[Wap项目前端文档 4](#_Toc14518)

[一、项目简介 4](#_Toc24300)

[1、Wap项目主要使用语言是javascript、html、css。 4](#_Toc31273)

[2、Wap项目分为4个端，但是都用同一套代码。各个端的区别： 4](#_Toc23751)

[二、项目搭建环境 4](#_Toc15905)

[1、使用gulp打包工具：（gulp 打包工具会把wap/src 目录下的HTML、Less、js、images图片等文件进行压缩，在压缩的同时，会为wap/src目录下的html文件内引用资源（css，js）添加版本号，并把wap/src/css目录下的less文件编译成css文件） 4](#_Toc19429)

[2、安装git工具 ：去官网 https://git-scm.com/ 下载git并安装 4](#_Toc25811)

[三、获取项目代码资源 5](#_Toc3149)

[1、git地址： http://172.16.128.252:8001/flicat/wap.git 5](#_Toc21571)

[2、把wap项目的代码克隆下来 5](#_Toc16800)

[四、wap/src目录下的文件介绍 6](#_Toc26747)

[五、Html文件介绍 7](#_Toc11015)

[六、Css文件 ，css文件使用less编写，再由gulp编译成css文件 9](#_Toc29238)

[七、css公共类名和组件介绍 10](#_Toc5029)

[1、layout.less: 公共类名样式 10](#_Toc8219)

[2、widget.less: popup.js 弹窗插件样式 11](#_Toc26464)

[3、animation.less： 动画效果类名样式 12](#_Toc20171)

[八、Js文件 12](#_Toc5959)

[1、config.js 全局配置文件，每次压缩代码是并为js缓存版本号，最重要的是可以获取设备/平台的信息 12](#_Toc24134)

[2、app-bridge.js： app桥接js ，通过里面的一些方法来连接wap站和APP 14](#_Toc4580)

[3、app-url.js ： 百洋app链接与wap链接互相转换 17](#_Toc24426)

[4、 widget.js 定义公共的方法，以下方法可以直接调用： 17](#_Toc30506)

[4.1、头部公共弹出菜单 data-act="menu"data-param="id=top-nav" 。 17](#_Toc17062)

[4.2、日历控件 data-act="calendar " data-param="range=1" 17](#_Toc11500)

[4.3、搜索弹窗：data-act="search" data-param="is\_global=xx& keyword=xx" 18](#_Toc14617)

[4.4、小能客服的使用 18](#_Toc24579)

[4.5、隐藏/显示某个class 19](#_Toc17320)

[data-act="toggle" data-param="class=xx&event=xxxx" 19](#_Toc8062)

[4.6、全选复选 20](#_Toc2010)

[data-act="select" data-param="class=xx&type=xx" 20](#_Toc27295)

[4.7、tab选项卡 data-act="tab" 20](#_Toc118)

[4.8、轮播插件 ：data-act="swipe" data-param="auto=1" 21](#_Toc3951)

[4.9、图片延时加载：data-act="lazy" data-param="url" 21](#_Toc4154)

[4.10、页脚菜单 ：data-act="footer" 22](#_Toc26407)

[5、ajax.js ：ajax请求插件；return [Function] 22](#_Toc28454)

[6、event.js: 触摸事件插件；return null 23](#_Toc9930)

[7、exbind.js：HTML控件定义插件；return exbind[Object] 23](#_Toc28455)

[8、mvvm.js：手机版MVVM插件；return MVVM[Object] 24](#_Toc28066)

[9、Qrcode.js: 生成二维码插件 return QRCode[Object]; 24](#_Toc15152)

[10、popup.js：弹窗插件；return popup[Object] 25](#_Toc18550)

[11、share.js：分享插件；return Share[Object] 25](#_Toc3074)

[12、public.js: 以下是已经定义好的方法，可以直接使用 26](#_Toc22459)

[13、router.js: 单页应用页面路由器 27](#_Toc26662)

[13.1、register(url, callback[, is\_default, is\_route]) js定义页面动作 27](#_Toc30738)

[13.2、onChange(handler)： 添加页面切换回调事 28](#_Toc28917)

[13.3、route(url, ignore) :手动路由到指定页面 28](#_Toc1988)

[13.4、registerNode(node, url)：手动注册DOM节点页面 28](#_Toc3231)

[13.5、config(option)：配置页面切换样式 28](#_Toc31012)

[14、sea.js：seajs插件 网上介绍文档：http://seajs.org/docs/ 29](#_Toc32135)

[15、jweixin.js：微信jssdk 29](#_Toc8444)

[16、log.js：手机版console工具；return log[Function] 29](#_Toc14709)

[17、 validate.js：表单验证插件 29](#_Toc26995)

[18、weixin.js：微信SDK签名 return wxsdk[Object] 30](#_Toc10093)

[九、image文件 30](#_Toc5389)

