# 概括

紫金扫码app使用的主框架是React-Native（以下简称RN），采用的开发模式是完全内接h5模式，h5部分掌管页面内容渲染，RN端掌管网络请求、数据转换与交互。利用核心模组react-native-webview内嵌本地h5文件实现视图链接。

# H5部分

## 框架介绍

H5页面的开发现在采用的是create-react-app（以下简称cra）脚手架，即react前端框架。理论上使用任何前端开发框架开发的页面都可以嵌入到项目当中。然而需要着重注意的是前端部分的交互过程需要额外预留接口。核心通讯方案：web端使用window.onMessage方法接收RN端回传数据，window.postMessage方法向RN端发送数据。RN端使用react-native-webview的onMessage属性监听web端回传数据，使用其postMessage方法向web端发送数据。

## 环境配置

这里使用cra只需要安装Nodejs作为包管理器与本地伺服的基础环境。

1. 安装NodeJs

请自行在网络搜索NodeJs下载并安装。

1. 安装create-react-app

选择一个合适的文件夹位置按shift+鼠标右键选择在此处使用powershell，然后输入命令npm install create-react-app。

新版本NodeJs已经内置了npm，所以无需单独安装npm。注意使用npm安装包时由于网络管理政策，需要配置npm安装地址（不配置则可能导致安装缓慢或极端缓慢甚至失败）。可直接输入命令npm config set registry <https://registry.npm.taobao.org/> 以改变安装地址。

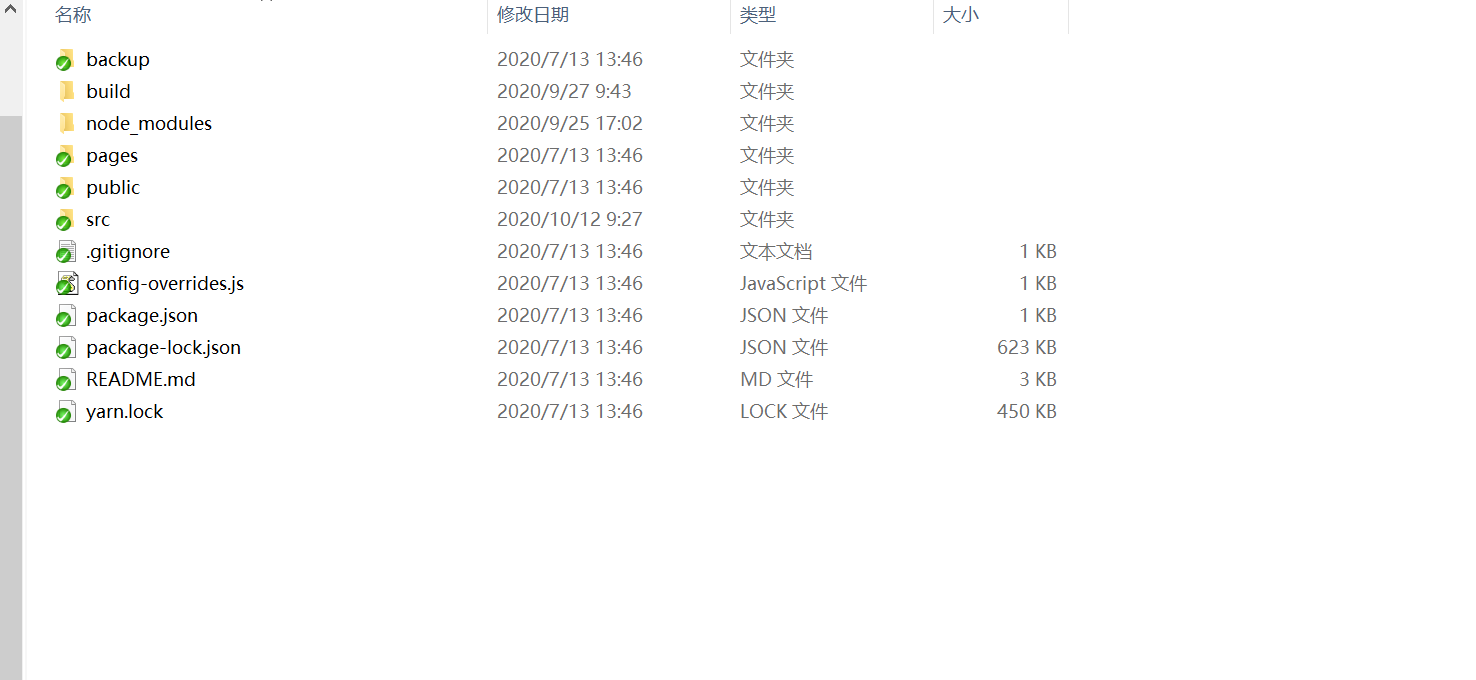
源码中不包含node\_modules文件夹，需要用户手动安装，执行命令npm install 即可安装所有依赖包。

