

Assignment



Testing Assignment

Submitted by

Chayut Piyaosotsun 6388098

Kanapon Suthiwongkorn 6388107

Present To

Dr. Wudhichart Sawangphol

ITCS371 Intro to Software Engineering, Semester 1, Section 3

Task 1: Unit Testing

Test case ID: T_01 **Test Designed by:** Chayut Piyaosotsun
Test Priority: Low **Test Designed date:** 14/11/2022
Module Name: StudentServiceRoutes **Test Executed by:** Kanapon Suthiwongkorn
Test Title: Test: GET / **Test Execution date:** 15/11/2022
Description: Test the StudentServiceRoutes.js via path /.
Pre-condition: Server localhost:3000 is running.

Dependencies;

Step	Test step	Test data	Expected Result	Actual Result	Status
1	Get router from "./routes/studentServiceRoutes"				
2	Send / path to router path via get method.	/	{ error: true, message: "hello" }	{ error: true, message: "hello" }	Pass

Post-condition: Server does not give any error and send the correct response.

Testing output:

```
> itcs371-testing-lab@1.0.0 test
> jest --forceExit "studentServiceRoutesUnitTestGET.test.js"

PASS __tests__/studentServiceRoutesUnitTestGET.test.js
  Unit Testing: Test the root
    ✓ Test: GET / (76 ms)

Test Suites: 1 passed, 1 total
Tests:      1 passed, 1 total
Snapshots: 0 total
Time:       1.065 s
Ran all test suites matching /studentServiceRoutesUnitTestGET.test.js/i.
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?
```

Code to test:

```
const express = require("express"),
  app = express(),
  router = require("../routes/studentServiceRoutes"),
  request = require("supertest");

app.use("/", router);

describe("Unit Testing: Test the root", () => {
  test("Test: GET /", async () => {
    const res = await request(app).get("/");
    expect(res.body).toEqual({ error: true, message: "hello" });
  });
});
```

Test case ID: T_02 **Test Designed by:** Chayut Piyaosotsun
Test Priority: Low **Test Designed date:** 14/11/2022
Module Name: StudentServiceRoutes **Test Executed by:** Kanapon Suthiwongkorn
Test Title: Test: GET /students **Test Execution date:** 15/11/2022
Description: Test the StudentServiceRoutes.js via path /students.
Pre-condition: Server localhost:3000 is running.

Dependencies;

Step	Test step	Test data	Expected Result	Actual Result	Status
1	Get router from "./routes/studentServiceRoutes"				
2	Send /students path to router path via get method.	/students	Data of all student	All student data	Pass

Post-condition: Server does not give any error and sends the correct response. Response comes in the expected form as the data store in res.body.data.

Testing output:

```
> itcs371-testing-lab@1.0.0 test
> jest --forceExit "studentServiceRoutesUnitTestGETstudents.test.js"

PASS __tests__/studentServiceRoutesUnitTestGETstudents.test.js
  Unit Testing: Test the root
    ✓ Test: GET /students (140 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        1.157 s
Ran all test suites matching /studentServiceRoutesUnitTestGETstudents.test.js/i.
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?
```

Code to test:

```
const express = require("express"),
  app = express(),
  router = require("../routes/studentServiceRoutes"),
  request = require("supertest");

app.use("/", router);

describe("Unit Testing: Test the root", () => {
  test("Test: GET /students", async () => {
    const res = await request(app).get("/students");
    expect(res.body.data).toEqual([
      {
        STU_ID: 1,
        STU_FNAME: "Andrew",
        STU_LNAME: "Black",
        STU_AGE: 25,
      },
      {
        STU_ID: 2,
        STU_FNAME: "Alexandra",
        STU_LNAME: "Brown",
        STU_AGE: 25,
      },
      {
        STU_ID: 3,
        STU_FNAME: "Amanda",
        STU_LNAME: "Davidson",
        STU_AGE: 25,
      },
      {
        STU_ID: 4,
        STU_FNAME: "Benjamin",
        STU_LNAME: "Duncan",
        STU_AGE: 25,
      },
      {
        STU_ID: 5,
        STU_FNAME: "Christopher",
        STU_LNAME: "Ellison",
        STU_AGE: 25,
      },
    ]);
  });
});
```

Test case ID: T_03 **Test Designed by:** Kanapon Suthiwongkorn
Test Priority: Low **Test Designed date:** 14/11/2022
Module Name: clientRoutes **Test Executed by:** Chayut Piyaosotsun
Test Title: Test: GET / **Test Execution date:** 15/11/2022
Description: Test the clientRoutes.js via path /.
Pre-condition: Server localhost:3100 is running.

