Weex总结

浅尝hybrid开发



Author：周荣锋、崔庆超

康之元信息技术有限公司

目录

weex 使用同一套代码来构建 Android、iOS 和 Web 应用 1

使用Weex需掌握 1

使用Weex场景建议 1

搭建开发环境 2

创建weex项目 4

开发weex项目 5

weex样式 8

weex环境变量 9

weex跨域问题 9

weex POST请求问题 11

weex vue-router 11

weex 三端开发 15

ios 开发 15

Android 开发 16

源码依赖(IDE Android Studio) 18

js文件打包及加载: 19

Web 开发 19

参考 填坑帮助 19

# weex 使用同一套代码来构建 Android、iOS 和 Web 应用

## 使用Weex需掌握

Weex目标是实现三端互通，使用一套代码，解决现阶段开发过程中需要支持三个平台的基本问题，由于牵扯三端，因此在开发中需要掌握以下语言

Android

IOS

H5/CSS/JavaScript

Vue.js

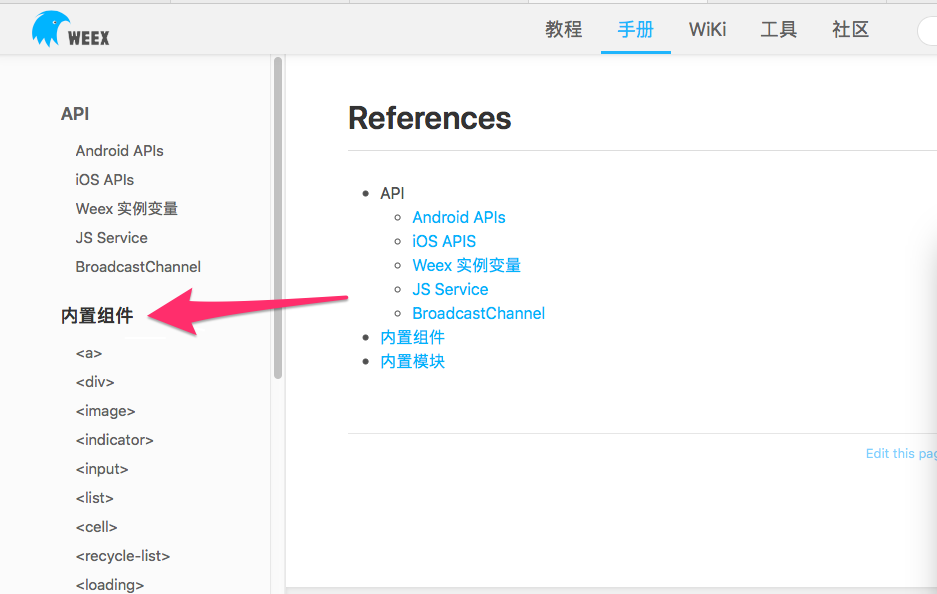
…(其它鬼)

## 使用Weex场景建议

Weex目标虽是三端使用，由于平台差异性以及Weex技术的相对新兴性，某些功能如使用Weex开发，几乎与三端各自开发所花精力一致。

如何快速定位哪个功能是否适合使用Weex开发？

方法很简单，通读Weex手册，如手册中内置组件不能很好实现项目需求或者没有对应组件实现需求，则需要衡量使用Weex的必要性。我们在尝试使用Weex开发的过程中，就遇到了图表、富文本这类功能，经过一些天的头痛后，最终放弃使用Weex实现图表功能，富文本功能也是在移动端自定义相关组件来完成

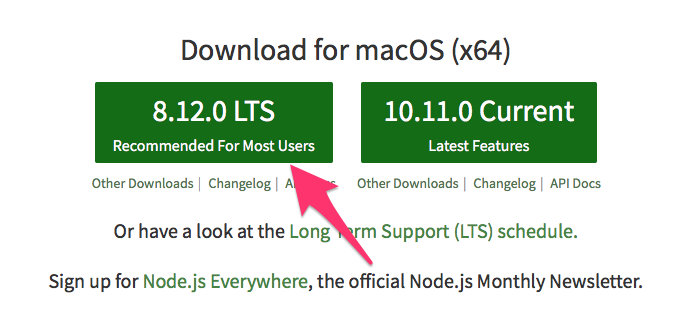


## 搭建开发环境

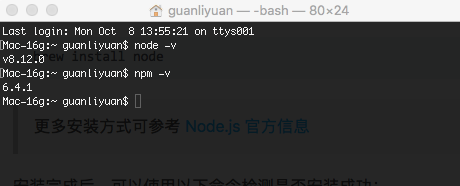
你的电脑需要安装依赖Node.js和weex-toolkit

**node.js**

最快捷方法，直接进入官网 <https://nodejs.org/en/> 请下载推荐版，否则可能出现node太新，与weex开放不兼容问题



在终端，通过 node –v命令查看是否安装成功, 此时，npm包管理工具也随之安装



**接下来，我们使用npm命令来安装weex-toolkit**

npm 是一个 JavaScript 包管理工具，它可以让开发者轻松共享和重用代码。Weex 很多依赖来自社区，同样，Weex 也将很多工具发布到社区方便开发者使用。

