

CucumberJS + Selenium Web Driver

Create a CucumberJS project

```
# install Cucumber and Selenium webdriver
$ npm install cucumber selenium-webdriver

# install browser drivers
$ npm install chromedriver geckodriver

# install an assertion tool
$ npm install assert
```

If you are using a package.json file you will probably want to save all of these modules to dev dependencies. to do that when running npm install add a switch `--save-dev` to the command.

Another nice thing to do with a package.json file is to add test script.

```
{
  "scripts": {
    "test": "./node_modules/.bin/cucumber-js"
  }
}
```

With this added to your NPM package file you can run all your cucumber tests by typing in "npm test" on the command line.

Folder Structure

First off we need to create a place for all of our feature files written in Gherkin. Within this folder you will create files with an extension of feature.

For example "testFile.feature".

```
mkdir features
```

After creating the features directory we need a place to store our code that runs the features and does the assertions. We call these step definitions.

```
cd features
mkdir step_definitions
```

Last but not least we need to create a support directory that handles initializing the World object for Cucumber and other enhancements for Cucumber.

```
mkdir support
```

Once done you should have a folder structure like this.

```
- features
- step_definitions
- support
```

Create the Cucumber World.

Within the support directory create a file named "world.js". In this file we can require some libraries, select browser to test with and set some timeouts.

```
var seleniumWebdriver = require('selenium-webdriver');

function CustomWorld() {
  this.driver = new seleniumWebdriver.Builder()
    .forBrowser('firefox')
    .build();
}

module.exports = function() {
```

```
this.World = CustomWorld;  
  
// sets a default timeout to 30 seconds. Time i  
this.setDefaultTimeout(30 * 1000);  
};
```

Describe your behavior.

Create a feature file like "test_cookbook.feature". Within the file describe the behavior that you would like.

Feature: Test Cookbook Home Page

Scenario: Test Cookbook Title Description

Given I visit Test Cookbook website

Then I see title Test Cookbook

Define your step.

Create a file called "test_cookbook.js" and we can start building our step definitions.

Within the file we first need to setup a javascript module.

```
module.exports = function () {  
  
};
```

When you run the tests Cucumber will be looking for regular expression matching

functions to execute. In the feature file the first step we have is a Given statement.

```
module.exports = function () {  
  this.Given(/^I visit Test Cookbook website$/, fu
```

```
    });  
};
```

Now that we have an expression to run, we also need to give it some direction. Lets have it take us to “www.testcookbook.com”.

```
module.exports = function () {  
  this.Given(/^I visit Test Cookbook website$/, function() {  
    return this.driver.get('http://www.testcookbook.com');  
  });  
};
```

If you were to run the tests at this point you should see a browser pop up and goto “www.testcookbook.com”.

At this point you may still see a browser on the screen. While starting out its not a big deal to close the window by hand. However we can add a hooks.js file within the step_definitions folder to handle that for us.

```
module.exports = function() {  
  this.After(function() {  
    return this.driver.quit();  
  });  
};
```

We now have some code in place to take us to a website. But thats not really testing anything. If we look back at the feature file we have a validation step that says the title of the page should be “Test Cookbook”. To accomplish this we need to get the text for the title and assert whether or not it equals the desired outcome.

```
module.exports = function () {  
  this.Given(/^I visit Test Cookbook website$/, function() {  
    return this.driver.get('http://www.testcookbook.com');  
  });  
};
```

```
this.Then(/^I see title Test Cookbook$/, function() {
  this.driver.getTitle().then(function (title) {
    assert.equal(title, "Test Cookbook");
    return title;
  });
});
};
```

Now give it a run.

UH OH!! It didn't work like expected did it? You probably see something that says "ReferenceError: assert is not defined". CucumberJS needs an extra library in order to assert or expect specific values. Luckily this is an easy fix. Provided that you installed the assert package at the beginning all we need to do is require assert.

```
var assert = require('assert');

module.exports = function () {
  this.Given(/^I visit Test Cookbook website$/, function() {
    return this.driver.get('http://www.testcookbook.com');
  });

  this.Then(/^I see title Test Cookbook$/, function() {
    this.driver.getTitle().then(function (title) {
      assert.equal(title, "Test Cookbook");
      return title;
    });
  });
};
```

If you run `npm test` this time you should be able to get some passing tests.

Please note if you are working with Firefox that newer versions will require `geckodriver`. If you see firefox open up and never does anything you are more than likely missing that module.

Would you like to pull down a working project? Look on GitHub.

<https://github.com/testcookbook/js-cucumber-selenium>