

[\[关闭\]](#)

@summer88123 2017-07-03 16:55 字数 2812 阅读 40

React Native 知识分享

- [React Native 知识分享](#)
- [learn once,write everywhere](#)
 - [Native 运行环境安装](#)
 - [React Native 运行环境](#)
 - [React Native 简介](#)
 - [创建项目](#)
 - [运行与调试](#)
 - [打包](#)
 - [IDE](#)
 - [React Native UI](#)
 - [flexbox布局](#)
 - [Touchable系列组件](#)
 - [滚动视图与列表](#)
 - [特定平台代码](#)
 - [Component生命周期](#)
 - [React Native 不足](#)

learn once,write everywhere

Native 运行环境安装

推荐文档: [RN官方文档](#), [RN中文文档](#)

PS:前端同学Native环境推荐使用ios, Xcode安装体验比较好。上面文档中Android环境搭建有些过时。

React Native 运行环境

- 安装Node.js

```
brew install node
```
- 安装watchman

```
brew install watchman
```
- 安装RN命令行

```
npm install react-native-cli
```

React Native 简介

创建项目

```
react-native init <ProjectName>
```

项目目录结构

packageName	
--android	原生android项目目录
--ios	原生ios项目目录
--index.android.js	android入口文件
--index.ios.js	ios入口文件
--__test__	测试文件目录
--package.json	npm依赖配置文件
--node_modules	npm第三方包目录

RN对于原生代码与js代码的相对位置并无要求。运行时原生代码通过加载js打包成的bundle文件与js代码通讯。Debug模式下，bundle文件位于开发机上的Node服务中；Release模式下bundle文件位于App安装包中。

运行与调试

```
react-native run-android
react-native run-ios
```

查看日志

```
react-native log-android
react-native log-ios
```

如果遇到ANDROID_HOME环境变量设置问题，参考[这里](#)

Dev Menu

Android模拟器：cmd+M

Android真机：摇一摇

Ios模拟器：cmd+D

Reload

重新打包并加载bundle文件

在Android模拟器中双击R，在Ios模拟器中cmd+R可快速reload

Hot Reloading

对bundle文件进行增量更新

Ps: 不会触发mount相关的生命周期函数

Debug JS Remotely

在Chorme中调试JS代码

Dev Settings->Debug server host & port for device

使用真机调试时，用于设置JS bundle文件的访问地址，内容为：192.168.x.x:8081。

Mac下可通过命令ifconfig查看ip。

打包

打包的过程分为两部分，首先通过react-native bundle命令将js文件打包并输出到指定位置，其次在编译原生安装包。下面是一段完整的js打包命令

```
react-native bundle --platform android --dev false --entry-file index.android.js \
--bundle-output android/app/src/main/assets/index.android.bundle \
--assets-dest android/app/src/main/res/
```

platform —— 指定平台，ios/android

dev —— 指定是否为开发模式

entry-file —— 指定入口文件

bundle-output —— bundle文件输出位置

assets-dest —— 资源文件输出位置

IDE

推荐使用WebStorm EAP，[下载地址](#)

React Native UI

flexbox布局

flexDirection

指定组件布局的主轴方向，可选row,column，默认column

justifyContent

指定组件子元素沿主轴的排列方式，可选flex-start、center、flex-end、space-around以及space-between。

alignItems

指定组件子元素沿次轴的排列方式，可选flex-start、center、flex-end以及stretch。

Touchable系列组件

TouchableHighlight

TouchableOpacity

TouchableWithoutFeedback

TouchableNativeFeedback

滚动视图与列表

ScrollView

FlatView

特定平台代码

后缀名方式

React Native会检测某个文件是否具有.ios或是.android的扩展名，然后根据当前运行的平台加载正确对应的文件。

假设你的项目中有如下两个文件：

```
BigButton.ios.js
BigButton.android.js
```

这样命名组件后你就可以在其他组件中直接引用，而无需关心当前运行的平台是哪个。

```
import BigButton from './components/BigButton';
```

