# 前端开发手册

|  |  |  |  |
| --- | --- | --- | --- |
| **版本** | **说明** | **创建者/修改者** | **创建日期/修改日期** |
| V1.0 | 初稿 | 邹琼俊 | 2017-07-26 |
| V1.1 | 修稿 | 邹琼俊 | 2017-07-26 |

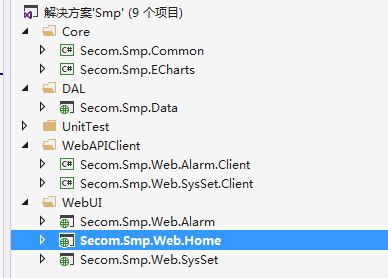
## 适用对象

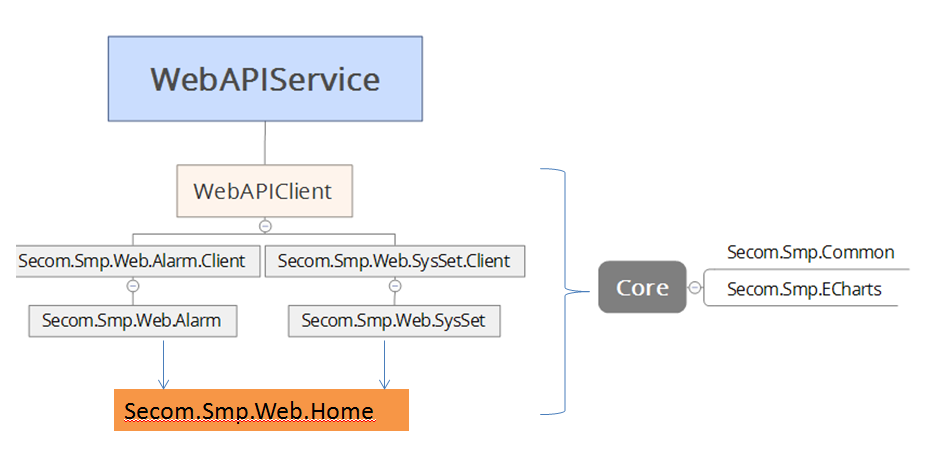
所有研发人员

## 开发环境

VS2015、MVC5

## 项目结构





|  |  |
| --- | --- |
| **项目名称** | **说明** |
| Secom.Smp.Web.Home | Web主站点 |
| Secom.Smp.Web.SysSet | Web模块站点，系统设置 |
| Secom.Smp.Web.Alarm | Web模块站点，报警管理 |
| Secom.Smp.Web.Alarm.Client | 报警管理模块的WebAPI客户端调用 |
| Secom.Smp.Web.SysSet.Client | 系统设置模块的WebAPI客户端调用 |
| Secom.Smp.Data | 测试用，生产都是通过API调用。 |
| Secom.Smp.Common | 公共模块 |
| Secom.Smp.ECharts | ECharts组件封装 |