安装前，确定npm版本大于5，否则可以通过npm i npm@latest -g更新一下npm的版本

npm install –g weex-toolkit

安装完后，通过weex –v查看当前weex版本

weex-toolkit升级子依赖方法(可忽略此步，如按步骤来一切正常)

weex update weex-devtool@latest //@后标注版本后，latest表示最新

**使用淘宝镜像安装weex-toolkit**

|  |
| --- |
| $ npm install -g cnpm --registry=https://registry.npm.taobao.org $ cnpm install -g weex-toolkit |

*提示：*

如果提示权限错误（*permission error*），使用 sudo 关键字进行安装

|  |
| --- |
| $ sudo cnpm install -g weex-toolkit |

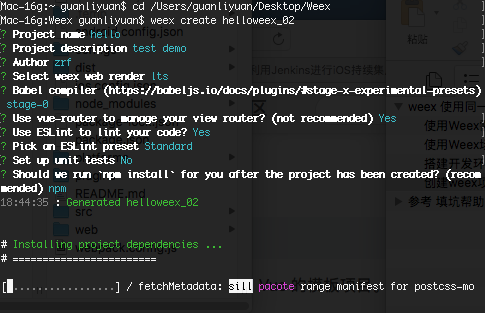
基本依赖已经安装完毕，接下来，开始一步步深入…

不要哭、

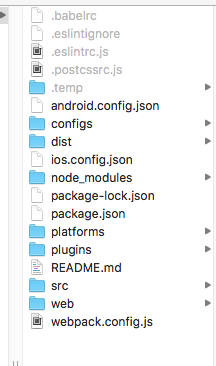
## 创建weex项目

先cd到目标文件夹下

$ weex create helloweex ,创建过程中，会要求填写一堆信息（如果你创建没出现，不要得意，可能你这个环境安装就有问题，webpack版本过低?）



此时在目标文件夹下生成了一个helloweex文件夹



## 开发weex项目

先、我们站在三端的高度看世界，可能会迷茫，到底应该在哪里开发？

我们就很开心的遇到了这样的情况，环境什么都配置好了，硬是不知道需要在哪里开，然后就对着Weex项目混乱地研究了几天。

请看上页项目结构截屏，对于初级菜鸟的我们，只需要关注src文件夹、platforms文件夹可能根本不需要（如果你已经有了对应的SDK代码），dist文件夹不需要我们去特别关注，因为它是通过src而来的，三端中使用的就是dist文件夹下相关js文件。我们是使用sublime来进行weex项目编写

**如果需要关注platforms文件夹状态**

默认情况下 weex create 命令并不初始化 iOS 和 Android 项目，你可以通过执行 weex platform add 来添加特定平台的项目。

|  |
| --- |
| weex platform add ios //生成ios项目 weex platform add android |

由于网络环境的不同，安装过程可能需要一些时间，请耐心等待。如果安装失败，请确保自己的网络环境畅通。

不建议使用

weex run ios  
weex run android  
weex run web

这几个命令在模拟器上启动，因为体验下来真的很差，你如果是这几端的开发人员，相信有你已经习惯的方式去启动的

现在是不是大概清楚点？对、编写代码的时候，只要在src文件夹下去编写。

在 package.json 中，已经配置好了几个常用的 npm script，分别是：

* build: 源码打包，生成 JS Bundle
* dev: webpack watch 模式，方便开发
* serve: 开启HotReload服务器，代码改动的将会实时同步到网页中

我们先通过 npm install 安装项目依赖。之后运行项目根目录下的 npm run dev & npm run serve开启 watch 模式和静态服务器。也可以尝试使用npm start启动

