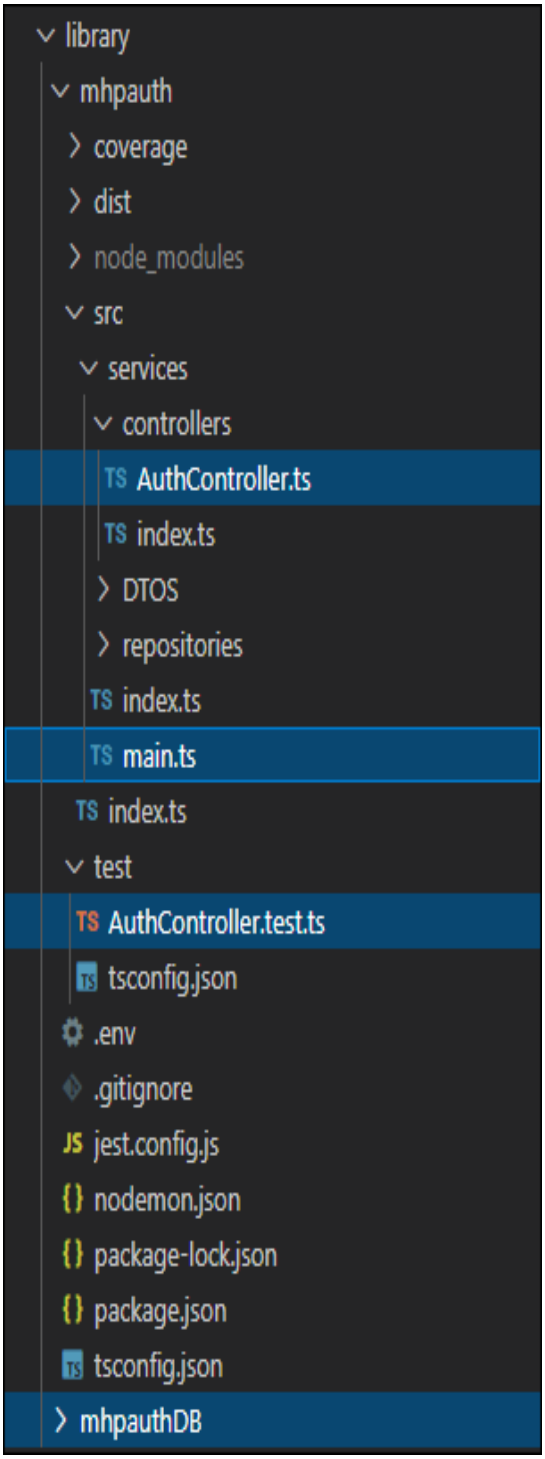
RUN!! MHPAUTH *Documentation* ***(Simplified)***

GitHub: https://github.com/DavidDexterCharles/MHPAUTH [please re-clone for the better changes]

The **library (**combination of **mhpauth and mhpauthDB)** is located at **\MHPAUTH\library**

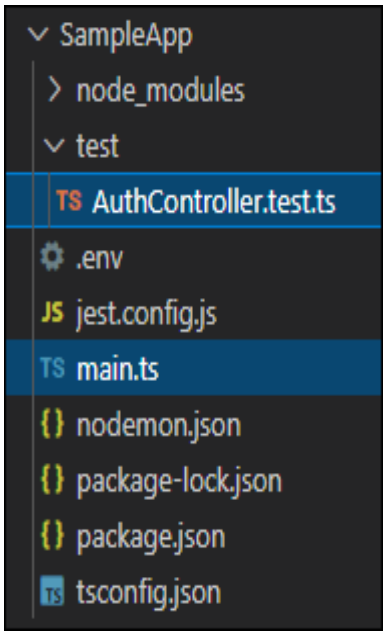


- **AuthController.ts** - as mentioned on the git repo has details of all the functions provided**. Note a new function added was password reset**. Fixes were made after I realized that the user was being notified as lock out after 4th attempt instead of at third attempt (this has been fixed and tests updated.)
- **main.ts** - what I forgot to mention was that running the command `**npm run test-env**` in this directory will also run main.ts. This file is equivalent to the file in SampleApp. The file has also been adjusted to include more organized code. In the order of register user, reset password, login, and authenticate. All the variables like `some_password` and `some_email` are at the top of the file and are used by the mentioned methods.
- **AuthController.test.ts**- again this file is also the same as the file in SampleApp\test. To run this file (called the test suite) use `**npm run test`.** Note all test have been rechecked and would have been adjusted to include `expect.assertions` which is important to use when dealing with async functions.

How to use the info above to test the library:

1. **AuthController.test.ts** executes all the main functionality offered by the library. Data is written and updated in test database each time this file is run due to the register and lockout test in the test suite. The only functionality not included in the suite is to check back after 20 minutes to see if a user has regained access. This can be achieved using main.ts.
2. **main.ts –** It's important to note that running this file using the command `**npm run test-env**` instead of `**npm start**` might be preferable for the main.ts file to be executed on the same db as Authcontroller.test.ts i.e. '**myhealthpasstest' .** In main.ts you can now just change the variable some_email to match an existing email in db, and all the passwords are the same i.e., "let some_password="SomePassword123";" Run main.ts on a locked account to try to unlock the user via login.

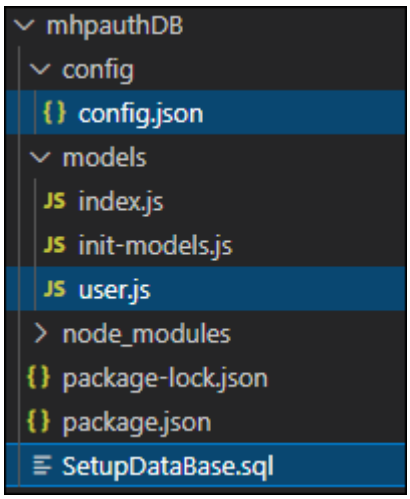
The example module '**SampleApp'** that uses the library is located at **MHPAUTH\SampleApp**



- To link the SampleApp to the library (if they are both in the same directory) the command ` **npm link ../library/mhpauth**` is used
- Here AuthController.test.ts and main.ts follow the same /similar instructions as they did in the library.
- The main difference is the remote import of the library into another app (SampleApp). For mhpauth `AuthController.test.ts and main.ts` were using the module directly
E.g: import {AuthController} from './index' . Whereas for SampleApp ` AuthController.test.ts and main.ts` they were using the module via npm link E.g. : import { AuthController } from "mhpauth"; .

- Commands to run : `**npm run test-env**` and `**npm run test**`

The **Database** connection **mhpauthDB**:



As was mentioned on GitHub docs the file config.json is at \MHPAUTH\library\mhpauthDB\config. The **\MHPAUTH\library** directory contains the ORM (object relational mapper) and the model itself (user.js). Its important to note that a developer can choose to switch to an entirely different database however note that sequelize db drivers would have to be installed . SetupDataBase.sql contains MySQL specific script for setting up the databases. This folder requires you to run` npm install`. Brief note the package.json file in all the mentioned directories are mostly the same.