Dependencies;

Step	Test step	Test data	Expected Result	Actual Result	Status
1	Get router from "./routes/clientRoutes"				
2	Send / path to router path via get method.	/	HTML file from ‘__dirname, "./views/studentForm. html")’ in utf8 format.	studentForm.html in utf8 format.	Pass

Post-condition: Server does not give any error and sends the correct response. Response comes in the expected form as the data store in res.text.

Testing output:

```
> itcs371-testing-lab@1.0.0 test
> jest --forceExit "clientRoutesUnitTestGET.test.js"

PASS __tests__/clientRoutesUnitTestGET.test.js
  Unit Testing: Test the root
    ✓ Test: GET / (59 ms)

Test Suites: 1 passed, 1 total
Tests: 1 passed, 1 total
Snapshots: 0 total
Time: 1.058 s
Ran all test suites matching /clientRoutesUnitTestGET.test.js/i.
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?
```

Code to test:

```
const express = require("express"),
  app = express(),
  router = require("../routes/clientRoutes"),
  request = require("supertest"),
  fs = require("fs"),
  path = require("path");

app.use("/", router);

describe("Unit Testing: Test the root", () => {
  test("Test: GET /", async () => {
    const res = await request(app).get("/");
    expect(res.text).toEqual(
      fs.readFileSync(path.join(__dirname, "../views/studentForm.html"), "utf8")
    );
  });
});
```

Test case ID: T_04 **Test Designed by:** Kanapon Suthiwongkorn
Test Priority: Low **Test Designed date:** 14/11/2022
Module Name: clientRoutes **Test Executed by:** Chayut Piyaosotsun
Test Title: Test: GET /js/callStudentService.js
Test Execution date: 15/11/2022
Description: Test the clientRoutes.js via path /js/callStudentService.js.
Pre-condition: Server localhost:3100 is running.

Dependencies;

Step	Test step	Test data	Expected Result	Actual Result	Status
1	Get router from "./routes/clientRoutes"				
2	Send /js/callStudentService.js path to router path via get method.	/	js file from ‘__dirname, "./js/callStudentService.js")’ in utf8 format.	StudentService.js in utf8 format.	Pass

Post-condition: Server does not give any error and sends the correct response. Response comes in the expected form as the data store in res.text.

Testing output:

```
> itcs371-testing-lab@1.0.0 test
> jest --forceExit "clientRoutesUnitTestGETjsfile.test.js"

PASS  _tests_/clientRoutesUnitTestGETjsfile.test.js
Unit Testing: Test the root
  ✓ Test: GET /js/callStudentService.js (58 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        0.944 s
Ran all test suites matching /clientRoutesUnitTestGETjsfile.test.js/i.
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?
```


Code to test:

```
const express = require("express"),
  app = express(),
  router = require("../routes/clientRoutes"),
  request = require("supertest"),
  fs = require("fs"),
  path = require("path");

app.use("/", router);

describe("Unit Testing: Test the root", () => {
  test("Test: GET /js/callStudentService.js", async () => {
    const res = await request(app).get("/js/callStudentService.js");
    expect(res.text).toEqual(
      fs.readFileSync(
        path.join(__dirname, "../js/callStudentService.js"),
        "utf8"
      )
    );
  });
});
```

Task 2: Integration Testing

Test case ID: T_05 **Test Designed by:** Chayut Piyaosotsun
Test Priority: Medium **Test Designed date:** 14/11/2022
Module Name: StudentServiceRoutes **Test Executed by:** Kanapon Suthiwongkorn
Test Title: Integration Testing: Getting information of the first student in database
with/students and /student/:id

Test Execution date: 15/11/2022

Description: Comparing information of the first student from /students and /student/:id.