当浏览器自动打开weex h5页面后，每次修改了代码，只要点击保存或快捷键，界面会自动更新，我们可以马上看到更新后结果

代码如下所示：

|  |
| --- |
| <template>  <div class="wrapper" @click="update">  <image :src="logoUrl" class="logo"></image>  <text class="title">Hello {{target}}</text>  </div> </template>  <style>  .wrapper { align-items: center; margin-top: 120px; }  .title { padding-top:40px; padding-bottom: 40px; font-size: 48px; }  .logo { width: 360px; height: 156px; }  </style>  <script>  export default {  data: {  logoUrl: 'http://img1.vued.vanthink.cn/vued08aa73a9ab65dcbd360ec54659ada97c.png',  target: 'World'  },  methods: {  update: function (e) {  this.target = 'Weex'  console.log('target:', this.target)  }  }  } </script> |

**按页面生成JS**

weex compile src dist

**读取本地资源问题**

Weex SDK 提供 local scheme 来访问打包在应用程序中的资源，此 scheme 无法在 H5 环境下使用。目前，开发者可以在 image 组件和字体文件中使用本地资源。

* 在 iOS 中，Weex 会在 bundle resources 中查找。例如，image 组件的 src 属性为 local:///app\_icon'， Weex 会加载 bundle resouce 中名为 app\_icon 的图像资源，而字体文件也以相同的方式工作。
* 在 Android 中，image 组件将从 drawable 资源文件夹加载，如  res/drawable-xxx。但加载字体文件是不同的，Android 框架无法从 res 加载字体文件，因此 SDK 将从 asserts 文件夹加载它。

跟着官网这句话去尝试，在iOS端，你会发现并没有显示出本地图片（起码我验证是这么一个结果），最终发现可能是个平台的差异性，在iOS上需要加上图片后缀，如 app\_icon.png。但是iOS内部图片都是由@2x或者@3x后缀，需要实现本地读取图片，估计还要耗费一些精力封装兼容方法。。。

## weex样式

Weex 对于长度值目前只支持*像素*值，不支持相对单位（em、rem等）

<template>  
   
</template>  
结构中只能包括一个块元素

每个样式只支持单一写法，不支持组合写法，如：border: 1 solid #ff0000;

你可以通过 background-image 属性创建线性渐变。

|  |
| --- |
| background-image: linear-gradient(to top,#a80077,#66ff00); |

目前暂不支持 radial-gradient（径向渐变）。

Weex 目前只支持两种颜色的渐变，渐变方向如下：

* to right  
  从左向右渐变
* to left  
  从右向左渐变
* to bottom  
  从上到下渐变
* to top  
  从下到上渐变
* to bottom right  
  从左上角到右下角
* to top left  
  从右下角到左上角

表示百分比值，如“50％”，“66.7％”等。

它是 CSS 标准的一部分，但 Weex 暂不支持。Weex默认屏幕宽度为750px,根据输入的像素，自动scale到对应的大小。

假设您需要显示固定为 88 px 的导航栏，该导航栏的高度将是：

|  |
| --- |
| var height = 88 \* 750 / env.deviceWidth |

Weex 在 0.13 版本 SDK 里实现了事件冒泡机制。注意事件冒泡默认是不开启的，你需要在模板根元素上加上属性 bubble=true 才能开启冒泡。

{  
 handleClick (event) {  
 // 阻止继续冒泡.  
 event.stopPropagation()  
 }  
}

## weex环境变量

Weex 提供了 weex.config.env 和全局的 WXEnvironment 变量（它们是等价的）来获取当前执行环境的信息。

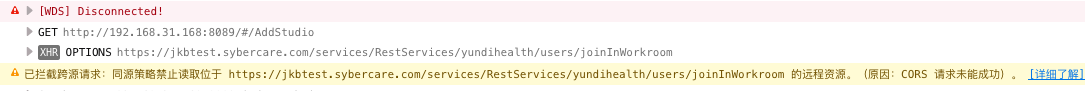
|  |
| --- |
| weex.config.env === WXEnvironment |

**Weex 环境变量中的字段**:

| **字段名** | **类型** | **描述** |
| --- | --- | --- |
| platform | String | Current running platform, could be “Android”, “iOS” or “Web”. |
| weexVersion | String | The version of Weex SDK. |
| appName | String | Mobile app name or browser name. |
| appVersion | String | The version of current app. |
| osName | String | The OS name, could be “Android” or “iOS”. |
| osVersion | String | The version of current OS. |
| deviceModel | String | Mobile phone device model. (native only) |
| deviceWidth | Number | Screen resolution width. |
| deviceHeight | Number | Screen resolution height. |