# Wap项目前端文档

## 一、项目简介

### 1、Wap项目主要使用语言是javascript、html、css。

### 2、Wap项目分为4个端，但是都用同一套代码。各个端的区别：

2.1、微信内：特有微信支付；没有头部导航栏；右下角有菜单按钮。

2.2、触屏机：微信扫码登录；微信扫码支付；支付宝扫码支付。

2.3、百洋APP内：

2.4、浏览器：wap 全站

## 二、项目搭建环境

### 1、使用gulp打包工具：（gulp 打包工具会把wap/src 目录下的HTML、Less、js、images图片等文件进行压缩，在压缩的同时，会为wap/src目录下的html文件内引用资源（css，js）添加版本号，并把wap/src/css目录下的less文件编译成css文件）

1.1、打包工具依赖 nodejs，首先安装node.js

1.2、接着执行 npm install –g gulp  安装gulp

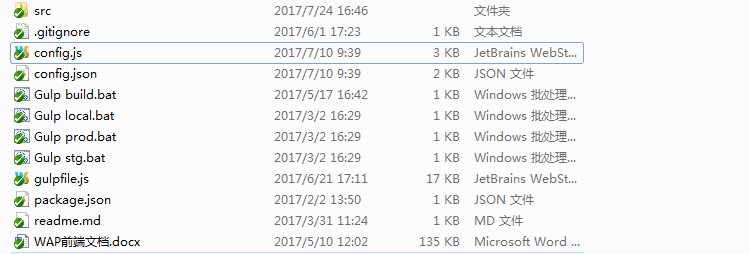
1.3、再通过敲命令行cd进入wap目录下运行 npm install 。

### 2、安装git工具 ：去官网 <https://git-scm.com/> 下载git并安装

## 三、获取项目代码资源

### 1、git地址： <http://172.16.128.252:8001/flicat/wap.git>

### 2、把wap项目的代码克隆下来



其中文件Gulp build.bat 、Gulp local.bat、Gulp prod.bat、Gulp stg.bat 这个四个文件都是压缩工具，点击运行会把wap/src下的文件进行压缩并把压缩代码保存在文件夹里：

Gulp build.bat 压缩所有环境的代码，根据文件config.json 里面有多少个key值就压缩出多少份以key值命名的文件夹。（这个能压缩出4个文件夹的代码，包括dist\_local、dist\_prod、dist\_stg、dist\_stg\_test（统一后台stg环境的压缩代码） ）

Gulp local.bat 压缩58环境的代码，压缩后的所有文件都放在dist\_local这个文件夹里面。

Gulp prod.bat 压缩正式环境的代码（线上代码），压缩后的所有文件都放在dist\_prod这个文件夹里面。

Gulp stg.bat 压缩stgm环境的代码，压缩后的所有文件都放在dist\_stg这个文件夹里面。（这个环境的代码不包括对接统一后的代码）

config.json 文件配置 （以prod 为例）

"prod": {  
 "debug": 0,  
 "platform": "prod",  
 "ajax\_url": "https://mservice.baiyjk.com/",  
 "auth\_url": "https://open.weixin.qq.com/connect/oauth2/authorize?appid=wx62eaed8770f89f93&redirect\_uri=",  
 "cdn\_url": "https://wapstatic.baiyangwang.com/wap/",  
 "login\_appid": "wxea88f5e6b8956165",  
 "skin\_file": "local.less"  
}

debug : 是否允许调试 0：不允许 1：允许

Platform : 环境信息。 prod：生产环境 ， stg：stg测试环境 local：58 测试环境

ajax\_url：ajax请求地址

Config.url = 'http://172.16.40.250:8011/'; // 58局域网ip

Config.url = 'http://58.63.114.90:9011/'; // 58测试环境

Config.url = 'https://mstest.baiyangwang.com/'; // stg测试环境

Config.url = 'http://mservice.baiyjk.com/'; // 正式环境

auth\_url：微信授权地址

cdn\_url： 静态资源地址

login\_appid：微信授权appid



## 四、wap/src目录下的文件介绍

Css : 样式表文件，以前是每个功能模块就一个less样式表，现在改为每个html页面就对应一个less样式表

Js：脚本文件

Js/module：功能业务逻辑脚本

Js/plugin：js公共插件

Images：样式图片文件，页面用到的静态图片

Theme : wap站皮肤主题

Template ：公共模板

Skin : 公共样式 和 网站主色调样式（字体颜色，按钮颜色，同颜色的icon）

## 五、Html文件介绍

1、 文件命名：按功能模块划分。例如会员中心页面以 user- 前缀开头，订单页面以order-前缀开头。

2、基本页面布局，现在的html页面基本都按照这个标准来写。（注意：特殊符号一律使用 HTML转义字符 <http://www.w3chtml.com/html/character.html> ）

