

Before anything, you need to have the latest version of MongoDB, npm, and Nodejs installed. To build this project I used Modular Javascript, Nodejs, Express, JWT, MongoDB, React js, and redux.

For testing business logic (lib), I used Jest.

For Testing API, I used Jest and Supertest (to mock the HTTP request)

For Testing React (client), I used `@testing-library/jest-dom`, `@testing-library/react`.

For Test

You Should go to each folder in the “shuffle space” folder path, such as api, client and, lib. And then “npm test” to run the test cases.

Testings are using “on memory repository”, so they have nothing to do with the actual database.

For Deployment

Then do as follow to run the project for the first time.

- 1) In shuffle space folder path cd to api, and run “npm install” command. Do the same for client and lib as well.
- 2) Go to mongodb folder path and run “mongod --config mongo.conf” command. Which will run MongoDB on port 22000.
- 3) Run “mongoexport --port 22000 dump/shufflespace/.” to fill the db.
- 4) Go to api folder and run “npm start”
- 5) Then go to the client and run “npm start” while API is running. The localhost:3000 will serve your react service.
- 6) You should sign in first. We have to workers with the following credentials:
 - a) User: john@shufflespace.ca Password: 123
 - b) User: bob@shufflespace.ca Password: 12345
- 7) After logging in, you will find each customer and their projects. By clicking on the “scan barcode” button you will change the status of the project. If the logged-in worker was not working on the project right now the status of the project will be inactive. After scanning the barcode project status will change to “active” and by scanning the barcode again, the total amount of time in seconds that the worker was working on the project will be added to the “total duration” of the customer projects and status will be back to “inactive”.
- 8) You will be logged in unless you sign out by clicking on the “sign-out” button on the top right.

-- Hints:

- 1) You can use “In memory repository” just by changing the interface mode to inMemory, in `interfaceFactory('inMemory')` in `Server.js` file in api.

- 2) If you want to test the client only without api, there is a mock api called DevelopmentAPI in src/config in client. In order to use it, you should change the environment to 'development' then the apiFactory will plug DevelopmentApi.
I used this method to have easier testing (in test cases) and to know what should I send from api to client via use-cases and business logic before using real api.

I will send another file and describe the whole architecture and how it works.