React Native会根据运行平台的不同引入正确对应的组件。

Platform模块

```
import {Platform} from 'react-native'
```

Platform.OS在iOS上会返回ios，而在Android设备或模拟器上则会返回android。

还有个实用的方法是Platform.select()，它可以以Platform.OS为key，从传入的对象中返回对应平台的值，见下面的示例：

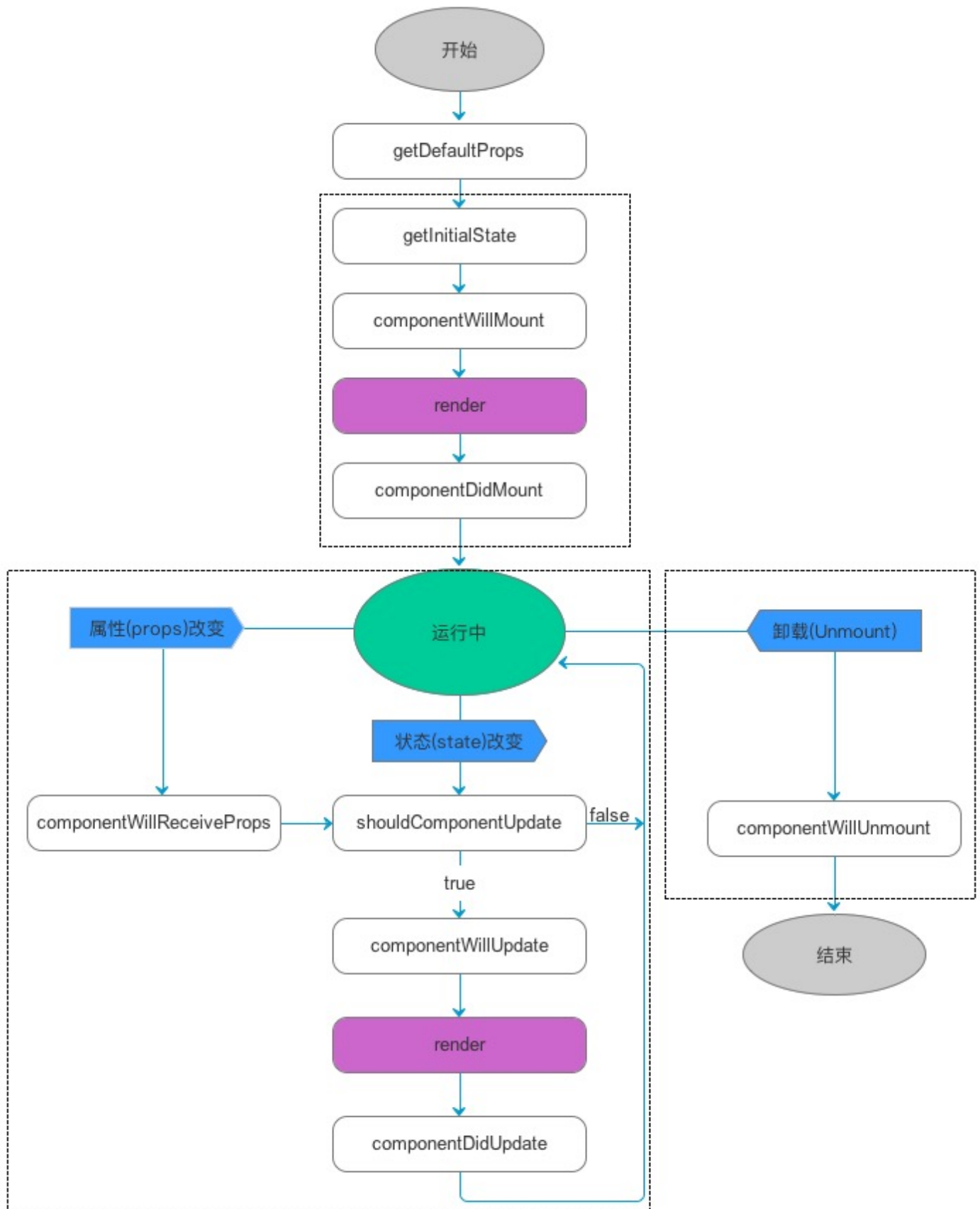
```
1.  import { Platform, StyleSheet } from 'react-native';
2.
3.  var styles = StyleSheet.create({
4.    container: {
5.      flex: 1,
6.      ...Platform.select({
7.        ios: {
8.          backgroundColor: 'red',
9.        },
10.       android: {
11.         backgroundColor: 'blue',
12.       },
13.     }),
14.   },
15. });
```

上面的代码会根据平台的不同返回不同的container样式——iOS上背景色为红色，而android为蓝色。

这一方法可以接受任何合法类型的参数，因此你也可以直接用它针对不同平台返回不同的组件，像下面这样：

```
1.  var Component = Platform.select({
2.    ios: () => require('ComponentIOS'),
3.    android: () => require('ComponentAndroid'),
4.  })();
5.
6.  <Component />;
```

Component生命周期



React Native 不足

- UI组件的平台差异性比较明显。
- 仍然处于快速迭代版本中，向前兼容性不好。
- 第三方库缺乏生产环境考验，甚至不能work。

参考资料

[RN官方文档](#)

[RN中文文档](#)

[《ECMAScript 6 入门》](#)

[React Native 组件的生命周期](#)

+

• 内容目录

-
- [React Native 知识分享](#)
 - [learn once,write everywhere](#)
 - [Native 运行环境安装](#)
 - [React Native 运行环境](#)
 - [React Native 简介](#)
 - [创建项目](#)
 - [运行与调试](#)
 - [打包](#)
 - [IDE](#)
 - [React Native UI](#)
 - [flexbox布局](#)
 - [Touchable系列组件](#)
 - [滚动视图与列表](#)
 - [特定平台代码](#)
 - [Component生命周期](#)
 - [React Native 不足](#)

•

- ◦
 - [RxJava 1](#)
 - [RxJava简介](#)
- - 未分类 3
 - [React Native 知识分享](#)
 - [云构开发者中心文档提纲](#)
 - [单元测试](#)
-
- 以下【标签】将用于标记这篇文稿：

•

•

•

- ◦ [下载客户端](#)
- [关注开发者](#)
- [报告问题, 建议](#)

- [联系我们](#)



添加新批注



保存

取消

在作者公开此批注前，只有你和作者可见。



保存

取消



修改

保存

取消

删除

- 私有
- 公开
- 删除

查看更早的 5 条回复

回复批注

x

通知

取消

确认

- ☐
- ☐