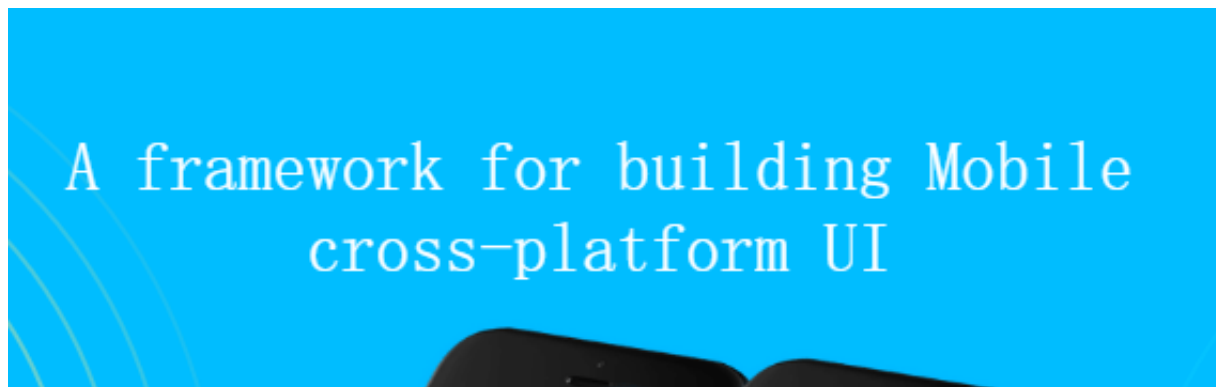


## 1、Weex是什么？

官网：<https://weex-project.io/>

github:<https://github.com/alibaba/weex>

A framework for building Mobile cross-platform UI. (一套构建高性能、可扩展的原生应用跨平台开发方案。)



Weex能够完美兼顾性能与动态性，让移动开发者通过简捷的前端语法写出Native级别的性能体验，并支持iOS、Android及Web等多端部署。

对于移动开发者来说，Weex主要解决了频繁发版本和多端研发两个痛点，同时解决了前端语言性能差和显示效果受限的问题。

## 2、为什么选择Weex

参考：<http://blog.csdn.net/lzyzsd/article/details/51991807>

以下拿Weex与ReactNative做比较。

### JS引擎：

weex使用V8，ReactNative使用JSCore。

### JS开发框架：

weex基于vue.js(2W+ star)。小巧轻量的前端开发框架，组件化，数据绑定，2.0引入virtual dom。

ReactNative使用React(4W+ star)。革命性的前端开发框架，组件化，数据绑定，virtual dom。

### Android版本要求：

ReactNative使用了Choreographer，因此必须在API16以上才可以使用。  
weex使用handler来代替Choreographer，可以在API14以上使用。

weex出来的初衷也是为了解决ReactNative使用过程中遇到的一些问题，当然具体决定使用那个框架，我觉得需要从一下几个方面来做对比：

## 学习成本

### 1、环境配置：

ReactNative需要按照文档安装配置很多依赖的工具，相对比较麻烦。  
weex安装cli之后就可以使用。

### 2、vue vs react

react模板JSX学习使用有一定的成本。

```
1. import React, { Component } from 'react';
2. import { AppRegistry, StyleSheet, Text, View } from 'react-native';
3.
4. class LotsOfStyles extends Component {
5.   render() {
6.     return (
7.       <View>
8.         <Text style={styles.red}>just red</Text>
9.         <Text style={styles.bigblue}>just bigblue</Text>
10.        <Text style={[styles.bigblue, styles.red]}>bigblue, then red</Text>
11.        <Text style={[styles.red, styles.bigblue]}>red, then bigblue</Text>
12.      </View>
13.    );
14.  }
15. }
16.
17. const styles = StyleSheet.create({
18.   bigblue: {
19.     color: 'blue',
20.     fontWeight: 'bold',
21.     fontSize: 30,
22.   },
23.   red: {
24.     color: 'red',
25.   },
26. });
27.
28. AppRegistry.registerComponent('LotsOfStyles', () => LotsOfStyles);
```

vue更接近常用的web开发方式，模板就是普通的html，数据绑定使用mustache风格，样式直接

使用css。

```
1. <template>
2.   <wxc-list-item onclick="{{redirect}}">
3.     <text class="item-txt">{{ title }}</text>
4.   </wxc-list-item>
5. </template>
6.
7. <style>
8.   .item-txt {
9.     font-size: 48px;
10.    color: #555;
11.  }
12. </style>
13.
14. <script>
15.   require('weex-components');
16.   module.exports = {
17.     data: {
18.       title: '',
19.       url: ''
20.     },
21.     methods: {
22.       redirect: function() {
23.         this.$openURL(this.url);
24.       }
25.     }
26.   }
27. </script>
```

