TDD in JavaScript

Section 1: Tooling and Introduction to Mocha

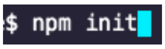
1.1 Visual Studio Code, NPM, Mocha and Chai

Visual Studio is a very powerful editor with an integrated terminal. Any editor can be used to write code such as Atom, Sublime, Notepadd++, etc. However, Visual Studio has grown in popularity with developers. You can install the VS Code editor by visiting Microsoft's VS Code website (https://code.visualstudio.com/). This is a free cross platform editor.

NPM is an acronym or Node Package Manager. This is a Package Manager for JavaScript which allows us to pull packages/libraries down into our npm based projects. When installing Node.js on our machines this installs the NPM as well. Node.js can be installed by visiting their website (https://nodejs.org/en/). You can search for npm packages/libraries by visiting https://www.npmjs.com/.

Mocha and Chai are two npm packages/libraries that we can install into our JavaScript (Node.js) projects using the npm commands in our terminal. We can learn more about these libraries from their website (https://mochajs.org/ and https://www.chaijs.com/).

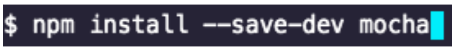
To create a new NPM/Node project we can run the following command in the terminal (ensuring we are within the project directory within the terminal i.e. using cd to navigate to the project path before running the following command):



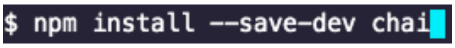
```
$ npm init
```

Note: We can use the -y flag to autofill all of the project setup questions.

Once the Node project has been created we should see a new package.json file which outlines the NPM project we are working with. We can now run the following commands to install the Mocha and Chai packages:

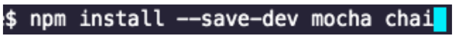


```
$ npm install --save-dev mocha
```



```
$ npm install --save-dev chai
```

Important Note: the --save-dev flag will save the packages as development dependencies and will not download the package when we deploy the live application. We can also run one command instead of two separate commands to install multiple packages at once as seen below. This will also apply the same flag parameters to both package installations.



```
$ npm install --save-dev mocha chai
```

The package.json file will now update adding the two packages as a devDependencies and a new node_modules directory will appear in the root of our project directory. This node_modules will contain the installed libraries along with their library dependencies.

We now have all of the tooling setup and can now see how to run some basic tests in mocha along with the chai assertion library.