**前后端分离前端设计实现**

摘要:

由于复制快捷键把之前写好的文档修改后保存了。很郁闷，等抽空重新写。

**关键词：**

# 前后端分离的调研

前后端分离的技术已经非常成熟，但是基于前段的简单开发和调试还是不得法。总结一下前端开发的方法和技巧。首先明白要达到什么目的，怎么达到目的，怎样实现。

目的：围绕之前做的RBAC模型对相关数据进行页面维护。

1. 前端技术框架的搭建、代码的调试方法、前端与后端的数据沟通。真正做到前后端分离。

技术:

1. 前端框架使用静态资源HTML+CSS+JavaScript(jQuery+jQueryUI)搭建代码骨架。如果使用现有的框架来搭建环境无法体会到操作DOM之美。
2. 使用静态文本数据加载数据测试组件状态，完善中台接口实现CURD操作。
3. 前后端分离必定涉及到跨域访问的问题。

实现：

## 前端环境搭建

目前我接触到的视图层方案如下：

1. jsp+HTML+jQuery

HTML标签和js代码会混合在一起。在搭建好的框架中，每编写一个功能页面都要在HTML标签中穿插js代码的引用。

1. HTML+webix

可以基于MVVM方式编写前端的代码。纯粹的编写js代码。

暂时还是使用第一种方法编写前端代码，但是在前后端分离的基础上进行编写代码，不会涉及jsp代码。

### 规范

1、HTML代码和js代码文件存放文件分离，放在同一个功能模块下面，这样有利于维护（之前处理过jsp文件和js代码夸文件存放，导致维护很难）。代码分为公共代码和业务代码。布局代码、插件代码、功能代码。

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
| HTML | name | .html |  |
| JS | name\_js | .js |  |
| CSS | name\_css | .css |  |

2、页面布局样式大小统一采用百分比。

### 引入依赖文件

按照网络资源进行搭建开发环境。

<https://www.cnblogs.com/xdp-gacl/p/4075079.html>

### 设置布局

使用jQueryUI构建视图。

[https://www.cnblogs.com/xdp-gacl/p/4088198.HTML](https://www.cnblogs.com/xdp-gacl/p/4088198.html)创建页面的布局。

<https://www.cnblogs.com/1312mn/p/4015455.html>

突然不知道自己想做成什么样式的前端开发框架。可以从两个出发点来考虑：

1. 现在主流的存在什么样式的前端交互方式。
2. 不同的交互方式适合哪类使用场景。

实在是无法穷举前段交互的方式，不过就我接触的两个项目来看，可以分析一下。

上一个政府项目布局是北、西、南、中。北展示下拉式菜单，菜单可以做到控制权限;列出了相关的操作。西展示的是组织、资源树，在单个组织展示数据时候显得很留白多，浪费了页面空间。南展示了一般的软件所属单位信息。中展示了主要的业务数据和数据修改空间。

优点：

缺点：下拉式菜单并在一列中，向左扩展，多级菜单展开后页面不优雅。页面只能打开一个使用页面，页面无法形成多页签缓存。

<https://www.cnblogs.com/szytwo/archive/2012/08/29/2662150.html>

任意一个新的功能都是以tabs 的方式打开新的页签，打开过的部分不再打开。

## 搭建Nginx

### 引用本地调试文件

使用EasyUI写了几个模块后发现要加载本地json数据做测试，发现浏览器跨域访问了。两种方法解决[修改浏览器配置](https://blog.csdn.net/yidboy/article/details/57482439)（客户端行为），使用Nginx反向代理（服务端行为）。NSSM+Nginx方案：NSSM服务用于系统开机自动启动Nginx服务，Nginx反向代理。本地搭建Nginx修改配置文件nginx.conf文件，配置文件如下：

|  |
| --- |
| listen 80;  server\_name localhost;  #charset koi8-r;  #access\_log logs/host.access.log main;  location / {  root F:/project/01RBAC/03code/rbacUI;  } |

本地的文件地址修改windows路径分割符号为左斜杠“/”。端口使用默认80，后续加入后端服务后还有再次修改。

|  |
| --- |
| 远程代理  location /api/ {  proxy\_pass http://127.0.0.1:8080/;  proxy\_set\_header Host $http\_host;  ; |

### 调用远程服务

由于自己的电脑配置太低，关闭spring boot 的数据持久层，从而不用启动数据库服务。放弃，由于之前的后端代码中间耦合了很多DAO层的代码，关持久层的工作量太大。还是自己新建一个后端项目供自己测试使用。

## 前端菜单导航

花了半天时间找[EaysUI实现横向导航菜单](http://www.jeasyui.net/demo/370.html)的方案，但是样式很不满意。看了半天CSS样式中有关[横向和纵向菜单](https://www.runoob.com/try/try.php?filename=trycss_navbar_vertical_borders&basepath=0)的实现。知道了菜单的有关特效的实现方法，基本可以自己实现菜单。通过原始的CSS去修饰EasyUI导致原有的layout样式完全失效。百度后没有找到兼容这两者的直接方法，转而一想我的目的不追求前端具体对技术实现细节，还是以实现纵向菜单来写代码。纵向的region：west可以有折叠的功能，这样看来并不会影响功能界面的展示区。