## weex跨域问题

使用浏览器调试时，如果界面包含网络请求，会遇到



接受不到网络请求结果，使用下面方法能解决此问题

打开weex项目，并且找到configs文件夹下，打开config.js,找到‘proxyTable’,在里面添加如下代码



target：你需要请求的服务器地址，替换成对应地址即可.增加完后，如果无效，建议重启服务试试

跨域处理前网络请求写法



跨域处理后写法，主要观察url



## weex POST请求问题

如上页截屏为post请求正确写法，在上面写法中，需要注意两点

**传参，需要将object对象转换为string对象**

../../Library/Containers/com.tencent.qq/Data/Library/Application%20Support/QQ/Users/229072123/QQ/Temp.db/29DF5CF1-58B5-4276-AEEC-3CFCF0247322.png

**header, post需要配置content-type为’application/json‘**

**如未按上述方法处理，将出现请求415错误**

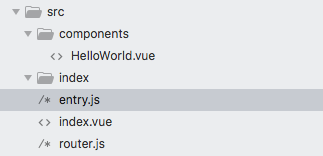
HTTP请求415错误 – 不支持的媒体类型(Unsupported media type)

## weex vue-router

Weex版本



Src路径结构



entry.js内容

某些全局方法与全局类可以定义在此处, 如: import mixins from ‘./mixins’,在此js文件中可以处理公共方法：网络请求、时间格式等. 在其他Vue页面中可以直接调用



**index.vue简介**

index.vue为项目主入口，不可单独当一个页面处理，类似IOS开发中的AppDelegate。如下网易严选weex项目index.vue截屏。Router-view用于控制多页面问题，tab-bar为主架构结构，因此放在Index.vue中



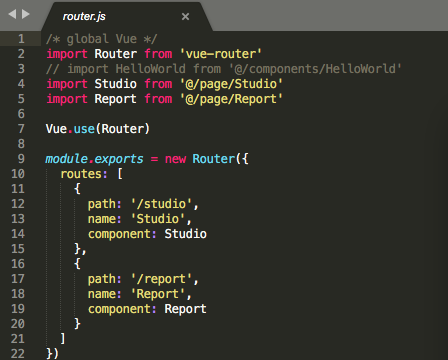
**Router-view多页面配置**

所有的页面在router.js中进行配置,初次显示页面可以在router中使用redirect来重定向entry.js中最后一句router.push(‘/’)



也可以直接修改entry.js中router.push方法

如：router.push('/studio') //指定一个路由入口，初始化显示的页面内容



# weex 三端开发

## ios 开发

依赖第三方框架，请确保工程中包括SDWebImage/WeexSdk框架

**必不可少的WXImgLoaderDefaultImpl文件**

除非你能立字据，说你整个项目都不会有网络图片下载，否则在工程里少不了它的影子

**Weex与IOS设备尺寸差异性**

由于Weex长宽是一个固定值，为了适配不同机型，会产生一个属性用于记录像素缩放比

/\*\*

 \*  scale factor from css unit to device pixel.

 \*/

@property (nonatomic, assign, readonly) CGFloat pixelScaleFactor;

如下两种情形传输高度情况

IOS计算的长宽传输到Weex展示

 WXPixelType pix = \_label.frame.size.height;

        if ([WXInteractiveDataManager shareManager].fPixScale > 0.001) {

            pix = pix / [WXInteractiveDataManager shareManager].fPixScale / 2;  //将坐标转化为weex坐标  由于weex中富文本标签做了乘以2的y处理，此次除以2，使真实结果不变

        }

fPixScale为 pixelScaleFactor 化身

Weex长宽传输到IOS展示使用WXConvert

 pix = [WXConvert WXPixelType:weexHeight scaleFactor:[WXInteractiveDataManager shareManager].fPixScale];

## Android 开发

#### ****Android 集成有两种方式****

1. 源码依赖：能够快速使用WEEX最新功能，可以根据自己项目的特性进行相关改进。
2. SDK依赖：WEEX 会在jcenter 定期发布稳定版本。[jcenter](https://bintray.com/alibabaweex/maven/weex_sdk/view)  
   注:国内可能需要翻墙

#### 前期准备

* 已经安装了[JDK](http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html) version>=1.7 并配置了环境变量
* 已经安装[Android SDK](https://developer.android.com/studio/index.html) 并配置环境变量。
* Android SDK version 23 (compileSdkVersion in [build.gradle](https://github.com/apache/incubator-weex/blob/master/android/sdk/build.gradle))
* SDK build tools version 23.0.1 (buildToolsVersion in [build.gradle](https://github.com/apache/incubator-weex/blob/master/android/sdk/build.gradle))
* Android Support Repository >= 17 (for Android Support Library)