### 3、布局

两者实现了flexbox的相同子集（都使用了FaceBook的代码解析），基本没有区别。

## 易用性

### 1、sdk使用

ReactNative需要解决mvn依赖的问题，因此必须自己修改源码，打包发布。  
weex可以直接在mvn项目中使用

### 2、调试

都可以在chrome中调试JS代码。

weex支持在chrome中预览页面dom节点，ReactNative不支持。

### 3、页面开发

weex提供了一个playground，可以方便的预览正在开发的页面。（手机预览效果）

ReactNative开发一个页面，需要建立一个native工程，然后编译运行。（需要基于native才能看效果）

### 4、即时预览

weex和ReactNative都有提供hot reload功能，可以边更改代码，边在手机上看到效果。

### 5、打包

ReactNative官方只能将ReactNative基础js库和业务js一起打成一个js bundle，没有提供分包的功能，需要制作分包打包工具。

weex默认打的js bundle只包含业务js代码，体积小很多，基础js库包含在weex sdk中。

### 6、部署

斑马目前同时支持weex和ReactNative页面，但是中心已经转向weex。

另外斑马提供了可以拖拽搭建weex活动页面的功能。

### 7、扩展性

组件的扩展上，weex和ReactNative具有一样的能力。

三方库的接入上，weex对网络，图片，统计等常见的用户可能想自己定制的功能，提供了相应的适配接口，可以由用户方便的定制，ReactNative需要自己修改源码。

### 8、集团库接入

weex有默认的mtop接入实现，UT接入实现。

windvane也提供了对weex页面的支持，可以复用app中的web容器。

## 9、跨平台

ReactNative支持Android iOS两个平台，需要自己扩展去支持web，windows和node-webkit的支持正在开发中。

weex可以支持Android iOS web三个平台。

## 10、Moudle方法调用线程

weex 可以通过注解标注是否在UI线程执行。

ReactNative在native\_modules线程执行。

## 11、异步

weex只支持callback。

ReactNative提供了Promise的支持。

## 性能

### 1、分包加载

ReactNative需要自己实现，从而优化JS加载执行时间。

weex默认提供分包实现。

### 2、官方支持

ReactNative官方关注的重心目前并不在性能上。

weex持续关注性能优化。

### 3、大块view渲染

ReactNative默认没有优化机制，长view渲染性能会比较差。

weex提供了node和tree两种渲染模式，优化长view的渲染。

### 4、ListView Android

ReactNative目前采用scrollView使用，有一些性能问题。  
weex使用recyclerview实现，性能稍好。

## 社区

ReactNative 3w+ star , issue , pull request , contributor多，社区活跃，围绕react产生了许多开发框架。  
weex开源较晚，目前只有4k+ start , contributor以阿里人为主，较少，issue和pull request也比较少，社区目前规模比较小。

## 工具链

### 1、debug tool

都有提供在chrome中调试的支持。

### 2、打包工具

ReactNative需要自己改造。  
weex默认提供的足够满足使用需求。

### 3、webpack , gulp , 脚手架工程

weex有相应的插件，方便开发，部署使用  
ReactNative有，但是很久未更新，需要自己维护

通过上面的一些对比，就我个人来说，我还是比较倾向于使用weex，我比较熟悉vue是一方面，另外性能和发布这一块也是我比较关注的点。使用ReactNative确实也可以做到不错，但是最终我发现，自己其实是在做weex团队已经做的事情。与其这样，为什么我不选择weex，去帮助weex解决一些其他问题，给自己留更多时间去做业务开发呢？另外从业务开发的角度，我也觉得weex的门槛相对比较低，更适合业务开发同学上手，简单就是不简单。