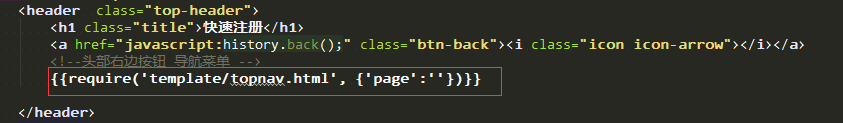
<!DOCTYPE html>  
<**html** manifest=**"manifest.appcache"** lang=**"zh-cmn-Hans"**>  
<**head**>  
 **{{require('template/head.html')}}** <**meta** name=**"applicable-device"** content=**"mobile"**>  
 <**meta** name=**"Keywords"** content=**"网上药店,药房网,药店网,网上药房,网上买药, 药房网商城,百洋商城网上药店"** />  
 <**meta** name=**"description"** content=**"百洋商城，妈妈的网上药店，药监局认证的合法网上药店，100%正品保证，由执业药师在线服务，为您提供全面的营养、保健、药用指导和专业、优质、便捷的网上买药服务。原青岛百洋健康药房网商城官网，买安全放心药就上百洋药房网"** />  
 <**link** rel=**"canonical"** href=**"https://www.baiyangwang.com"**>  
 <**title**>**百洋商城-妈妈的网上药店，青岛正规专业药房 送药上门**</**title**>  
 <**link** rel=**"stylesheet"** href=**"css/index.css"**/>  
</**head**>  
<**body** class=**"product-list"** >

<**div** class=**"wrap box"** ></**div**>  
</**body**>  
</**html**>

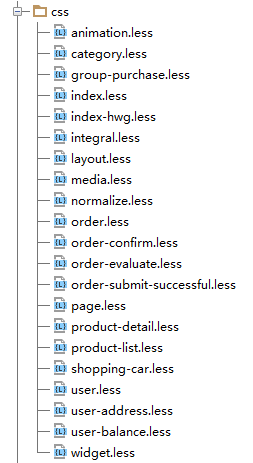
3、html页面会根据需要，可以引入三个公共模板：footer.html 、head.html 和 topnav.html

<!--底部页脚导航-->  
**{{require('template/footer.html', {'page':'index'})}}**

**{{require('template/head.html')}}**



## 六、Css文件 ，css文件使用less编写，再由gulp编译成css文件



1、以下几个为公共样式：

animation.less：动画效果css

layout.less：全局公共样式

normalize.less：样式重定义

widget.less：js控件样式

media.less：媒体查询css（未使用）

2、文件命名：按照栏目功能来命名，例如会员中心使用user-前缀

3、less使用的字体大小：

如果psd图的宽度是640，则用 @rem640，如果psd图的宽度是750，则用 @rem750

rem640: (640 / 320 \* 12**rem**);

@rem750: (750 / 320 \* 12**rem**);