Pre-condition: Server localhost:3000 is running. The first /student is not an error.

Dependencies;

Step	Test step	Test data	Expected Result	Actual Result	Status
1	Get router from "./routes/student ServiceRoutes"				
2	Send /students path to router path via get method.	/students	Data of all student	All student data	Pass
3	Store first data of response				
4	Send /student/:id path to router path via get method to get data from id.	/student/0	Data of first student.	Data of first student.	Pass

Post-condition: Server does not give any error and sends the correct response. Response comes in the expected form as the data store in res.body.data.

Testing output:

```
> itcs371-testing-lab@1.0.0 test
> jest --forceExit "studentServiceRoutesIntegrationTest.test.js"

PASS  __tests__/_studentServiceRoutesIntegrationTest.test.js
  Integration Testing: Getting information of the first student in database with /students and /student/:id
    ✓ Test: GET all students via /students (80 ms)
    ✓ Test: Get the information of the first student via /student/:id (18 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.013 s
Ran all test suites matching /studentServiceRoutesIntegrationTest.test.js/i.
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?
```

Code to test:

```
const express = require("express"),
    app = express(),
    router = require("../routes/studentServiceRoutes"),
    request = require("supertest");

app.use("/", router);

describe("Integration Testing: Getting information of the first student in database with /students and /student/:id", () => {
    let firstStudent;
    test("Test: GET all students via /students", async () => {
        const res = await request(app).get("/students");
        expect(res.body.data).toEqual([
            {
                STU_ID: 1,
                STU_FNAME: "Andrew",
                STU_LNAME: "Black",
                STU_AGE: 25,
            },
            {
                STU_ID: 2,
                STU_FNAME: "Alexandra",
                STU_LNAME: "Brown",
                STU_AGE: 25,
            },
            {
                STU_ID: 3,
                STU_FNAME: "Amanda",
                STU_LNAME: "Davidson",
                STU_AGE: 25,
            },
            {
                STU_ID: 4,
                STU_FNAME: "Benjamin",
                STU_LNAME: "Duncan",
                STU_AGE: 25,
            },
            {
                STU_ID: 5,
                STU_FNAME: "Christopher",
                STU_LNAME: "Ellison",
                STU_AGE: 25,
            },
        ]);
        firstStudent = res.body.data[0];
    });
    test("Test: Get the information of the first student via /student/:id", async () => {
        const res = await request(app).get(`/student/${firstStudent.STU_ID}`);
        expect(res.body.data).toEqual(firstStudent);
    });
});
```

Test case ID: T_06 **Test Designed by:** Kanapon Suthiwongkorn

Priority: Medium **Test Designed date:** 15/11/2022

Module Name: clientRoutes **Test Executed by:** Chayut Piyaosotsun

Test Title: Integration Testing: Get the src in HTML file via / to compare with callStudentService.js file via /js/callStudentService.js

Test Execution date: 16/11/2022

Description: Comparing information of the src in HTML file from / to compare with callStudentService.js file from /js/callStudentService.js.

Pre-condition: Server localhost:3100 is running. Files are located in the correct directory; "../views/studentForm.html" and "/js/callStudentService.js".

Dependencies;

Step	Test step	Test data	Expected Result	Actual Result	Status
1	Get router from "../routes/clientRoutes"				
2	Send / path to router path via get method.	/	HTML file	HTML file	Pass
3	Store only the part of js file from res.text of the response				
4	Send /js/callStudentService.js path to router path via get method to get data from id.	/js/callStudentService.js	Js file	Js file	Pass

Post-condition: Server does not give any error and sends the correct response. Response comes in the expected form as the data store in res.test.