至此基本环境已安装完成，可以启动h5部分以查看项目页面。

H5页面源码位置



## 文件夹说明



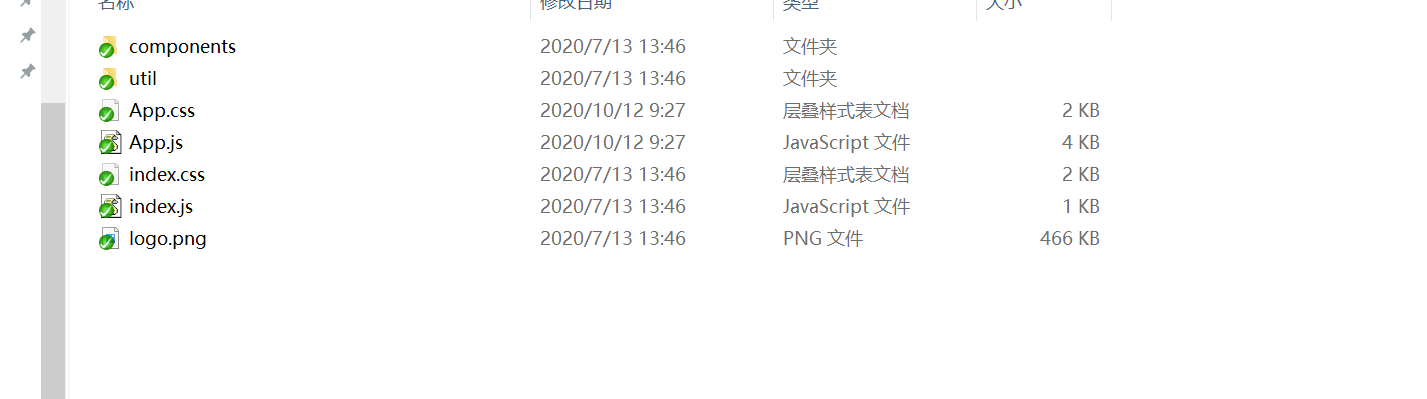
build => 打包完成后的页面存放点

node\_modules => 所有依赖的包文件（不推荐复制带走，过于繁杂耗时费力，用时可自行安装）

pages => 所有页面文件存放位置

src => 运行时所需文件存放位置

Src文件夹内部



Components => 所有页面的通用组件部分

Util => 所有页面的通用工具函数部分

App.css 、App.js => 页面主体部分

日后页面的再开发只需将pages文件夹内的App.css 、App.js文件于此处替换即可实现。

在apph5 内部位置打开powershell输入命令 npm run build即可实现打包页面文件。

打包之后的h5文件替换掉RN内的位置 \rtzl\_smapp\android\app\src\main\assets\h5 即可。

# RN部分

## 环境配置

RN的开发环境配置稍显复杂，请直接观看RN官网或RN中文网。

官网：https://reactnative.dev/docs/getting-started

中文网：<https://reactnative.cn/docs/getting-started>

核心需求：

Nodejs

Jdk 1.8

Android Studio

Python2.x

安装完Android Studio之后需要注意安装相关sdk，和配置环境变量。

## 安装依赖

必须安装的依赖有：Node、JDK 和 Android Studio。

虽然你可以使用任何编辑器来开发应用（编写 js 代码），但你仍然必须安装 Android Studio 来获得编译 Android 应用所需的工具和环境。

### Node, JDK

我们建议直接使用搜索引擎搜索下载 Node 和[Java SE Development Kit (JDK)](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html)

注意 Node 的版本应大于等于 12，而 JDK 的版本必须是 1.8（目前不支持 1.9 及更高版本，注意 1.8 版本官方也直接称 8 版本）。安装完 Node 后建议设置 npm 镜像（淘宝源）以加速后面的过程（或使用科学上网工具）。

注意：不要使用 cnpm！cnpm 安装的模块路径比较奇怪，packager 不能正常识别！

# 使用nrm工具切换淘宝源

npx nrm use taobao

# 如果之后需要切换回官方源可使用

npx nrm use npm

### Yarn

[Yarn](http://yarnpkg.com/)是 Facebook 提供的替代 npm 的工具，可以加速 node 模块的下载。

npm install -g yarn