## 七、css公共类名和组件介绍

### 1、layout.less: 公共类名样式

hide // 强行隐藏

.clear // 清除浮动

.text-hide // 隐藏文字（将文字推到外面）

.pull-left // 左浮动

.pull-right // 右浮动

.toggle-box // 仿爱疯开关按钮

.form-control // 表单控件

.form-control > .toggle-box // 仿爱疯开关按钮

.form-control > .toggle-psw // 密码隐藏/可见按钮

.form-control.error // 表单验证错误样式

.form-control > .input-static // 文字仿表单控件

.form-control .verify-img // 验证码图片

.form-control.error > .error-msg // 表单验证错误信息

.btn // 按钮样式

.btn.btn-sm // 小按钮

.btn.btn-default // 粉红色按钮

.btn.btn-disabled // 灰色按钮

/\* 栅格布局 \*/

.row // 栅格布局父层（宽度100%）

.row > .col-1 (col-1至col-12 将栏目划分成12分) // 栅格

.row > .col-offset-1 (col-offset-1至col-offset-12 ) // 栅格偏移

/\* box自适应内滚动布局 \*/

.box // 盒模型父层（宽高为100%）

.box > .flex // 自适应高度层

.box > .flex > .flex-content // 内滚动区域

### 2、widget.less: popup.js 弹窗插件样式

// 弹窗

.popup //弹窗父层

.popup > .popup-content //弹窗内容区域

.popup > .popup-content > .popup-success //成功信息弹窗

.popup > .popup-content > .popup-fail //失败信息弹窗

.popup > .popup-content > .popup-error //错误信息弹窗

.popup > .popup-content > .popup-loading //loading弹窗

.popup > .popup-content > .popup-share //分享弹窗

.popup > .popup-content > .popup-confirm //确认信息弹窗

.popup > .popup-content > .popup-goods //商品详情页 分享弹窗

### 3、animation.less： 动画效果类名样式

shake1() //闪动

.shake2() //闪动

.tada() //跳动

.bounceIn() //闪入

.bounceInDown() //由下向上闪入

.bounceInLeft() //由左向右闪入

.bounceInRight() //由右向左闪入

.bounceInUp() //由上向下闪入

.fadein() //淡入

.fadeInDown() //由下向上淡入

.fadeInLeft() //由左向右淡入

.fadeInRight() //由右向左淡入

.fadeInUp()  //由上向下淡入

## 八、Js文件

### 1、config.js 全局配置文件，每次压缩代码是并为js缓存版本号，最重要的是可以获取设备/平台的信息

/ 设备/平台信息  
*config*.platform = (function() {  
 var userAgent = *win*.navigator.userAgent;  
 var platform = *win*.navigator.platform;  
 var testResult = {  
 isFromMobile: /AppleWebKit.\*Mobile.\*/.test(userAgent) || /i(Phone|P(o|a)d)/gi.test(platform), // 移动端  
 isFromAndroid: /android/gi.test(userAgent), // 安卓  
 isFromIos: /iphone|ipod|ipad|ios/gi.test(userAgent), // 苹果设备  
 isFromWx: /MicroMessenger/gi.test(userAgent), // 微信  
 isFromQQ: /mobile.\*qq/gi.test(userAgent), // QQ  
 isFromUC: /ucbrowser/gi.test(userAgent), // UC  
 isFromQQBrower: /mqqbrowser[^LightApp]/gi.test(userAgent),

// QQ浏览器  
 isFromQQBrowerLight: /MQQBrowserLightApp/gi.test(userAgent),

// QQ内置浏览器  
 isFromBaiYangApp: /\bbaiyang\b/gi.test(userAgent), // 百洋APP  
 byAppVersion: (userAgent.match(/\bBaiYang V([\d\.]+)\b/i)||[])[1] || '', // 百洋APP版本号  
 isFromYXYApp: /CYTYXY/gi.test(userAgent), // 育学园APP  
 isFromBYTouchMachine: /\bBaiYangTouchMachine\b/gi.test(userAgent), // 触屏机  
 isFromWochacha: /\bWochacha\b/gi.test(userAgent), // 我查查app  
 isFromPc: /\bWindows|Macintosh\b/gi.test(userAgent), // 判断是否在PC端  
 DeviceId: (userAgent.match(/\bDeviceId\/(\w+)\b/i)||[])[1] || '' // 触屏机设备号  
 };  
 var className = [], classList = html.classList;  
 testResult.isFromMobile && className.push('mobile');  
 testResult.isFromAndroid && className.push('android');  
 testResult.isFromIos && className.push('ios');  
 testResult.isFromWx && className.push('weixin');  
 testResult.isFromBaiYangApp && className.push('baiyang-app');  
 testResult.isFromBYTouchMachine && className.push('by-touch-machine');  
 testResult.isFromWochacha && className.push('wochacha');  
 testResult.isFromPc && className.push('pc-view');  
 testResult.isFromYXYApp && className.push('yxy-app');  
 className.length && classList.add.apply(classList, className);  
 return testResult;  
})();

### 2、app-bridge.js： app桥接js ，通过里面的一些方法来连接wap站和APP

// 调用app登录事件

*require*.async('app-bridge', function(*bridge*) {  
 *bridge*.loginApp(function(*token*) {},{});  
});

// app内等待app登录事件, 获取APP的token

*require*.async('app-bridge', function(*bridge*) {  
 var timer = setTimeout(function() {  
 that.checkLogin();  
 }, 2000);  
 *bridge*.getToken(function(*token*) {  
 clearTimeout(timer);  
 that.setToken(*token*);  
 that.checkLogin();  
 });  
});

//隐藏app分享按钮 true 显示， false 隐藏  
*require*.async('app-bridge', function(*bridge*) {  
 *bridge*.showShareButton(function() {},{"show":false});  
});

//跳转到app首页

*require*.async('app-bridge', function(*bridge*) {  
 *bridge*.jumpToAppHomePage(function() {  
 });  
});

// 获取app udid 主要用于抢红包

*require*.async('app-bridge', function(*bridge*) {  
 var timer = setTimeout(checkBonusEnable, 2000);  
 *bridge*.getUdid(function(*data*) {  
 clearTimeout(timer);  
 is\_bonus\_enable = true;  
 if(typeof *data* === 'string'){  
 try{  
 AppUdiDSignID = JSON.parse(*data*);  
 } catch (*e*) {  
 AppUdiDSignID = null;  
 }  
 } else {  
 AppUdiDSignID = *data*;  
 }  
 checkBonusEnable();  
 });  
});

//调用app的分享

*require*.async('app-bridge', function(*bridge*) {  
 *bridge*.shareInApp(function() {  
 js调用该方法成功后，js想做的操作  
 },{"share\_url":"分享点击的url","share\_icon":"分享的图片 icon","share\_title":"分享标题","share\_detail":"分享副标题"});

});