纵向的菜单功能区有两部分：

1. 使用Accordion（手风琴）来实现。Content 中填充列表实现。样式由jQuery UI控制。
2. 使用树来实现二级菜单甚至是三级菜单。

动态加载数据的具体的实现：

<https://blog.csdn.net/hao134838/article/details/49227505>

继续找，找到一个可用的

<https://blog.csdn.net/J_Demon/article/details/47259399>

一些图标

<https://blog.csdn.net/qq_35393869/article/details/82702164>

## 前端权限控制

<https://blog.csdn.net/yin_you_yu/article/details/80408340>

前端权限的控制暂且放一放。首先把权限的维护工作做好之后，回过头来处理权限的问题。

## 表格的加载

1. 使用jQuery easy UI 建立的表格
2. 建立可编辑的表格，实现数据的修改、保存、提交、刷新。

这篇文章中有对这样的案例进行介绍，看来比较复杂，要花点时间进行研究。

民间案例：

<https://www.cnblogs.com/linybo/p/10054276.html>

官网案例：

<http://www.jeasyui.com/demo/main/index.php?plugin=DataGrid&theme=material-teal&dir=ltr&pitem=&sort=asc>

遇到一个坑在工具toolbar.hander中引用函数，如果在HTML中进行绑定事件，可以直接引用内联js函数，但是在jQuery中引用函数要使用匿名函数，匿名函数可以使用function（定义函数）。

1. 数据的加载

数据加载使用组件ajax 方式加载，如果是加载本地json文件，要指明get方式。

1. 数据的维护和提交

数据的维护分为编辑、取消改变、接受改变。当然在体验上还需完善。

|  |
| --- |
| $('#dg').datagrid('getChanges','inserted'); |

## 表单整理

使用Jquery UI表单进行加载，样式控制起来优点困难。

## 弹出窗

使用Jquery UI的弹出窗非常方便，控制好弹出框的状态

## 延迟加载

延迟加载难在对数据结构的扩充

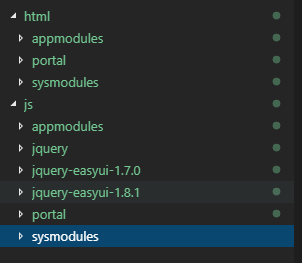
## 项目结构整理

这个页面还需要改造，原因如下：

1. 代码应该分为三部分，骨架代码、系统功能代码、业务功能代码。这样代码后期维护的时候很方便。
2. 此页面用作index.html页面，作为主页面加载其中不能夹杂太多功能模块代码，这样index.html页面显得很臃肿。
3. HTML文件和js文件要分开放置。

调整过程中遇到的问题：

1. 按照模块放置分文html和js文件夹分别放置代码



1. 为了测试和调试方便之前的index.html页面中内嵌了js代码，url中加载的是本地的json文件通过nginx反向代理后浏览器可以夹杂本地的json文件。但是，html与js代码分文件夹存放，同时把js中引用的json文件也迁移至js代码的同文件夹保存原来js文件与json文件的相对路径不变，结果js中无法加载到对于的json文件数据。
2. 分离出来的功能代码出现了jQuery-easyUI所有相关的文件都找不到404.

分析：所有的引用路径都发生了变化。jQuery相关的库文件无法加载时由于功能的index.html页面中还是引用的第一次开发页面的js库引用方式。导致相对地址不准确。js中的代码加载json文件无法找到也是url路径发生变化导致。在客户端html引用js代码后进行加载解析，相当于是在html中解析js代码中的url，此时的url相对地址还是在html中。把js对json文件的引用修改为html中引用json文件的地址后，页面可以加载数据了。

有三点要解决：第一，每个html 文件中都要映入jQuery 库很麻烦，不同层级下的html文件引入同一个jQuery库的写法还不一样。第二，在js文件中引入文件的路径很诡异。第三，如何在js文件中引入js文件，封装组件的问题。（这个还需要加强，可以把js与html和css分开来看，避免了html和css对js代码的干扰）。

暂时不考虑代码的封装，还是分开放置，在使用的过程中对jQuery使用的比较属性了再考虑封装的问题。

# 总结

## 通过之前的找资料

HTML:HTML定义了骨架，当然包括事件。

CSS:定义了骨架的模样，我们看到的好多动态样式都是CSS完成的。

JS:是对前两种已有的属性进行操作。

## Jquery

方法：在代码中进行操作节点、读取节点参数。

事件：在空间加载完成后在客户端进行操作出发的。

属性：在空间进行加载的时候设置参数值。

Jquery的封装

$.data

## CSS样式归纳