# 1. 框架选择

**基础：**

jQuery：<http://jquery.com/>

jQuery UI：<http://jqueryui.com/>

**MVC框架：**

Backbone：<http://backbonejs.org/>

Mustache：<https://github.com/janl/mustache.js>

jquery-Mustache：<https://github.com/jonnyreeves/jquery-Mustache>

**UI组件：**

ESUI：<http://errorrik.com/er/#esui>

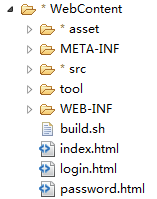
HighCharts：<http://www.highcharts.com/>

Plupload：<http://www.plupload.com/>

使用方法见官方文档和demo即可。

# 2. 文件部署

## 2.1 根目录



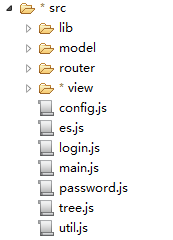
|  |  |
| --- | --- |
| 目录 | 说明 |
| asset/ | 静态资源文件 |
| src/ | js源码 |
| tool/ | 压缩工具 |
| build.sh | 加时间戳（防缓存），合并压缩css、js，上线前执行一次 |
| \*.html, \*.jsp | 页面 |

## 2.2 资源目录



|  |  |
| --- | --- |
| 目录 | 说明 |
| asset/css/ | CSS |
| asset/img/ | 图片 |
| asset/js/ | js导入 |
| asset/swf/ | Flash |
| asset/tpl/ | html模版 |

## 2.3 源码目录



|  |  |
| --- | --- |
| 目录 | 说明 |
| src/lib/ | 类库 |
| src/models/ | Model声明 |
| src/router/ | Router声明 |
| src/views/ | View声明 |
| src/config.js | 常量配置：  **DEBUG**：调试log开关，上线前关闭  **MOCK**：假数据开关，上线前关闭  **CRF\_MENU**：CRF菜单项  **STAT\_MENU**：全局统计菜单项 |
| src/es.js | 名字空间声明 |
| src/main.js | 主程序 |
| src/tree.js | 树形菜单类 |
| src/util.js | 工具 |
| src/login.js | 登录 |
| src/password.js | 修改密码 |

# 3. 系统执行流程图



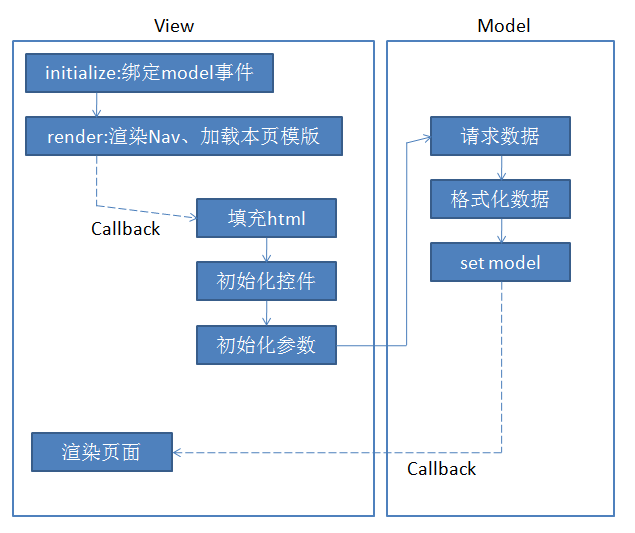
一些要点：

* Hash即index.jsp# 后面的字符串，定义在src/router/Router.js中
* 不同的Hash会创建不同的View及对应的Model，同时销毁旧的View，

当前的View由es.main引用

* 每个View在创建时会自动调用initialize方法来绑定Model事件
* 每个View必须定义render方法，用于绘制页面
* 除了Head、Nav和Foot，每个View必须定义destroy方法，用于销毁自身

View的执行流程：



# 4. 如何开发一个新页面

1. 在Router.js中的rout对象中加入新URL和回调函数
2. 在models目录新增model声明，在views目录新增view声明
3. 编写Model和View的逻辑：

Model主要负责数据处理逻辑，比如请求数据、格式化数据等等；

View主要负责模版的加载、控件的初始化、事件的绑定、页面的销毁等等

1. 在asset/js/all.js中加入本次新增的js文件的引用