

Objectives

Create a file management UI following the recommended guidelines provided by the customer.
Create a file management RESTful api with a database backend to save user and file information

Primary Technologies

- Angular 10 framework for UI
- Angular Material - UI design
- NodeJS - Defacto standard for service side Javascript
- ExpressJS - simplify the setup of RESTful apis
- MongoDB for the Data Layer
- MongooseJS - ORM for the objects you interact with.
- Gridfs - a middleware to chunk file data into smaller parts, due to the 6 mb limit in native mongo
- Multer - An express middleware to parse file data in the req object.

Primary Goals

- Create a Plan. Due to limited time. Breaking down the plan into ten ~ 2 hour sessions. Making sure that I account for development time and documentation.
- Keep it simple. There is no time to get fancy with special widgets unless time allows it.
- Start by using simple constructs first then extend as needed.
- Maintain work within a 20 hour window.
- Complete the work if possible.

Breakdown of Work

- ~~REST API with Mongo Backend along with Grid File and User Data~~
 - ◉ ~~Login Endpoint~~
 - ◉ ~~User Endpoint~~
 - ◉ ~~File Endpoint~~
 - ◉ ~~Postman Collection~~
- Web UI - Angular
 - ◉ ~~Login Screen~~
 - ◉ ~~Login Functionality and Service~~
 - ◉ ~~Register Screen~~
 - ◉ ~~Registration Functionality and Service~~

- ~~○ Navigation Screen~~
- ~~○ Navigation Functionality~~
 - ~~■ Side Nav Functionality~~
 - ~~■ Top Bar Functionality~~
 - ~~■ Routing Functionality~~
- ~~○ Home / File Screen~~
- ~~○ Home / File Functionality~~
 - ~~■ File / Folder Card~~
- ~~○ Create Share Dialog~~
- ~~○ Share Functionality / Service~~
- ~~○ Create Upload Dialog~~
- ~~○ Upload Functionality / Service~~
- ~~○ Create Download Functionality~~

Out of Scope

- Complex security elements such as identity and certificates or database credentials

Training

- I have never had to work with files before using Angular. I had to research and learn the following:
 - GridFS with Mongo
 - Multipart Forms with Angular

Plan

9/18/2020

- ~~● Create the DB Layer with NodeJS and Express RESTful api, keep the implementation fairly simple and leverage boiler plate code if it makes sense~~
- ~~● Start documentation in README file~~

9/19/2020

- ~~● Continuing and Finalized building out the API and Database Layer~~
- ~~● Setup a Postman Collection to test the endpoints.~~
- ~~● Define the models of the object to be used.~~
- ~~● Continue documentation in README~~

9/21/2020

- ~~Stub out the scaffolding to include the UI in the project by using the ng cli~~
- ~~Setup Navigation and ToolBar~~
- ~~Create Login Screen~~
- ~~Create Register Screen~~
- ~~Continue documentation in README~~

9/22/2020

- ~~Create angular login service~~
- ~~Create Current User service~~
- ~~Populate the side bar with user info~~
- ~~Wire up the Nav and Side bars~~

9/23/2020

- ~~Create angular file service~~
- ~~Create the folder and file screen~~
- ~~Continue documentation in README~~

9/24/2020

- ~~Reassess progress~~
- ~~Create the file download process~~
- ~~Update documentation~~

9/25/2020

- ~~Create the file upload dialog~~
- ~~Update the file process as needed~~
- ~~Logout~~
- ~~Registration~~

9/26/2020

- Create the file sharing dialog
- Create the file sharing process

Daily Log

9/18/2020 7:01 pm - 10:01 pm (3 hrs)

- Worked on the API. Setup Routes and Mongo Service
- Did not finish what was planned
- Created initial README
- Will continue tomorrow changing plan

9/19/2020 7:00 pm - 9:30 pm (2.5 hrs)

- Started reorganizing my plan since Day 1 did not go as planned.
- Setup MultiPart and GridFS data
- Wired up to Mongo
- Created Postman Collection for Testing.
- Started Testing
- Created Project in GitHub

9/21/2020 8:40 pm 11:10 pm (2.5 hrs)

- Finished the login screen
- Finished the register screen
- Stubbed out the navigation
- Tested navigation functionality
- Started Angular Services to interact with the NodeJS RESTFul api service.