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。

### Android 开发环境

如果你之前没有接触过 Android 的开发环境，那么请做好心理准备，这一过程相当繁琐。请万分仔细地阅读下面的说明，严格对照文档进行配置操作。

译注：请注意！！！国内用户必须必须必须有稳定的翻墙工具，否则在下载、安装、配置过程中会不断遭遇链接超时或断开，无法进行开发工作。某些翻墙工具可能只提供浏览器的代理功能，或只针对特定网站代理等等，请自行研究配置或更换其他软件。总之如果报错中出现有网址，那么 99% 就是无法正常翻墙。

#### 1. 安装 Android Studio

[首先下载和安装 Android Studio](https://developer.android.com/studio/index.html)，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](https://android-developers.googleblog.com/2018/07/android-emulator-amd-processor-hyper-v.html))
* Android Virtual Device

然后点击"Next"来安装选中的组件。

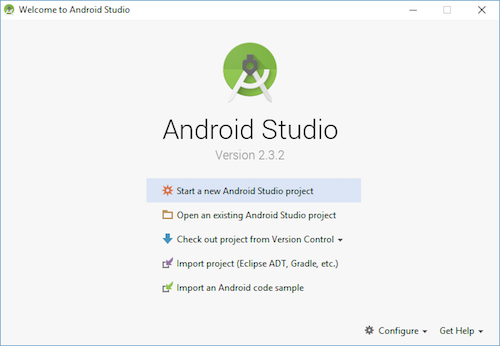
如果选择框是灰的，你也可以先跳过，稍后再来安装这些组件。

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 10 (Q)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 10 (Q)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 29
* 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）

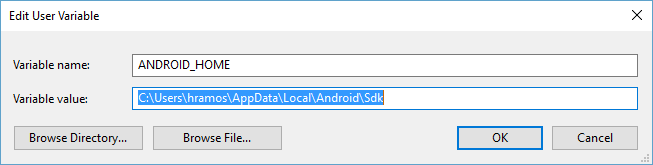
然后点击"SDK Tools"选项卡，同样勾中右下角的"Show Package Details"。展开"Android SDK Build-Tools"选项，确保选中了 React Native 所必须的29.0.2版本。你可以同时安装多个其他版本。

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

## 创建新项目

如果你之前全局安装过旧的react-native-cli命令行工具，请使用npm uninstall -g react-native-cli卸载掉它以避免一些冲突。

使用 React Native 内建的命令行工具来创建一个名为"AwesomeProject"的新项目。这个命令行工具不需要安装，可以直接用 node 自带的npx命令来使用（注意 init 命令默认会创建最新的版本）：

