Joint Venture 2k15

[Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.]

[Type the document subtitle]

Table of Contents

[Categorizing Application Units 1](#_Toc427518162)

[Back End 1](#_Toc427518163)

[Front End 3](#_Toc427518164)

[Project Folder Structure 3](#_Toc427518165)

[Other Backend Files 5](#_Toc427518166)

[Front End Files – under the ‘public’ folder 6](#_Toc427518167)

# 

# Categorizing Application Units

1. Back End – nodeJS , Express JS, MongoDB,
2. Front End – AngularJS, Bootstrap, etc

# Back End

The application backend is nodeJS with expressJS as middleware and uses MongoDB as Database.

# Front End

The application front end – that is the web site when opened in the browser runs on HTML, CSS3, Bootstrap, jQuery and most importantly ‘AngularJS’.

# Project Folder Structure



A closer look into the files in the root folder, we have a list of files described in the table.

The basic must have files –‘package.json’ and ‘bower.json’



|  |  |
| --- | --- |
| .bowerrc | has the path of the folder where bower install will download the dependencies. Currently it is ‘public/lib’ |
| .gitattributes | has the type of extension and applications associated with it |
| .gitignore | list the files/folders to be ignored by Git |
| 0.firstRun.bat, 1.startMongoDB.bat, 2.sartNodeServer.bat, 3.closeAll.bat | Batch files to be run in that order to launch the application in the web browser. |
| bower.json | The target file used by the command ‘bower install’. This file has the list of front end dependencies , for example, Bootstrap, AngularJS, Material Design. Etc. |
| Gruntfile.js | Grunt is a Javascript task runner. We define list of tasks that can be triggered from Grunt. For example, everytime we create a new module, we add new folders in the ‘public/modules’ folder, but we then concatenate all the ‘js’ files together and minify it to a location ‘public/assets/js/angular/angApp.js’  This task and many others can be run using Grunt. |
| package.json | The target file used by the command ‘npm install’. This file has the list of dependencies used by nodeJS application. For example, we need expressJS, BodyParser, Grunt, etc. |
| server.js | This file is the ‘curx’ of our application. It stitches various individual files together, sets up an Express JS Server, and responds to the requests received. |
| projectWalkthrough.docx, Readme.md | Are help files about the application. |
| sublime-project.sublime-project | This fie is a help file used to load sublime text editor with project folder navigation. It has no impact on the application. |

# Other Backend Files



|  |  |
| --- | --- |
| jv.js | This file has the schemas of different collections ( tables ) of joint venture database |
| routes.js | Contains all the possible routes and their actions as handled by the ExpressJS server. For example, when the user visits the location localhost:8080/api/createSampleUser  Will create a sample user in the ‘users’ collection in the db.  Localhost:8080/api/checkLogin – will use the username, password used in the POST and verifies the User in the ‘users’ collection. |
| db.js | A configuration file that says the database is available at which location and port |

# Front End Files – under the ‘public’ folder



|  |  |
| --- | --- |
| Assets | The css, img, and js files used. The ‘css’ folder contains the ‘css’ generated by the Grunt task (LESS compiler ) – We donot change a css file here, we change it in its corresponding LESS file.  Similarly, the ‘js’ folder contains the ‘angApp.js’ which is cumulated and at time minified version of all other ‘angular’ related files in the application. It is generated by the Grunt task. |

|  |  |
| --- | --- |
| libs | This folder has all the css, js, etc specified in the ‘bower.json’. We donot modify anything here. |
| modules | This folder is a well organized containing different folders. Organized means, the folders are named in a specific way, as they are appear in the application. |
| templates | This folder has all the ‘.html’ files that contain the ‘div’ content of different angular directives used in the application. |
| views | This folder has the ‘div’ contents of various views. These are basically ‘.html’ files that are loaded dynamically whenever different links are clicked on the application. |
| index.html | The root/base file that will loaded by the ExpressJS for all other ‘requests’ apart from back end. This file is the skeleton of the UI of whole application. |

# Front End – Modules

This folder is well organized. It contains list of modules ( Visual Separation ) present in the application.



As per the naming/numbering convention, 0 contains the absolute basics, and it will loaded before others in the page.

1, is the first visible component / module on the application. In our case, it is the navigation menu.

The navigation menu has, a ‘login’ link, so the login modal (popup) is a sub module of ‘navigation’ and hence the number 1.2

And if you look at the landing page, you can identify the components displayed in that order.

To make it convenient to map the code.

Going a level deeper.