利用yeoman搭建的目录结构如下：



App文件夹的目录结构：



App是项目的开发目录，主要的coding工作都在这里，文件夹以最简洁的方式区分：

Scripts中存放所有的自定义js文件（controller、service、factory）

Views中存放所有自定义html页面和模板

Style中放所有的自定义css文件

Images中放所有的自定义图片和自定义资源等

由于用到kingadmin，所以多了assets和demo-style-switcher两个目录。（最终可能会调整到对应目录下，原因是kingadmin的js不是用angularjs，所以必定会做调整。）

所有自定义添加的html和js需要引入到index.html中。

启动：在上一级目录运行grunt serve后，会自动启动app目录下的index.html。

Scripts目录下的结构：



App.js：控制路由切换页面，可以在里面配置。

config.js：任何配置文件内容往里面填，例如api host、版本号等。（自定义的）

startup：额外的启动文件，用于在某些情况下启动时需要写的内容。（自定义的）

controllers和services是为了区分两类文件

（如果觉得不好，这里甚至可以不用文件夹区分，完全自己控制。）

以上就是整个项目的前端文件结构，它的优势在于：

1. 快速搭建环境，文件结构合理、简洁。可以快速启动网站进入调试，增加开发效率（自动刷新调试窗口）

2. 用bower解决了依赖项的问题，如果后续有需要升级也支持。（类似之前用的nuget）

3. 解决了部署时会遇到的问题：

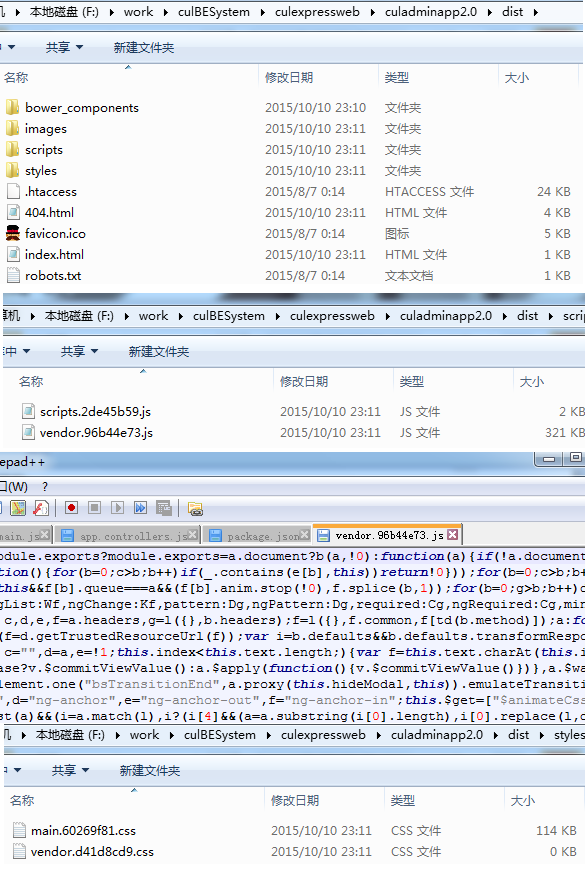
css压缩；

js压缩；

依赖项的资源文件存放位置

浏览器缓存（重要）

最后看下最终项目发布后的文件结构(dist目录)：



环境依赖：

git、yo、grunt、bower；

运行方式:

1. 打开当前目录的cmd窗口

2. 安装依赖项:

npm install

bower install

3. 安装成功后，仍然在当前cmd窗口中输入grunt serve即可启动项目

4. 框架支持debug和release两种模式

debug: 即上面说的grunt serve模式

release: 输入grunt build命令编译，会在当前目录下生成dist文件夹。