Writing Unit Test in TypeScript

References: https://medium.com/@RupaniChirag/writing-unit-tests-in-typescript-d4719b8a0a40

https://buddy.works/guides/how-automate-nodejs-unit-tests-with-mocha-chai

1. Specify `Scripts` for `test` in `package.json`

   The `package.json` file is located under root of project and is generated when you execute `npm init`

2. Install Mocha/Chai (Unit Test Framework)

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

   Chai:

   THe best library to pair Mocha with would be Chai. It's a BDD/TDD library that works very well with this tool.

   Mocha:

   It's one of the simplest test suites for Node.js available, and allows for fairly accurate reporting, asynchronous tests, test coverage reports and, most importantly, can use any assertion library.

   nyc:

   Test code coverage

   --save-dev:

   Auto save to package.json

   We shall use Mocha as the test running framework, and Chai as the assertion library

3. a. `describe()` is merely for grouping, which you can nest as deep

   b. `it()` is a test case

4. Example From Website

   ```
   // Source code
   // Load express module with `require` directive
   var express = require('express')
   var app = express()

   //Define request response in root URL (/)
   ```

```
app.get('/', function (req, res) {
  res.send('Hello World')
})

//Launch listening server on port 8080
app.listen(8080, function () {
  console.log('App listening on port 8080!')
})
```

Test Code

```
// Test Code
// Use it
var expect  = require('chai').expect;
var request = require('request');

it('Main page content', function(done) {
    request('http://localhost:8080' , function(error, response, body) {
        expect(body).to.equal('Hello World');
        done();
    });
});
```

Results:

```
Bartek-Macbook: ~/myapp/test $ npm test

> hello-world@1.0.0 test /Users/Bartek-Macbook/myapp
> mocha



  ✓ Main page content
  ✓ Main page status
  ✓ About page content

  3 passing (39ms)

Bartek-Macbook: ~/myapp/test $
```

Group Tests:

```
// Group Test
var expect  = require('chai').expect;
var request = require('request');

describe('Status and content', function() {
    describe ('Main page', function() {
        it('status', function(done){
```

```
            request('http://localhost:8080/', function(error, response,
body) {
                expect(response.statusCode).to.equal(200);
                done();
            });
        });

        it('content', function(done) {
            request('http://localhost:8080/' , function(error, response,
body) {
                expect(body).to.equal('Hello World');
                done();
            });
        });
    });

    describe ('About page', function() {
        it('status', function(done){
            request('http://localhost:8080/about', function(error,
response, body) {
                expect(response.statusCode).to.equal(404);
                done();
            });
        });

    });
});
```

Test Result:

```
Bartek-Macbook: ~/myapp/test $ npm test

> hello-world@1.0.0 test /Users/Bartek-Macbook/myapp
> mocha



  Status and content
    Main page
      ✓ status
      ✓ content
    About page
      ✓ status


  3 passing (44ms)

Bartek-Macbook: ~/myapp/test $
```