**注意**：请不要单独使用常见的关键字作为项目名（如 class, native, new, package 等等）。请不要使用与核心模块同名的项目名（如 react, react-native 等）。请不要在目录、文件名中使用中文、空格等特殊符号。

npx react-native init AwesomeProject

提示：你可以使用--version参数（注意是两个杠）创建指定版本的项目。例如npx react-native init MyApp --version 0.44.3。注意版本号必须精确到两个小数点。

**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！**

如果你是想把 React Native 集成到现有的原生项目中，则步骤完全不同，请参考[集成到现有原生应用](https://reactnative.cn/docs/integration-with-existing-apps)。

## 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。后面我们所有的文档除非特别说明，并不区分真机或者模拟器。Android 官方提供了名为 Android Virtual Device（简称 AVD）的模拟器。此外还有很多第三方提供的模拟器如[Genymotion](https://www.genymotion.com/download)、BlueStack 等。一般来说官方模拟器免费、功能完整，但性能较差。第三方模拟器性能较好，但可能需要付费，或带有广告。

### 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，然后遵照[在设备上运行](https://reactnative.cn/docs/running-on-device)这篇文档的说明操作即可。

### 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：

Android Studio AVD Manager

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](https://developer.android.com/studio/run/managing-avds.html)。点击"Create Virtual Device..."，然后选择所需的设备类型并点击"Next"，然后选择**Q** API Level 29 image.

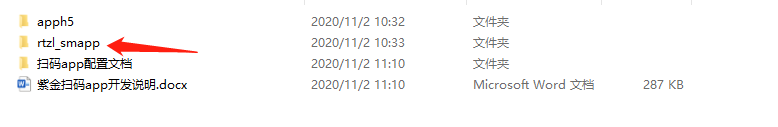
译注：请不要轻易点击 Android Studio 中可能弹出的建议更新项目中某依赖项的建议，否则可能导致无法运行。

