# Web前端开发体系设计

Version 0.1.0 by pakinguo

随着业务增多，Web前端页面颗粒度和复杂度也随之增加；而目前组内的开发体系没有统一的一套标准，可以复用的组件、模块在使用中有一定程度的不方便；业务与组件之间耦合度高；代码质量、部分优化性能还有很大的提升空间……等等。之前在ivy admin已经尝试过应用新的架构和框架来做研究实验，对于优化开发规范、效率和性能都有一定提高，同时参考了其他开发者设计的前端集成自动化架构，针对我们小组的开发模式，提出适应我们项目的前端开发架构体系：

1. 模块化&组件化
2. 开发规范&国际化
3. 性能优化&自动化

## 前端开发架构体系

### 模块化&组件化

* 应用Seajs框架，用于对js进行模块封装和按需调用，自定制Cssloader框架，对css进行按需调用
* 第三方框架cmd化，使其适应Seajs
* 通用模块组件化，组件由css与js封装，用工具输出其整合版

### 开发规范&国际化

* **文件结构规范**

|  |
| --- |
| project  ├ resource  ├ conf  ├ libs  └ mods  ├ \_\_src  ├ app  ├ common  ├ normalize.css  ├ common.css  └ common.js  ├ index  ├ index.css  ├ index.png  └ index.js  ├ login  ├ css  └ index.css  ├ img  ├ icon.png  ├ index.png  └ ...  ├ js  └ index.js  └ widget  ├ ivy.alert  ├ ivy.alert.css  └ ivy.alert.js  └ tpl.ivy.alert.htpl (--> tpl.ivy.alert.js 通过编译工具)  ├ en  └ zh\_TW  ├ view  ├ common  ├ head.tpl  └ foot.tpl  ├ index  └ index.tpl  └ login  └ login.tpl  └ i18n  └ i18n.po |

* **文件结构规范详解**

1. Web前端关注两个文件夹：resource和view（建议改为front和front\_view）；
2. conf放置不同页面需要的Seajs配置文件，其规范有待详细分析(TODO)；
3. libs放置第三方框架，包括未cmd化与cmd化不同模式，框架间用版本号区分

对于2、3点举例子：

Seajs的配置seajs.conf.js：

seajs.config({

base: "/cms/resource/",

alias: {

"jquery": "lib/cmd/jquery/2.0.3/jquery.js",

"jqueryui": "lib/cmd/jquery/jqueryui/1.11.0/jquery-ui.min.js"

},

debug: false

});

libs下的文件目录

|  |
| --- |
| projects  ├ resource  ├ conf  └ seajs.conf.js  └ lib  └ cmd  └ jquery  ├ 2.0.3  └ jquery.js  └ 1.11.0  └ jquery.js |

1. mods分业务逻辑层app和通用组件层widget，其上还有一层目录，为语言类型；开发时使用\_\_src，通过自动化编译工具（配合i18n文件夹）生成en、zh\_TW等不同语言；
2. app是页面逻辑业务模块，以页面为单位建立文件夹，其子文件夹结构可以灵活变动；
3. widget是通用组件层，一般是其他页面需要用到的通用的模块抽出而出，可通过编译工具输出单文件形式（待确认）；
4. 前端开发html、css、js都分离开来，利用定制的自动化工具，为自定义的某些后缀名html模板文件（如.htpl）编译成seajs模块文件或其他适用的js文件，配合模板引擎工具如[backbone](http://www.backbonejs.org/)、[underscore](http://underscorejs.org/)、[mustache](https://github.com/janl/mustache.js)、[HandlebarsJS](http://handlebarsjs.com/)等来进行前端渲染；
5. 国际化，包括mods下的所有模块和view层下的tpl，以文件夹命名语言来进行国际化（有别于本地化，本地化部分逻辑上需要调整，国际化只是翻译问题，关于本地化有待进一步考虑方案）；

### 性能优化&自动化

* 可应用sass、scss、less，优化css编程模式，并提供雪碧图自动合并生成css；
* 图片内嵌，将小图转化为base64格式；
* 减少请求，利用gruntjs对Seajs模块进行构建（合并压缩），可选择手动配置合并规则或根据项目定制自动化合并规则；
* 文件监听，提供部分功能即时编译机制；
* 国际化多语言翻译工具。

## 前端开发架构体系图示