Testing output:

```
> itcs371-testing-lab@1.0.0 test
> jest --forceExit "clientRoutesIntegrationTest.test.js"

PASS __tests__/clientRoutesIntegrationTest.test.js
  Integration Testing: Get the src in HTML file via / to compare with callStudentService.js file via /js/callStudentService.js
    ✓ Test: GET / (34 ms)
    ✓ Test: GET /js/callStudentService.js (6 ms)

Test Suites: 1 passed, 1 total
Tests:       2 passed, 2 total
Snapshots:   0 total
Time:        1.095 s
Ran all test suites matching /clientRoutesIntegrationTest.test.js/i.
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?
```

Code to test:

```
const express = require("express"),
  app = express(),
  router = require("../routes/clientRoutes"),
  request = require("supertest"),
  fs = require("fs"),
  path = require("path");

app.use("/", router);

describe("Integration Testing: Get the src in HTML file via / to compare with
callStudentService.js file via /js/callStudentService.js", () => {
  let src = "";
  test("Test: GET /", async () => {
    const res = await request(app).get("/");
    expect(res.text).toEqual(
      fs.readFileSync(path.join(__dirname, "../views/studentForm.html"), "utf8")
    );
    script = res.text;
    for (let i = 3554; i <= 3577; i++) {
      src += script[i];
    }
  });
  test("Test: GET /js/callStudentService.js", async () => {
    const res = await request(app).get("/js/callStudentService.js");
    expect(res.text).toEqual(
      fs.readFileSync(path.join(__dirname, "../", src), "utf8")
    );
  });
});
```

Task3: System Testing

Test case ID: T_07 **Test Designed by:** Kanapon Suthiwongkorn

Priority: High **Test Designed date:** 16/11/2022

Module Name: clientRoutes **Test Executed by:** Chayut Piyaosotsun

Test Title: Test: Getting the information of students through the user interface.

Test Execution date: 17/11/2022

Description: Test the select button in the web page.

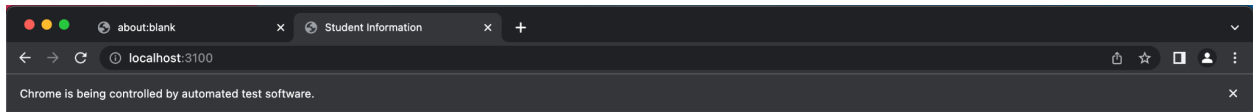
Pre-condition: Server localhost:3000 and localhost:3100 are running.

Dependencies;

Step	Test step	Test data	Expected Result	Actual Result	Status
1	Lunch browser				
2	Open new page				
3	Navigate to http://localhost:3100				
4	Click to input the STU_ID textbox.				
5	Enter the number “2” in the textbox.	2			
6	Click the select button.				
7	Wait for the response from the website.		firstName: "Alexandra", lastName: "Brown", age: "25"	firstName: "Alexandra", lastName: "Brown", age: "25"	Pass

Post-condition: Server does not give any error and sends the correct response. Response comes in the expected form as an object.

Testing output:



```
> itcs371-testing-lab@1.0.0 test
> jest --forceExit "System.test.js"

PASS  __tests__/System.test.js (7.884 s)
  ✓ Test: Getting the information of student through the user interface. (6707 ms)

Test Suites: 1 passed, 1 total
Tests:       1 passed, 1 total
Snapshots:   0 total
Time:        7.963 s, estimated 13 s
Ran all test suites matching /System.test.js/i.
Force exiting Jest: Have you considered using `--detectOpenHandles` to detect async operations that kept running after all tests finished?
```

Code to test:

```
const express = require("express"),
  app = express(),
  router = require("../routes/studentServiceRoutes"),
  puppeteer = require("puppeteer");

app.use("/", router);

test("Test: Getting the information of student through the user interface.", async () => {
  const browser = await puppeteer.launch({
    headless: false,
    slowMo: 80,
    args: ["--window-size=1920,1080"],
    executablePath:
      "/Applications/Google Chrome.app/Contents/MacOS/Google Chrome",
  });

  const page = await browser.newPage();
  await page.goto("http://localhost:3100/");

  await page.click("input#STU_ID");
  await page.type("input#STU_ID", "2");

  page.on("dialog", async (dialog) => {
    await dialog.accept();
  });
  await page.click("input#select");

  const studentObject = await page.evaluate(() => {
    return {
      firstName: document.getElementById("STU_FNAME").value,
      lastName: document.getElementById("STU_LNAME").value,
      age: document.getElementById("STU_AGE").value,
    };
  });
  expect(studentObject).toEqual({
    firstName: "Alexandra",
    lastName: "Brown",
    age: "25",
  });
}, 20000);
```