9/22/2020 8:45 pm - 11:45 pm (3 hrs)

- Updated API with a login controller
- Updated POSTman collection for Login controller
- Created Angular Login Service
- Created Angular Current User Service
- Wired up Login Service to Page
- Wired up Top Navigation bar and Side bar to Current User
- Stubbed out Side Nav bar

9/23/2020 8:15 pm - 11:15 (3 hrs)

- Added new route for on API server for downloading files
- Setup Card and UI layout for files/folders
- Finalizing Card Styling
- Create Angular File Service

9/24/2020 8:45 pm - 11:45 pm (3 hours)

- Create Download Service
- Testing the Download functionality
- Fix Card Layout
- Fix Card to Data binding
- Fix Routing issue.

9/25/2020 8:30 pm - 10:30 (2.0 hours)

- Setup Logout
- Fixed back login feedback
- File Uploads with Folder creation
- Registration Complete
- Changes aspects of the File Dialog

9/26/2020 8:30 pm - 9:30 (1 hours)

- Share Files
- Setup Shared Files
- Setup Shared with Me Files.

TODO

- Upload feature
- Sharing feature
- Shared with me screen
- Shared screen.
- Implement side-bar links.
- Add link for email

Not Completed

- Image on top of the name in the sidebar. I have a placeholder for it but no logic. Put icon as placeholder
- Implement add folder logic. Since I am writing to a database, there is no concept of a “creating a folder” since the database stores a flat list of files. There is a way to do this by creating a hierarchy, but with the time limit, it did not want to go down that path since previous experiences with hierarchies can get messy. As a work around, for each file I included a path attribute/field to each file and built logic on the service to determine what files are part of certain folders/path. I disabled the add folder feature and incorporated a path in the file upload dialog based on your current context

- Did not address Browser history.
- Did not implement counts under folders.
- When files are a certain length they exceed the card limits.
- Did not implement the hidden text button
- The sharing system. I started some plumbing but could not complete it in the timeframe

Retrospective

- Sometimes you can't rely on the tools used. I had a challenge with the Chrome and Edge debugger in the client project. Even though I worked with this on a few projects prior and made debugging a lot simpler. For some reason, the ng server would start but the browser would not launch, until I stopped the ng server. After some investigation, I was able to attach to the chrome process to debug.
- I found myself working too long on the API. Since this was not a primary goal, I should have tried to find something that was already available instead of creating it.
- I ran into challenges with styling. Styling has always been a challenge for me. Angular Material at times can make this more difficult for overrides. After some discovery, I found using sass mixins vs local component or global styling(i.e. style.scss), is better wired with the Material framework
- Testing Driven may have helped me

References

- GFS - https://github.com/bradtraversy/mongo_file_uploads/blob/master/app.js
- GFS - <https://dev.to/shubhambattoo/uploading-files-to-mongodb-with-gridfs-and-multer-using-nodejs-5aed>
- Mat-icons - <https://jossef.github.io/material-design-icons-iconfont/>
- <https://www.angularjswiki.com/angular/angular-material-icons-list-mat-icon-list/#mat-icon-list-category-action>
- Uploading files - <https://www.ahmedboucheфра.com/angular-tutorial-example-upload-files-with-formdata-httpclient-rxjs-and-material-progressbar/>
- Dialog - <https://www.freakyjolly.com/angular-material-109-file-upload-ui-design-in-form-for-input-with-file-type-using-material-components/#.X26x5Y5UvUI>