

Exercises for testing

This lecture is about testing your code, we will write tests for our code.

Setting up our project

Before we do this you will need to setup your environment.

1. Check that you have node installed in your environment

You can do this by running: `node -v`

If you get a response like the following you are ok you already have node.

```
2018-06-05 10:53:12 [bts-MacBook-Pro-4]
± |master U:3 ? :2 x| → node -v
v8.10.0
```

Otherwise you will need to download node and install it.

You can use this link <https://nodejs.org/en/>

Once you do this you are ready to go with our exercise

For each exercise create three different folders each time.

- Exercise-Mocha
- Exercise-Chai
- Exercise-Sinon

The structure of each folder will be the same for all.

- A **test folder** with a **tests.js** file
- And an **exercise.js** file

```
└─ exercise-X
  └─ test
    JS tests.js
    JS exercise.js
```

Navigate with command line to that folder

Type on the command prompt: **npm init**

You will need to type the following (3 times for each one of the exercises)

```
± |master U:3 ?:3 x| → npm init
This utility will walk you through creating a package.json file.
It only covers the most common items, and tries to guess sensible defaults.

See `npm help json` for definitive documentation on these fields
and exactly what they do.

Use `npm install <pkg>` afterwards to install a package and
save it as a dependency in the package.json file.

Press ^C at any time to quit.
package name: (exercise-mocha) exercise-mocha
version: (1.0.0) 0.0.1
description: my first test
entry point: (exercise.js) exercise.js
test command: mocha
git repository: none
keywords: none
author: Your name
license: (ISC) ISC
About to write to /Users/arismarko/Documents/projects/lecturecode/Lecture 5+/exercise-mocha/package.json:

{
  "name": "exercise-mocha",
  "version": "0.0.1",
  "description": "my first test",
  "main": "exercise.js",
  "directories": {
    "test": "test"
  },
  "scripts": {
    "test": "mocha"
  },
  "repository": {
    "type": "git",
    "url": "none"
  },
  "keywords": [
    "none"
  ],
  "author": "Your name",
  "license": "ISC"
}

Is this ok? (yes) █
```

Ln 1, Col 1

Press Enter that will create a **package.json** file.

Next we need to install mocha we will have to do the same for all three exercises

To do this run on the command line **npm install mocha -- save-dev**

(this command will add it to the development dependencies on your package.json)

Exercise 1: Mocha

For our first exercise we will use the Exercise.js The task is to build a function that we will take 2 inputs:

```
Function bmi(height, weight) {  
    Var bmi =  
    return bmi;  
}
```

The formula is:

$(\text{weight (kilograms)} / \text{height (metres)}) / \text{height (kilograms)}$

Our test should check for a person with the following characteristics

Weight: 90 kilos

Height: 1.90 metr

The formula should return **24.9**

What we want to check if we pass 90 as weight and the 1.90 as height the result will be 24.9.

Write a mocha test that will check for this

Step 1:

Inside the test.js add the following code:

```
var assert = require('assert');  
var BMI = require('../exercise.js');  
  
describe('BMI tests', function(){  
    it('should return 24.9 if height is 1.90 and weight is 90', function(){  
        assert.equal(BMI(1.90,90), 24.9);  
    })  
})
```

Step 2:

Inside the exercise.js write the following code

```
module.exports = BMI;

function BMI(height, weight){
  return (weight/height)/height;
}
```

Step 3:

On your console run **npm run test**

What is the result of that? Did the test pass?

Exercise 2: Chai

For this exercise we will have to install chai

To do this in our command prompt (console) we will write the following command.

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

Step 1:

;Copy the code from the previous exercise to exercise.js

```

module.exports = BMI;

function BMI(height, weight){
  return (weight/height)/height;
}

```

Step 2:

On your test will need to re-write it with chai

m

```

var chai = require('chai');
var should = require('chai').should();
var expect = chai.expect;
var BMI = require('../exercise.js');

describe('BMI Tests', function(){
  it('should return 24.9 if the weight is 90, and the height 1.90', function(){
    var height = 1.90;
    var weight = 90;

    var expected = 24.9;
    var actual = BMI(height, weight);

    expect(actual).to.equal(expected);
  })
})

```

Step 3:

Run your test **npm run test**

What was the outcome?

Exercise 3: Sinon

For this exercise we will try the mock example of the Sinon

Before that we will need to create our folder and then install Sinon.

npm install sinon --save-dev

Step 1:

We want to know more this time so as we can see how many times the `getData` method has been called on this `userAccountsService`

Rftrtg “

FASv

Step 2:

Now we need to write our code that will get the

```
var chai = require('chai');
var should = chai.should();
var sinon = require('sinon');
var assert = require('assert');

var UserAccountsService = require('../exercise.js');

it('Get users data', function(){
  var email1 = "aris@city.ac.uk";
  var email2 = "aristos.markogiannakis.1@city.ac.uk";

  var accountService = new UserAccountsService();

  var mockAccountService = sinon.mock(accountService);

  mockAccountService.expects('getData').exactly(3);

  accountService.fetchData();

  mockAccountService.verify();
});
```

Step 3:

Execute your tests.