设备运维 maintain

设备监控 watch

报警管理 alarm

能耗监测 energy

系统设置：SysSet

首页：Home

-------------------------------------------------------------

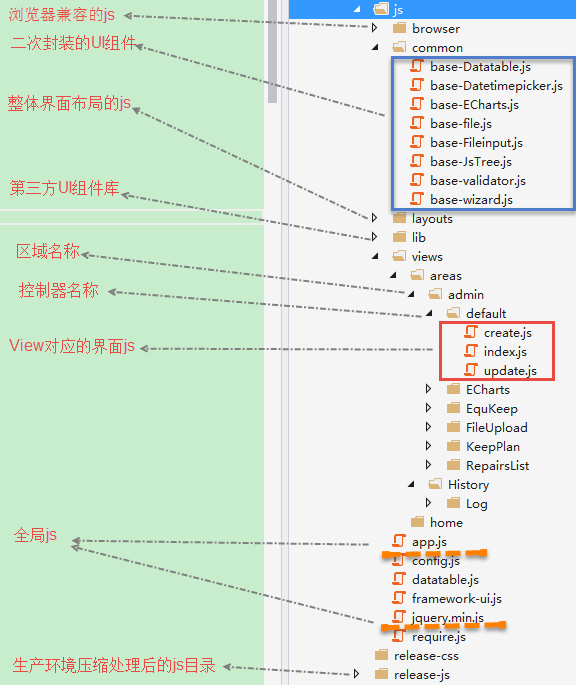
设备维修：EquipRepair

设备保养：EquipMaintain

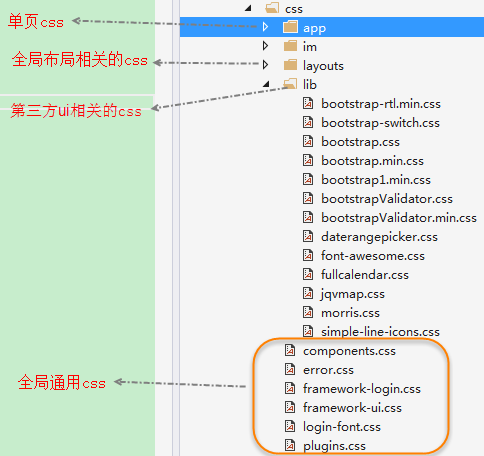
### Secom.Smp.Web.Home目录

#### Js目录结构

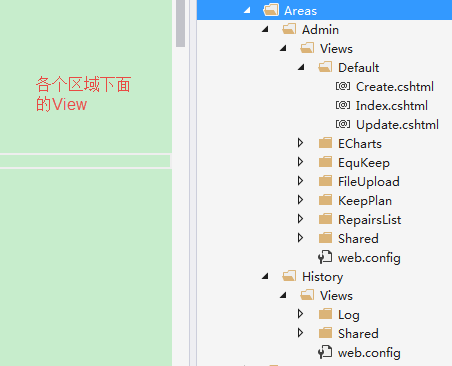




#### Css目录结构



#### 视图



尽量不要在View中直接写js和css，那种基本上只这个界面用一次并且低于50行代码以内的可以例外。

放到独立的文件中一方面可以复用，一方面可以利用缓存减少流量。

### 布局

系统后台布局采用的css框架是Bootstrap v3.3.7，异步刷新采用的是ajax，jquery的html()和load()。所有的UI组件优先考虑Bootstrap相关的UI组件，保障风格统一，和响应式布局。

## 注意事项

说明：请开发人员务必严格遵守。

### 引入js的方式

**单界面js引入如下：**

<script src="@Html.ScriptsPath("common/base-Datatable.js")"></script>

说明：这是对js的引入进行了封装，一是方便发布到生产时统一压缩，二是方便以后js独立部署后的引用。

**合并压缩的js引入方式如下：**

@Scripts.Render("~/bundles/core")

说明：开发环境下，默认不合并和压缩，方便开发调试，发布到生产时，release模式会自动进行压缩。动态加载的JS问题

通过ajax动态加载过来的js文件，默认情况下，你在前端是无法调试的，需要在被引入的js文件的最顶端加上如下代码：

//# sourceURL=自定义js文件名.js

可以参照js/common中的相关js。

### 引入Css的方式

考虑到我们独立的CSS文件不会太多，更多的是一些比较通用的css文件，为了方便合并和压缩，统一通过如下方式调用

@Styles.Render("~/Content/lib/css")

### 异步加载界面

两种方式

1、 直接在超级链接中输入需要加载进行的页面地址，同事设置class为“ajaxify”

<a href="@menu.LinkUrl" class="ajaxify nav-link">

需要注意的是，这将默认加载页面到Div Id为“page-content”的主容器中。详情见左侧菜单的链接方式。SubMenu.cshtml

2、通过方法调用

Layout.loadAjaxContent('/Home/LoadMenu?name=@menu.Name',undefined,'leftMenu',true)

方法参数： //界面地址，[需要定位到的菜单模块]，[容器ID]，[是否显示加载中图片]

### 使用MVC自带的压缩合并时要注意的问题

当我们引入一些第三方的UI库，默认是带着有xx.js、xx.min.js，xx.css，xx.min.css这样的文件。

如果我们改UI组件源码的时候需要注意：

我们知道使用@Scripts.Render 方式引入js文件，当我们不在代码中显示指定是否压缩合并的情况下，在Debug模式下默认是不进行压缩合并的，然而当发布到生产为Release模式下，就会开启压缩合并。而且我猜测其有一个处理机制，假设项目中存在1.js、1..min.js的情况下，如果你修改了1.js的代码，发布的生产的时候，压缩后将使用1.min.js中的代码，而不是压缩1.js代码。所以，如果你修改了1.js的代码，要么就不要在项目中存在1.min.js的代码，要么就用压缩工具手动压缩1.js成1.min.js再放到目录中。[这可能是@Scripts.Render](mailto:这可能是@Scripts.Render)的一个优化机制，当它检测到有已经压缩过的代码（.min.js后缀的文件就是认为是压缩过的）就不会再压缩而是直接使用相应的.min.js文件。

Css文件的处理也是同样的机制。

那么我们不要直接修改第三方组件的css和js，而是通过在他们的扩展文件中进行修改。Css组件样式就到plugins.css这个文件中加，页面自定义样式就到custom.css中加，这个文件会放到所有其他页面之后进行加载。js就通过二次封装的形式进行扩展。

自己在页面中手写的css样式全部写到/Content/css/app/custom.css文件中，所有View界面中不要存在任何颜色控制的代码，严禁直接硬编码，如：<div style=”color:red”>，因为后面可能会存在换主题。单个页面特有的非颜色样式可以例外。

