

Lab 4.3 Cucumber with JavaScript

This section will guide you to:

- Write code in Gherkin and execute via cucumber with JavaScript
- Use visual studio code (the popular text editor for web-based projects)
- Confirm the output of cucumber with the positive result.

Note: You cannot proceed with Lab 4.3 until you complete all the steps mentioned in the document Lab 4.2 completely.

This lab has 3 subsections:

- 4.3.1 See scenario reported as undefined.
- 4.3.2 See scenario reported as failing.
- 4.3.3 See scenario reported as passing.

In behavior-driven development with cucumber, concrete examples are used to specify “what we want the software to do”. An example is referred as a scenario. Scenarios are defined in “.feature” files which are stored in the “features” directory.

In this lab, we will use the scenario: **Sunday is not Friday**.

Create an empty file named “**features/is_it_Friday_yet.feature**” with the following code, and save the file.

```
Feature: Is it Friday yet?  
  Everybody wants to know when it's Friday  
  
  Scenario: Sunday isn't Friday  
    Given today is Sunday  
    When I ask whether it's Friday yet  
    Then I should be told "Nope"
```

Step 4.3.1: See scenario reported as undefined.

Run the command **npm test** in the terminal of visual studio code to confirm the following output:

```
1 Scenario (1 undefined)  
3 steps (3 undefined)  
0m00.000s
```

Step 4.3.2: See scenario reported as failing.

Replace the existing complete code with the following code in the **features/step_definitions/stepdefs.js** file.

```
const assert = require('assert');
const { Given, When, Then } = require('cucumber');

function isItFriday(today) {
  // We'll leave the implementation blank for now
}

Given('today is Sunday', function () {
  this.today = 'Sunday';
});

When('I ask whether it\'s Friday yet', function () {
  this.actualAnswer = isItFriday(this.today);
});

Then('I should be told {string}', function (expectedAnswer) {
  assert.equal(this.actualAnswer, expectedAnswer);
});
```

Run the command **npm test** in the terminal of visual studio code to confirm the following output:

```
1 scenario (1 failed)
3 steps (1 failed, 2 passed)
0m00.004s
```

Step 4.3.3: See scenario reported as passing

Add **return 'Nope';** to the code available in the **features/step_definitions/stepdefs.js** file.

```
function isItFriday(today) {
  return 'Nope';
}
```

Run the command **npm test** in the terminal of visual studio code to confirm the following output:

```
1 scenario (1 passed)
3 steps (3 passed)
0m00.002s
```

Congratulations! You've got your first green cucumber scenario.