行设备的shell(命令行)
- 管理模拟器或设备的端口映射
- 计算机和设备之间上传/下载文件
- 将本地apk软件安装至模拟器或android设备

真机调试好处：

模拟机有时候测不出来问题，就需要直接在真机上进行调试。

八：项目打包

```
react-native bundle --platform ios --entry-file code-publish/build-js/$1.ios.js --  
bundle-output ./bundles/$1.$2.ios.bundle --dev false
```

```
react-native bundle --platform android --entry-file code-publish/build-  
js/$1.android.js --bundle-output ./bundles/$1.$2.android.bundle --dev false
```