#### 快速接入

如果你是尝鲜或者对稳定性要求比较高可以使用依赖SDK的方式。  
步骤如下：

1. 创建Android工程，没有什么要特别说明的，按照你的习惯来。
2. 修改build.gradle 加入如下基础依赖

|  |
| --- |
| compile 'com.android.support:recyclerview-v7:23.1.1' compile 'com.android.support:support-v4:23.1.1' compile 'com.android.support:appcompat-v7:23.1.1' compile 'com.alibaba:fastjson:1.1.46.android' compile 'com.taobao.android:weex\_sdk:0.18.0' |

1. 注:版本可以高不可以低。

#### 代码实现

注:附录中有完整代码地址

* 实现图片下载接口，初始化时设置。

|  |
| --- |
| /\*\*  \* Created by lixinke on 16/6/1.  \*/ public class ImageAdapter implements IWXImgLoaderAdapter {    @Override  public void setImage(String url, ImageView view, WXImageQuality quality, WXImageStrategy strategy) {  //实现你自己的图片下载，否则图片无法显示。  } } |

* 初始化

|  |
| --- |
| /\*\*  \* 注意要在Manifest中设置android:name=".WXApplication"  \* 要实现ImageAdapter 否则图片不能下载  \* gradle 中一定要添加一些依赖，否则初始化会失败。  \* compile 'com.android.support:recyclerview-v7:23.1.1'  \* compile 'com.android.support:support-v4:23.1.1'  \* compile 'com.android.support:appcompat-v7:23.1.1'  \* compile 'com.alibaba:fastjson:1.1.45'  \*/ public class WXApplication extends Application {   @Override  public void onCreate() {  super.onCreate();  InitConfig config=new InitConfig.Builder().setImgAdapter(new ImageAdapter()).build();  WXSDKEngine.initialize(this,config);  try {  WXSDKEngine.registerModule("poneInfo", PhoneInfoModule.class);//注册Module  WXSDKEngine.registerModule("netInfo", WeexNetModule.class);  WXSDKEngine.registerComponent("rich", RichText.class, false);//注册Component  } catch (WXException e) {  e.printStackTrace();  }  } } |
|  |

* 开始渲染

|  |
| --- |
| public class MainActivity extends AppCompatActivity implements IWXRenderListener {   WXSDKInstance mWXSDKInstance;   @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.activity\_main);   mWXSDKInstance = new WXSDKInstance(this);  mWXSDKInstance.registerRenderListener(this);  /\*\*  \* WXSample 可以替换成自定义的字符串，针对埋点有效。  \* template 是.we transform 后的 js文件。  \* option 可以为空，或者通过option传入 js需要的参数。例如bundle js的地址等。  \* jsonInitData 可以为空。  \* width 为-1 默认全屏，可以自己定制。  \* height =-1 默认全屏，可以自己定制。  \*/  mWXSDKInstance.render("WXSample", WXFileUtils.loadFileContent("hello.js", this), null, null, -1, -1, WXRenderStrategy.APPEND\_ASYNC);  } } |

### 源码依赖(IDE Android Studio)

这种方式不利于weex sdk的更新维护 不推荐使用。

## js文件打包及加载:

将编写好的index.vue文件编译成index.is:

weex compile index.vue dist

然后将js文件拷贝的android/ios项目 android的放在assets目录下，使用WXFileUtils.loadAsset(“index.js”, this)获取文件路径

## Web 开发

Weex build web/npm run build:prod:web/npm run pack:web 生成release文件夹，里面html文件为web端使用

# 参考 填坑帮助

## 设置样式后，web端显示，移动端无效

<div class="add-studio-title">

<text class="title">加入工作室</text>

</div>

.title {

font-size: 35px;

text-align: center;

padding-top: 12px;

padding-bottom: 15px;

border-bottom-width: 2px;

border-bottom-style: solid;

border-bottom-color: #ccc;

}

效果大概为居中显示，上面为正确写法

造成不显示原因为：修改前加上了父组件样式前缀，移动端无法识别

原写法: .add-studio-title .title{…}

weex官网： <http://weex.apache.org/cn/guide/index.html>

weex IDE: <https://www.jianshu.com/p/f2fca2719f23>

weex例子: <https://weex.apache.org/cn/examples.html>

weex web打包启示: <https://www.cnblogs.com/crazycode2/p/7989932.html>

weex UI组件: <https://alibaba.github.io/weex-ui/#/?id=weex-ui>

weex vue-router: <https://www.jianshu.com/p/2a2ae8a4118f>

vue mixin: <https://blog.csdn.net/china_bomber_20/article/details/79034365>

请求415: <http://www.cnblogs.com/cocoajin/p/3986204.html>