

## fisp学习——google主页

fisp的官方demo中只给出关于demo 的各种操作，初学fisp类型框架的孩纸不禁想如何能从零建一个项目？其实关键在于，根据目录规范（如下图）建立自己的各个文件夹，然后根据每个文件夹再开发出相应功能的文件，例如common是指通用库/基础库，就是放些复用性强的文件，如菜单组件等，common中又有几个子文件，page（模版页面，主要用于继承的模版页面，就是整个项目都需要用的部分）、widget（组件资源，如导航等、意在将页面内容聚合度较高的部分整合成一块）、static（能整合的块整合之后，剩下来的零散的资源）、plugin（用于存放smarty插件）、fis-conf.js（包括fisp的压缩打包等其他配置）

```
l---site
|   l---common #通用子系统
|   |   l---config #smarty配置文件
|   |   l---page #模板页面文件目录，也包含用于继承的模板页面
|   |       layout.tpl
|   |   l---widget #组件的资源目录，包括模板组件,JS组件,CSS组件等
|   |       menu #widget模板组件
|   |       |   menu.tpl
|   |       |   menu.js
|   |       |   menu.css
|   |       ui #UI组件
|   |       |   dialog #JS组件
|   |       |       dialog.js
|   |       |       dialog.css
|   |       reset #CSS组件
|   |       |   reset.css
|   |   l---static #非组件静态资源目录，包括模板页面引用的静态资源和其他静态资源
|   |   l---plugin #模板插件目录(可选，对于特殊需要的产品线，可以独立维护)
|   |   l---fis-conf.js #fis配置文件
|   l---module1 #module1子系统
|   |   l---test
|   |   l---config
|   |   l---page
|   |       index.tpl
|   |   l---widget
|   |   l---static
|   |       |   index #index.tpl模板对应的静态资源
|   |       |       index.js
|   |       |       index.css
|   |   l---fis-conf.js #fis配置文件
```

图1

- 1、建立主目录google-index，并在其下建立通用系统目录common,以及业务目录（此处是home）  
（注：可直接在phpstrom或其他编辑器中建立项目，以及其他文件及目录）

```
mdndeMacBook-Pro:~ mdn$ cd google-index/
mdndeMacBook-Pro:google-index mdn$ ls
common  home
```

2、在common中建立如图1中的目录，如下图

```
mdndeMacBook-Pro:common mdn$ ls
fis-conf.js  page  plugin  static  widget
```

3、同样的在home中建立类似目录，如下图

```
mdndeMacBook-Pro:home mdn$ ls
fis-conf.js  page  server.conf  static  widget
```

4、现在便可以在相应目录下分别开发出自己需要的文件，这里开发的是google的主页



5、首先需要建立整个项目都有用到的模版layout.tpl，放在common—>page中， {%block name="block\_head\_static"%}用于加载页面中需要的js\css文件， {%block name="content"%} 用于加载页面的内容。

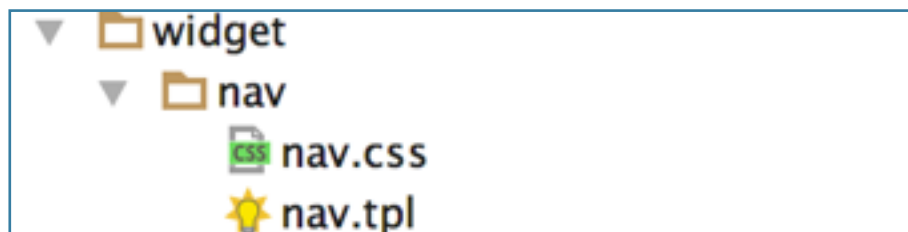


```

/* layout.tpl */
<!DOCTYPE html>
{%html framework="common:static/mod.js" class="expanded"%}
  {%head%}
    <meta charset="utf-8"/>
    <meta content="" name="description">
    <title>google-index</title>
    {%block name="block_head_static"%}{%/block%}
  {%/head%}
  {%body%}
    {%block name="content"%}{%/block%}
  {%/body%}
{%/html%}

```

6、在做google主页中，我们将顶端的Gmail/图片部分当作通用导航组件，则将其放到common—>widget目录中



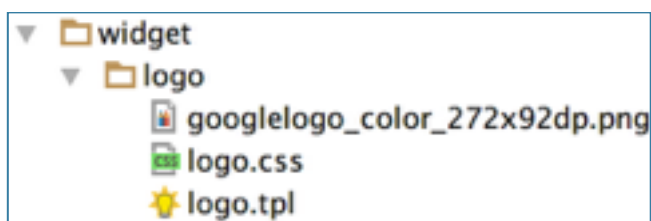
```

/* nav.tpl */
<nav id="nav" class="navigation" role="navigation">
  <ul>

    <li class="active">
      <a>
        Gmail
      </a>
    </li>
    <li>
      <a>
        图片
      </a>
    </li>
  </ul>

```

7.将google logo与japan合成一个组件，放入到业务模块home—>widget中，



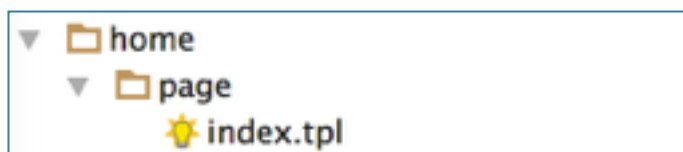
```

/* logo.tpl */
<section class="logo-container">
  <div class="img-container">
    
  </div>
  <div class="region">
    <span>Japan</span>
  </div>

```

8.之后，将搜索栏部分同样整合成一个组件，在此不再赘述。

9.我们已经将页面分成了各个部分/组件，还记得之前layout.tpl模版中设置的 {%block%} 吗？现在便是将其中的block“填充”上内容。在代码中我们可以看到分别有name为block\_head\_static、content的块，在content中，我们分别引入了nav.tpl/logo.tpl文件。



```

/* index.tpl */

{%extends file="common/page/layout.tpl"%}
{%block name="block_head_static"%}

  {%require name="home:static/index.css"%}
{%/block%}

{%block name="content"%}
  <div id="wrapper">
    <div id="sidebar">
      {%widget name="common:widget/nav/nav.tpl"%}
    </div>
    <div id="container">
      {%widget name="home:widget/logo/logo.tpl"%}
    </div>
  </div>
{%/block%}

```

10.到了此时，项目并没有结束，还需要配置fis-conf.js文件进行文件的打包或压缩等，可以防止后期因为缓存无法得到预期页面。

11.项目架构已经大致出现，现在便可以进入终端发布项目并启动进行调试。再次进入终端，进入到google-index目录下，执行如下命令，发布并启动项目。

```

$ fisp release -r common
$ fisp release -r home
$ fisp server start

```