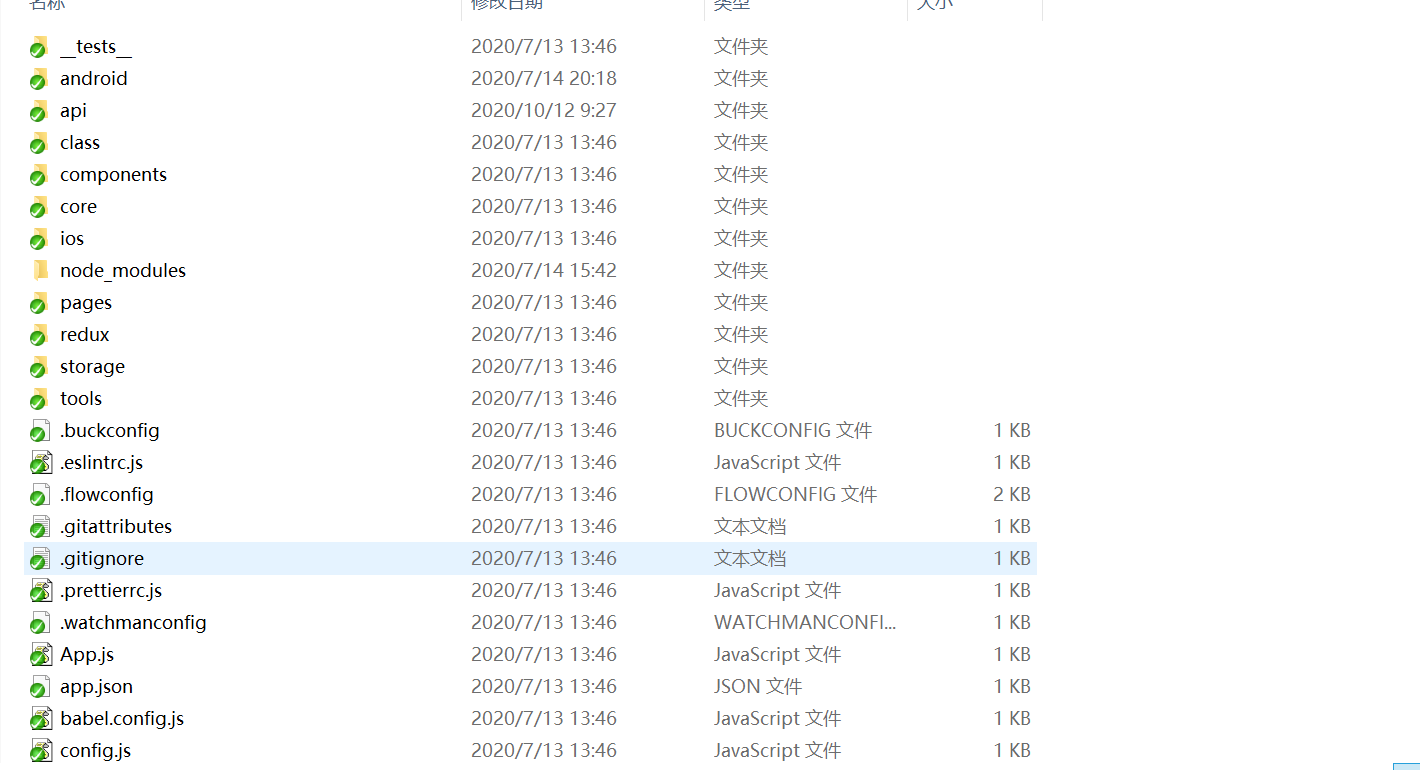
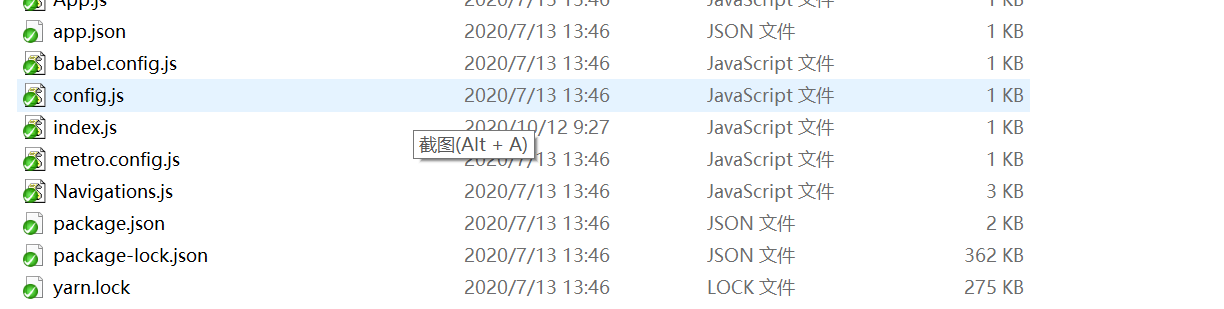
如果你还没有安装 HAXM（Intel 虚拟硬件加速驱动），则先点击"Install HAXM"或是按这篇[文档说明](https://software.intel.com/en-us/android/articles/installation-instructions-for-intel-hardware-accelerated-execution-manager-windows)来进行安装。

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。

## 文件说明

RN端源码



Android => 安卓相关部分

Api => 所有请求接口部分

Components => RN端所有通用组件

Core => 核心模块

Pages => 所有页面入口

Redux => redux全局数据相关部分

Storage => 本地存储部分

Tools => 页面通用工具部分

App.js => app 主入口文件

Config.js => 一键配置文件

Index.js => 程序最初入口

Navigations.js => 全页面导航注册文件

在此位置运行命令 npm run go 即可运行测试

运行 npm run build 可打包.apk文件，打包好的文件位于 \rtzl\_smapp\android\app\build\outputs\apk\release\app-release.apk