### JS说明

进行二次封装的UI组件我都是将其包装成js对象的，通过js引用后，这个对象就是一个可以全局访问的入口。

所有js方法注释中的参数如果是用[]包起来的，表示可选参数。

由于系统后台布局是采用的ajax，所以手写的js注意命名不要和Layout布局页中引入的js组件等冲突，建议包装成js对象，只开发一个访问的入口，尽量避免全局变量污染。

#### 关于Datatables的使用

通过2个Demo分别演示了客户端分页和服务端分页的用法。

客户端分页：【用户列表】页面，当表中数据量非常少的情况下，一般250条记录（经验值）以内可以考虑一次性取出存到内存中进行分页，减少请求次数。否则使用服务端分页。

服务端分页：【日志列表】页面。

#### 寻找UI资源

可以访问我本地部署的站点：http://192.168.1.95:8002/，里面有许多可以参考的界面，包括js和css以及img、fonts等。

#### 关于报表

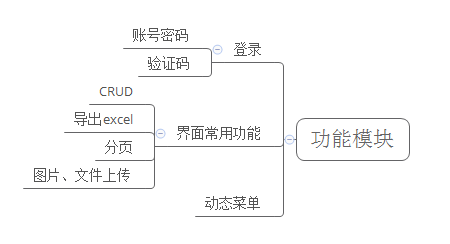
所有的报表使用百度的ECharts，以下提供2个官网地址：

<http://echarts.baidu.com/examples.html>

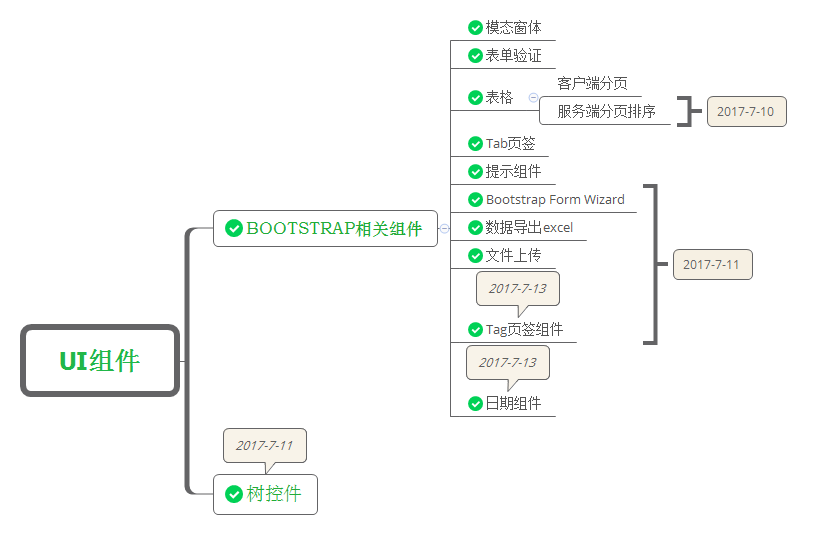
<http://echarts.baidu.com/index.html>

开发报表时，先去<http://echarts.baidu.com/examples.html>找到类似的报表实例，然后参考官方文档，在线配置出options，确认没问题之后，再通过C#代码来构造options的JSON格式

## 已完成功能



## 已封装的UI组件



## 演示地址

Mvc站点：http://192.168.1.95:8001

账号密码：admin/123

UI资源：http://192.168.1.95:8002/

## 说明

为了开发调试方便，我暂时是直接在Secom.Smp.Web.Home主项目中添加视图的，没有在Secom.Smp.Web.Alarm等模块项目中添加视图，然后Build的时候再自动Copy到Secom.Smp.Web.Home中的指定视图目录中。

而且，我Secom.Smp.Web.Home是直接添加了Secom.Smp.Web.Alarm等模块的引用，方便调试。

## 问题

如果在Secom.Smp.Web.Alarm、Secom.Smp.Web.SysSet等模块中独立开发、独立部署，有两个问题需解决：

### 资源、登录、权限验证等共享问题

如何共享Secom.Smp.Web.Home中的静态资源，我想到的是：1、通过按Shift+Alt 选择Secom.Smp.Web.Home中的Content目录拖动到Secom.Smp.Web.Alarm中，通过创建一个快捷方式。2、将静态资源独立部署到站点。

登录权限等共享的问题：部署时好办，放到同一个目录下面就可以，但是开发的时候怎么办？在Secom.Smp.Web.Alarm中添加Secom.Smp.Web.Home的引用吗？

### 调试问题

如果不添加项目的引用（也就是没有产生依赖引用），而是直接一个dll在bin目录下，如何调试？

这是我困惑的两大问题，不知道有没有好的解决方案？