### 3、app-url.js ： 百洋app链接与wap链接互相转换

appUrl.app\_to\_wap(*words*)； //把app链接转换成wap链接

appUrl.wap\_to\_app(*words*)； //把wap链接换成app链接转

### 4、 widget.js 定义公共的方法，以下方法可以直接调用：

#### 4.1、头部公共弹出菜单 data-act="menu"data-param="id=top-nav" 。

主要用在template下的topnav.html

<**a** href=**"javascript:;"** class=**"btn-menu"** data-act=**"menu"** data-param=**"**<%= 'id=' + *id* %>**"**><**i** class=**"icon icon-menu"**></**i**></**a**>

#### 4.2、日历控件 data-act="calendar " data-param="range=1"

/\* 日历控件  
\* param： range 日历选择范围 1：只能选择今天以前的日期 2：只能选择今天以后的日期  
\*/

使用方法：

<**input** type=**"text"** class=**"calendar-input"** data-act=**"calendar"** data-param=**"range=2"** vm-value=**""**/>

#### 4.3、搜索弹窗：data-act="search" data-param="is\_global=xx& keyword=xx"

/\* 搜索弹窗  
 \* is\_global: 是否是海外购  
 \* keyword: 默认关键词(如果没有传递则从url获取)  
 \*/

使用例子：

<**a** class=**"col-3"** href=**"javascript:;"** data-act=**"search"**>

<**i** class=**"icon icon-search"**></**i**>**搜索**

</**a**>

#### 4.4、小能客服的使用

普通商品客服：data-act="ntkf" data-param="type=0&itemid=商品id&itemparam=4"

药师咨询：data-act="ntkf" data-param="type=2 &itemid=商品id&itemparam=4"

/\* 小能客服  
 \* type: // 客服类型 0：售前客服 1：售后客服 2：药师咨询  
 \* itemid: // 商品id  
 \* itemparam: // 商品接口扩展字段  
 \* orderid: // 订单ID  
 \* orderprice: // 订单总价  
 \*/

使用例子：

<a class="popup-callcenter-btn onlinecall" data-act="ntkf" data-param="type=0&itemid=8011&itemparam=4"></a> 售前服务

<a class="popup-callcenter-btn onlinecall" data-act="ntkf" data-param="type=1&itemid=8011&itemparam=4"></a> 售后服务

<a class="popup-callcenter-btn onlinecall" data-act="ntkf" data-param="type=2&itemid=8011&itemparam=4"></a> 药师咨询

#### 4.5、隐藏/显示某个class

#### data-act="toggle" data-param="class=xx&event=xxxx"

/\*toggle class  
 \* class // class名称  
 \* event // 事件类型 默认为click点击事件  
 \*/

使用例子：

<a data-act="toggle" data-param="class=ishow"></a>

#### 4.6、全选复选

#### data-act="select" data-param="class=xx&type=xx"

\*全选复选框  
 \* 控件名称：data-act="select"  
 \* 控件参数：data-param="class=@class&type=@type"  
 \* @class：需要全选的复选框按钮组的class名称  
 \* @type：全选类型 1（只有全部按钮选中，全选按钮才会勾上）|2（只要有一个按钮选中，全选按钮就会勾上）  
 \*/

使用例子：

<**input** type=**"checkbox"** name=**"type"** class=**"js-type"** data-act=**"select"** data-param=**"class=js-shop"** />

#### 4.7、tab选项卡 data-act="tab"

\* tab选项卡  
\* tab标题类名 js-tab-title  
\* tab内容类名 js-tab-content  
\*/

使用例子：

<**div** class=**"tab-wrap"** data-act=**"tab"**>  
 <**ul** class=**"table-title row"**>  
 <**li** class=**"js-tab-title col-6"**>**网站资质**</**li**>  
 <**li** class=**"js-tab-title col-6"**>**药师资质**</**li**>  
 </**ul**>

</div>

#### 4.8、轮播插件 ：data-act="swipe" data-param="auto=1"

\* 首页顶部轮播插件  
\* auto: 是否自动轮播 0 不自动轮播 1 自动轮播 默认1  
\*/

使用例子：

<**div** class=**"scroll-wrap"** data-act=**"swipe"** data-param=**"auto=0"** ></**div**>

#### 4.9、图片延时加载：data-act="lazy" data-param="url"

/\*\*  
 \* 图片延时加载  
 \* src 延时加载图片链接  
 \* style 延时加载背景图  
 \*/

例子

<**img** data-act=**"lazy"** data-param=**"src=**<%= *item*.image\_url %>**"** alt=**""**/>

<**a** href=**"**<%= *item*.location %>**"** data-act=**"lazy"** data-param=**"style=**<%= *item*.image\_url %>**"**></**a**>

