前端开发规范

1. 目录结构(参考,可根据项目再行划分)
2. Assets：动态资源（images，style，other）

2．Components：组件（common，header，footer，other）

3．Layouts：模板布局（detault，login）

4．Locales：语言包（en.json，fr.json）

5．middleware：中间件（浏览器端渲染前的事件处理，如i18n）

6．view：页面目录（index.vue，product.vue…）

7．Plugins：第三方插件（i18n.js，axios.js）

8．Static：静态资源（images…）

9．Store：状态树管理

例:(vue目录)

├─public

├─src

│ ├─api

│ ├─assets

│ │ ├─404\_images

│ │ ├─images

│ │ └─style

│ ├─components

│ ├─plugins

│ ├─store

│ │ └─modules

│ ├─utils

│ └─views

│ ├─becomePartner

│ │ ├─becomePartnerHome

│ │ ├─Benefits

│ │ └─Registration

│ ├─errorPage

│ ├─home

│ │ └─components

│ ├─integrationProtocols

│ │ ├─artificial

│ │ │ └─components

│ │ └─integrationProtocolsHome

│ ├─layout

│ │ └─components

│ │ └─Header

│ ├─marketSupport

│ │ ├─caseStudy

│ │ ├─e\_News

│ │ │ ├─components

│ │ │ └─e\_NewsHome

│ │ ├─e\_NewsDetail

│ │ ├─marketing

│ │ │ ├─events

│ │ │ └─solutionCatalogs

│ │ └─marketSupportHome

│ ├─productsSolution

│ │ ├─productsSolutionAny

│ │ └─productsSolutionHome

│ ├─projectSupport

│ │ ├─designTool

│ │ ├─projectRegistration

│ │ ├─projectSupportHome

│ │ └─samplesRequest

│ └─user

│ └─forgotAssets

└─tests

└─unit

二．代码规范

1．Js规范

（1）.尽量使用es6语法

（2）.使用eslint进行代码检查

（3）.命名

**命名方式** : 小驼峰方式 ( 构造函数使用大驼峰命名法 )  
**命名规则** : 前缀为动词

**例**:

| **动词** | **含义** | **返回值** |
| --- | --- | --- |
| can | 判断是否可执行某个动作 ( 权限 ) | 函数返回一个布尔值。true：可执行；false：不可执行 |
| has | 判断是否含有某个值 | 函数返回一个布尔值。true：含有此值；false：不含有此值 |
| is | 判断是否为某个值 | 函数返回一个布尔值。true：为某个值；false：不为某个值 |
| get | 获取某个值 | 函数返回一个非布尔值 |
| set | 设置某个值 | 无返回值、返回是否设置成功或者返回链式对象 |

（4）.类的成员

* 公共属性和方法 : 同变量命名方式
* 私有属性和方法 : 前缀为下划线(\_)后面跟公共属性和方法一样的命名方式

（5）.注释(参考[javadoc(百度百科)](https://baike.baidu.com/item/javadoc)

函数名语义化, 函数 参数 返回值 变量类型语义尽量写清楚

(推荐使用vscode插件document this)

| **注释名** | **语法** | **含义** | **示例** |
| --- | --- | --- | --- |
| @param | @param 参数名 {参数类型} 描述信息 | 描述参数的信息 | @param name {String} 传入名称 |
| @return | @return {返回类型} 描述信息 | 描述返回值的信息 | @return {Boolean} true:可执行;false:不可执行 |
| @author | @author 作者信息 [附属信息：如邮箱、日期] | 描述此函数作者的信息 | @author 张三 2015/07/21 |
| @version | @version XX.XX.XX | 描述此函数的版本号 | @version 1.0.3 |
| @example | @example 示例代码 | @example setTitle('测试') |  |

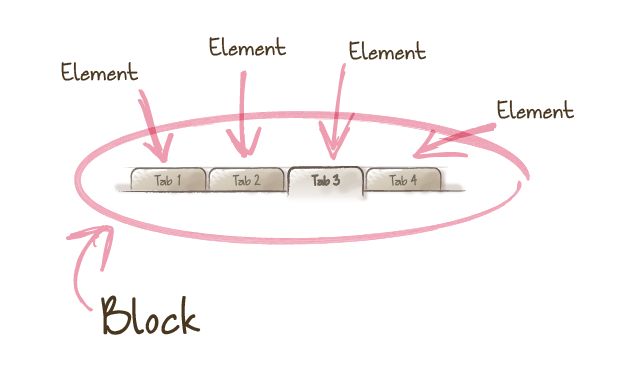
1. /\*\*
2. \* @desc各级模块首页
3. \*
4. \* @param {String} model model model =（products\_solution, integration\_protocols, become\_a\_partner, marketing\_support, project\_support,）
5. \* @param {String} type type = (home【首页模块导航】,left【侧边导航】)
6. \* @returns {Promise}
7. \*/
8. getHomeNav(model, type) {
9. // 单行注释说明 (// 后面与说明之间空一格)
10. return fetch({
11. url: `/dipp/home/nav/model/${model}/type/${type}`,
12. method: "get"
13. });
14. },

2．Css规范

（1）.使用less预处理器

（2）.盒子模型：IE盒子模型（box-sizing：border-box）

（3）.命名规范：BEM（block，element，modify）



如：.site-search{} /\* 块 \*/.site-search\_\_field{} /\* 元素 \*/ .site-search--full{} /\* 修饰符 \*/

（4）.书写规范：防止渲染回流（width，height，position，display，border等写在最前面）

（5）.页面适配：移动端,使用media控制不同分辨率的根字体大小（单位统一用rem）

3．HTML规范

* 使用 HTML5 的文档声明类型 : <!DOCTYPE html>
* 注意js脚本加载顺序 (项目只需要兼容ie10+或者只是在移动端访问，那么可以使用HTML5的新属性async，将脚本文件放在<head>内)
* 语义化
* 不要使用无意义空标签
* 结构、表现、行为三者分离

1. 其他规范

1. 注意浏览器兼容

2．IDE：VScode

3．Git提交（参考地址: [git commit 规范指南](https://segmentfault.com/a/1190000009048911#articleHeader7)）

* **feat**：新功能（feature）-工号

例：commit –m “feat 增添头部模块Header -50548

* **fix**：修补bug
* **docs**：文档（documentation）
* **style**： 格式（不影响代码运行的变动）
* **refactor**：重构（即不是新增功能，也不是修改bug的代码变动）
* **test**：增加测试
* **chore**：构建过程或辅助工具的变动

**4.** 所有素材图片应将文件名第一个单词命名为图片分类，第二个单词为图片名称，第三个单词可以是数字或其它内容.

例: banner\_xxx.gif  //广告条文件名

bg\_xxx.gif      //背景图片文件名