

# ReactNative入门

## 简介

ReactNative旨在开发过程的高效，它具有一些原生App所不具备的优势，它的跨平台性非常强，代码的复用程度非常的高，并且可以具有热更新的能力，其次就是它的社区也在日趋的强大起来。

## 为什么选择ReactNative

纵观国内的比较大的厂商并没有完全的转移到React Native，但也没有完全的排斥说不搞React Native，所以学习这样一个新的技能对我们有百利而无一害，在学习React Native的过程中，我们也会接触到React这样一个新的UI方案，同时对一些ES6，ES7新的特性也都会有接触，总体感觉还不错～

## 入门

学习React Native最好还是要有一些React JavaScript 以及原生应用的开发经验，当然没有也无妨，慢慢一步一步的来嘛。

目前我介绍的是针对Android端介绍的，后续也会把ios相关的内容补上。

需要有的软件：

- Node.js
- Android studio
- Watchman

运行：

```
react-native init AwesomeProject
cd AwesomeProject
react-native run-android
```

## 知识点

项目的入口：index.android.js / index.ios.js

视图呈现：

```
render() {
  return (
    <View style={cStyles.container}>
      <View style={styles.row}>
        <Text style={styles.label}>AAA</Text>
        <View style={styles.rowCount}>
          <Text style={styles.countUnit}>BBB</Text>
          <Text style={styles.price}>CCC</Text>
        </View>
      </View>
      <LinkRow label="DDD" onPress={() => {
      }}/>
      <LinkRow label="EEE" onPress={this.logout}/>
    </View>
  );
}
```

我们的视图需要写在return()下面，并且这个下面只能有一个根View

抽出公共控件：

在界面中，可能我们有很多控件是可以复用的，如果我们不把它们抽出来的话，可能会导致代码的臃肿，并且不是很利于维护，所以我们应该将公共部分抽出来。

将公用的style抽象出来

同上

## 推荐编译器

---

- 需要测试性能的时候需要使用**Android studio**
- 与性能无关时，编写代码时还是推荐使用**WebStorm**编写代码

好了，就先介绍到这里吧，在等到我有时间的时候，可以介绍一下，手写一个**tab**，或者实现登录注册的全部逻辑(包含网络框架)。