#### 4.10、页脚菜单 ：data-act="footer"

/\*\*  
 \* 页脚菜单  
 \*/

// 主要使用在 src\template\footer.html 这个模板文件

<**footer** class=**"footer row"** data-act=**"footer"** ></**footer**>

### 5、ajax.js ：ajax请求插件；return [Function]

jax({  
 type: 'GET', // 请求类型  
 url: '', // 请求url  
 async: true, // 默认异步请求  
 timeout: null, // 请求超时  
 cache: false, // 是否使用缓存  
 data: {}, // 请求参数  
 header: { // 默认头信息  
 'Content-Type': 'application/x-www-form-urlencoded'  
 },  
 dataType: '', // 获取的数据类型  
 jsonp: '', // jsonp  
 beforeSend: function() {}, // 请求发送前回调  
 success: function() {}, // 成功回调  
 error: function() {} // 失败回调  
});

此插件的ajax请求可以使用缓存回调，例子：



### 6、event.js: 触摸事件插件；return null

/ 轻触，长按，滑屏，滑屏结束，左滑屏，右滑屏，上滑屏，下滑屏  
 'tap', 'longTap', 'swipe', 'swipeEnd', 'swipeLeft', 'swipeRight', 'swipeUp', 'swipeDown'

*Node*.addEventListener('swipe', handler, false);

使用例子

*Node*.addEventListener('swipe', function(*e*) {

//do something  
});

### 7、exbind.js：HTML控件定义插件；return exbind[Object]

/\*

\* @name 方法名(自定义)

\* @event 触发事件 例如： load ， click

\* *e*.param 传递的参数对象

\*/

exbind.register( @name , @event , function(e) {

var param = *e*.param,

node = this;   
 });

使用例子：

<**input** type=**"text"** data-act=**"calendar"** data-param=**"range=1"** />

如果传递的参数有多个，则参数间用 & 链接 ，例如data-param=**"a=1&b=2&c=3"**

exbind.register('calendar', 'load', function(*e*) {

var **range** = *e*.param.range, // 获取传递的参数

//do something   
 });

### mvvm.js：手机版MVVM插件；return MVVM[Object]

MVVM文档介绍网址：<https://github.com/flicat/mvvm>

MVVM对象的定义： var vmData = MVVM.define(controllerName, data);

MVVM声明控制区域：<div vm-controller="controllerName"> ....... </div>

MVVM控件类型的使用方法：

1. vm-html 绑定html赋值

<p vm-html="data"></p> //直接输出data

<!-- 使用内置artTemplate模板：-->

<div vm-html="data">

<script type="text/html">

<%= data.value %>

</script>

</div>

1. vm-value 绑定表单赋值（注意：表单赋值具有双向绑定，修改表单值的同时也会修改数据）

<input vm-value="data" type="text" />

1. vm-attr 绑定属性赋值（）

<div vm-attr-class="data"></div>

1. vm-css 绑定css属性赋值：

<div vm-css-display="data.value ? 'none': 'block'"></div>

1. vm-on 绑定事件：

<div vm-on-click="test()">按钮</div>

MVVM对象的方法：

vmData.\_\_update\_\_(); 当MVVM绑定的数据复杂，无法及时更新时，可以调用该方法，手动更新同步数据和视图

vmData.\_\_addObserveEvent\_\_(); //手动添加视图更新变化事件

vmData.\_\_removeObserveEvent\_\_() //手动删除视图更新变化事件

简单示例：

//js代码

var vmData = MVVM.define(controllerTest, {

input\_value: '',

text\_p: '测试测试',

text\_obj: {name:'小明', sex: '男'} ,

test:function(){

console.log(11111);

}

}});

//html代码

<div vm-controller="controllerTest">

<p vm-html="text\_p" ></p>

<input type="text" name="name" vm-value="input\_value" />

<div vm-html="text\_obj" >

<script type="text/html">

<p><%= text\_obj.name %></p>

<p><%= text\_obj.sex %></p>

</script>

</div>

<a href="javascript:;" vm-on-click="test()"></a>

</div>

### 9、Qrcode.js: 生成二维码插件 return QRCode[Object];

网上api文档介绍 <http://davidshimjs.github.io/qrcodejs/>

new QRCode(document.getElementById("qrcode"),"https://www.baidu.com/")

### 10、popup.js：弹窗插件；return popup[Object]

popup.success: function(*msg*, *callback*) {},

popup.success('成功提示');

popup.success('成功提示',function(){});

popup.error: function(*msg*, *callback*) {},

popup.error('错误提示');

popup.fail('错误提示',function(){});

popup.fail: function(*msg*, *callback*) {},

popup.fail('失败提示');

popup.fail('失败提示',function(){});

popup.loading: function(*act*) {}, // show:显示弹窗 hide:隐藏弹窗

popup.loading('show');

popup.loading('hide');

popup.share: function() {}

popup.share();

popup.confirm: function(*msg*,*surecall*,*cancelcall*) {}

popup.confirm(‘提示语’,function(){

//确认事件 do something

},function(){

//取消事件 do something

});

