

COMP3006 – Full-Stack Development

Workshop 11 – Model Answer

Autumn 2020

This set of model answers are intended to show possible ways of building unit tests. I am not providing a model answer for the mini project or extension activities. Please attempt the exercises before looking at the answers.

1. A full command and code listing is given in the worksheet.
2. A full command and code listing is given in the worksheet.
3. A full command and code listing is given in the worksheet. For reference, here is an example [package.json](#) to show you what you should be aiming to produce:

```
{
  "name": "calcapp",
  "version": "1.0.0",
  "description": "",
  "main": "server.js",
  "directories": {
    "test": "test"
  },
  "scripts": {
    "test": "mocha -ui tdd test/",
    "start": "node server.js"
  },
  "repository": {
    "type": "git",
    "url": "git+https://github.com/djw213/CalcApp.git"
  },
  "author": "David Walker",
  "license": "ISC",
  "bugs": {
    "url": "https://github.com/djw213/CalcApp/issues"
  },
  "homepage": "https://github.com/djw213/CalcApp#readme"
}
```

4. The task here is to add to the project, so that when you commit it you can see Travis build and test the new code. Begin by editing the [logic.js](#) code to add the following:

```
function addNumbers(a, b) {
  /**
   * Add the numbers a and b together and return the result.
   */
  return a + b;
}

module.exports.addNumbers = addNumbers;
```

Then add a unit test that will exercise the new code:

```

suite("Test addition function", function() {

  test("Test addition works correctly", function() {
    chai.assert.equal(17, logic.addNumbers(5, 12), "5+12 should = 17");
    chai.assert.equal(26, logic.addNumbers(17, 9), "17+9 should = 26");
  });

});

```

Run **git commit** and **git push** as above to commit the changes and push to the repo, and your Travis build should start. Your tests should pass.

5. Here you need to expose the **addNumbers** function using a new route. Add the following to the **routes.js** file:

```

function addNumbersRoute(request, response) {
  let number1 = Number(request.params.number1);
  let number2 = Number(request.params.number2);
  response.send(logic.addNumbers(number1, number2).toString());
}

module.exports.addNumbersRoute = addNumbersRoute;

```

It's important to convert the inputs to numbers and the outputs to strings, or you will concatenate the inputs as strings (e.g., **"12" + "34" = "1234"**). Sending the output as a number will cause Express to treat it as a status code rather than the output message.

Add the following integration test to ensure that the new route operates correctly:

```

suite("Integration tests for add function", function() {

  test("Test GET /add", function() {
    let app = server.app;

    chai.request(app).get("/add/5/12").end(function(error, response) {
      chai.assert.equal(response.status, 200, "Wrong status code");
      chai.assert.equal("17", response.text, "Wrong response text");
    });
  });

});

```

Again, run **git commit** and **git push** as above to commit the changes and push to the repo, and your Travis build should start. Your tests should pass.