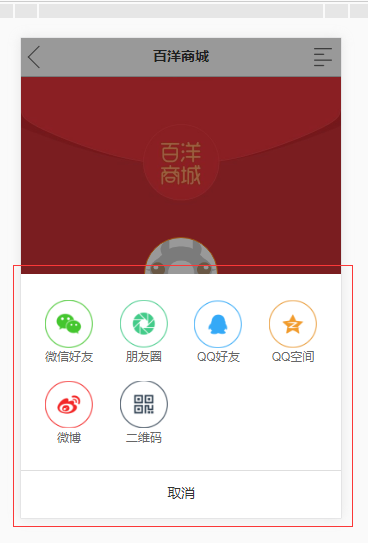
popup.shareGoods: function(){} //微信内 商品详情分享弹窗

popup.shareGoods();

### 11、share.js：分享插件；return Share[Object]

var shareObject = new *Share*({  
 title: '百洋商城-正规药房网上药店,妈妈网上买药的专业药店网站',  
 description: '百洋商城-正规药房网上药店,妈妈网上买药的专业药店网站',  
 img: seajs.data.cwd + 'images/icon-download.png',  
 url: 'String(window.location).replace(/(isband|token|unionid)=([^&]\*)(&|$)/ig,'),  
 type: 'link'  
});

shareObject.show(); //直接调用此方法，会弹出默认的分享弹窗样式



//重新修改分享文案

shareObject.setShareConfig({

title: '百洋商城-正规药房网上药店,妈妈网上买药的专业药店网站',  
 description: '百洋商城-正规药房网上药店,妈妈网上买药的专业药店网站',  
 img: seajs.data.cwd + 'images/icon-download.png',  
 url: 'String(window.location).replace(/(isband|token|unionid)=([^&]\*)(&|$)/ig,'),  
 type: 'link'  
})

### 12、public.js: 以下是已经定义好的方法，可以直接使用

// 从字符串中获取日期对象 , return data[Object]  
 Public.getDateFromString(*date*)

// 获取特定格式日期字符串, 返回字符串（string）  
 Public.getDateString(*date* , *format* , *isStamp*)

// 获取一个月的天数, 返回天数（number）  
 Public.getMonthDays(*yy*, *mm*)

// 根据传入参数返回当前日期的前几天或后几天, 返回字符串（string）  
 Public.plusDate(*n*, *format*, *isStamp*)

// 获取日期相差天数,返回相差天数（number）  
 Public.getDaysNum(*time1* , *time2*)

// 根据参数名获取URL中参数值，URL中没有该参数则返回null  
 Public.getSearchParam(*name*, *url*)

// 获取url传递参数 ， return param[Object]  
 Public.getUrlParam(*url*)

// 页面无刷新路由(通过hash跳转)  
 Public.pageRouter(*callback*) { }

// 页面无刷新路由(通过history跳转)  
 Public.stateRouter(*callback*) { }

### 13、router.js: 单页应用页面路由器

#### 13.1、register(url, callback[, is\_default, is\_route]) js定义页面动作

/\*

\* @url [String] 页面url,如果url省略 则默认为当前页面

\* @is\_route [Boolean] 是否需要初始化定位到当前页面  
\* @is\_default [Boolean] 当前页面是否是页面

\*/

<**div** page-router=**""> </div>**

<**div** page-router=**"?hwg\_agreement" > </div>**

router.register('',function() {} , 1 );

router.register('?hwg\_agreement', function() {} , 0 , 0 );

#### 13.2、onChange(handler)： 添加页面切换回调事

router.onChange(function(*page*) {

// do something  
});

#### 13.3、route(url, ignore) :手动路由到指定页面

/\*

\* @url [String] 页面url,跳转到指定的页面

\* @ignore [Boolean] 是否取消加入历史记录

\*/

// router.route('?invoice');

#### 13.4、registerNode(node, url)：手动注册DOM节点页面

#### 13.5、config(option)：配置页面切换样式

### 14、sea.js：seajs插件 网上介绍文档：<http://seajs.org/docs/>

### 15、jweixin.js：微信jssdk

### 16、log.js：手机版console工具；return log[Function]

debug= 关键词

[http://58.63.114.90:8074/index.html?debug=document.querySelectorAll(div);](http://58.63.114.90:8074/index.html?debug_ajax=check_login)

debug\_ajax = 接口关键词

<http://58.63.114.90:8074/index.html?debug_ajax=check_login>

### validate.js：表单验证插件

<https://github.com/flicat/form-validate>

// 文本格式验证正则表达式  
var regex = {  
 email: /^(\w)+(\W\w+)\*@(\w)+(-\w+)\*((\.\w+)+)$/, // email  
 phone: /^1[3|4|5|7|8][0-9]\d{8,8}$/, // 手机号码  
 tell: /\d{3}-\d{8}|\d{4}-\d{7,8}/, // 固话  
 number: /^[\-\+]?((\d+)([\.,](\d+))?|([\.,](\d+))?)$/, // 数字  
 integer: /^\d+$/, // 整数  
 date: /^\d{4}\W\d{1,2}\W\d{1,2}$/, // 日期  
 time: /^\d{1,2}:\d{1,2}$/, // 时间  
 cn: /^[^u4e00-u9fa5\w~!@#$%^&\*()\_+{}:"<>?\-=[\];\',.\/]+$/ig,// 中文  
 plus: /^[\+]?((\d+)([\.,](\d+))?|([\.,](\d+))?)$/, // 正数  
 url: /^[a-zA-z]+:\/\/(\w+(-\w+)\*)(\.(\w+(-\w+)\*))\*/, // 链接  
 password: /^[\w~!@#$%^&\*()\_+{}:"<>?\-=[\];\',.\/]{6,18}$/ // 密码  
};

// 例子

<**input** class=**"col-10"** type=**"text"** vm-value=**"account"** id=**"account"** data-validate=**"empty,phone|email"** placeholder=**"已验证手机号"**/>

### weixin.js：微信SDK签名 return wxsdk[Object]

### image-manage.js 图片压缩工具

可以参考 user-order-refund.js 里面有用到该方法

*Img 图像容器*

*Width压缩图片的最大宽度*

*Height 压缩图片的最大高度*

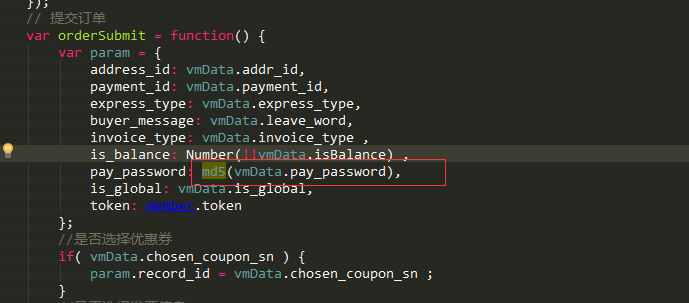
*Callback 回调函数*

image\_manage(*img*, *width*, *height*, *callback*)；

使用例子：

// 上传照片  
getFile: function(*input*) {  
 setTimeout(function() {  
 var imgFiles = [].slice.call(*input*.files);  
 var len = imgFiles.length;  
 if(imgFiles.length + mvData.update\_photo.length > 5) {  
 imgFiles.length = len = 5 - mvData.update\_photo.length;  
 }  
 imgFiles.forEach(function(*item*, *index*) {  
 var fileReader = new FileReader();  
 fileReader.onload = function(*e*) {  
  
 var img = new Image();  
 img.onload = function() {  
 // 压缩图片大小  
 var finally\_width = img.width > 640 ? 640 : img.width,  
 finally\_height = img.height > 640 ? 640 : img.height ;  
 image\_manage(img, finally\_width, finally\_height / img.width \* img.height, function(*canvas*) {  
 mvData.update\_photo.push(*canvas*.toDataURL('image/png').split(',')[1]);  
 len--;  
 if(!len) {  
 // 手动刷新图片预览列表  
 mvData.\_\_update\_\_();  
 }  
 });  
 };  
 img.src = *e*.target.result;  
 };  
 fileReader.readAsDataURL(*item*);  
 });  
 }, 0);  
} ,

### md5.js 加密 md5( string ) ;



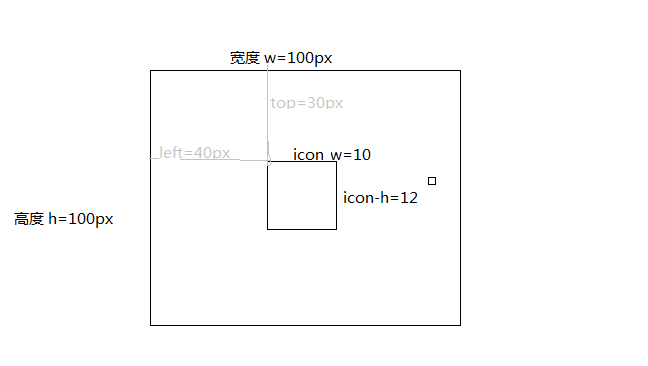
## 九、image文件

1、Css精灵图按照页面合并例如会员中心的图标合并到 user.png

2、合并的图片必须保存psd源文件

3、图片必须压缩：色彩单一的图片使用png格式存储；要求背景透明的图片也使用png格式存储；色彩鲜艳的使用jpg格式存储。Png格式的图片可以使用智图压缩工具，而jpg格式的图片可以使用fireworks压缩工具。

4、精灵图百分比计算方法：



icon-user {  
 *width*: (10 / 24**rem**);  
 *height*: (12 / 24**rem**);  
 *background*: url() **no-repeat** (40/ (100 - 10) \* 100%) (30/ (100- 12) \* 100%);  
 *background-size*: (100/ 10 \* 100%) (100/ 12 \* 100%);  
}