2.1 Writing a Simple Test

This section will guide you to:

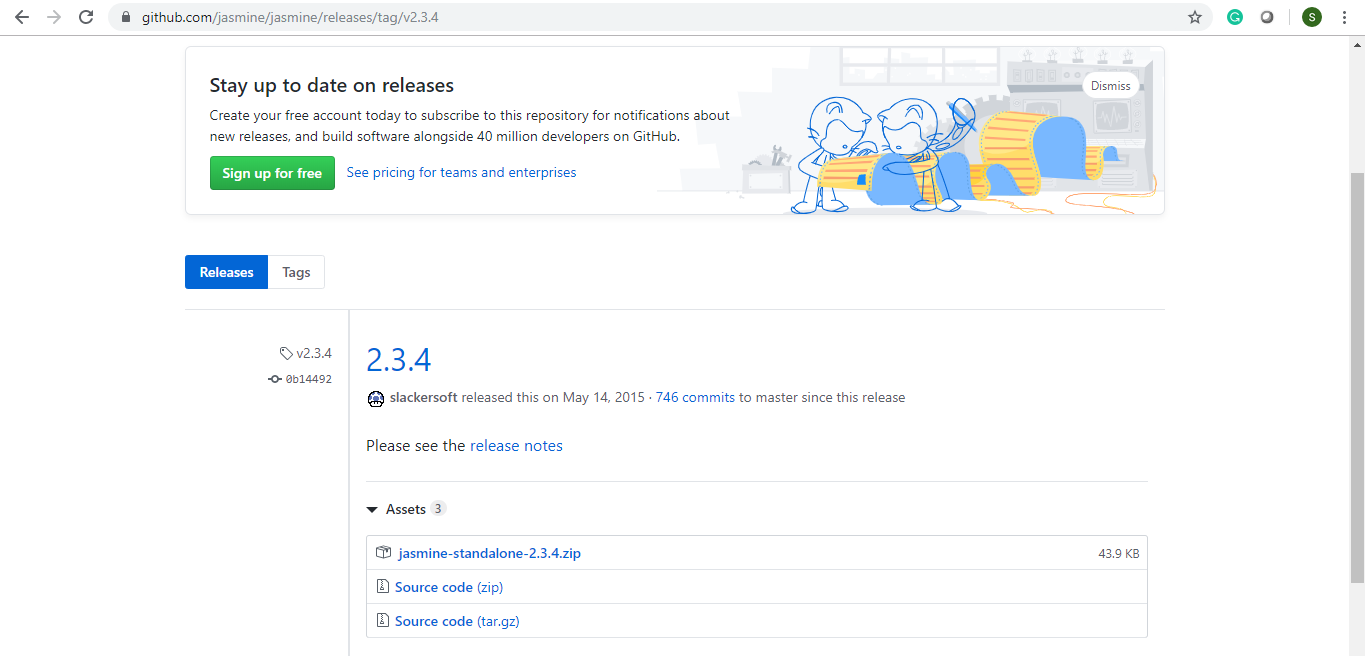
* Write simple tests

This lab has seven subsections, namely:

* + 1. Downloading Jasmine
    2. Writing a test using Jasmine
    3. Running the test and watching it fail
    4. Writing code to pass the tests
    5. Refactoring the code
    6. Using Backbone.js in Jasmine Tests
    7. Pushing the code to GitHub repositories
* *Jasmine 3.5.0 is already installed in your lab. (Refer MEAN: Lab Guide - Phase 4)*

**Step 2.1.1:** Downloading Jasmine

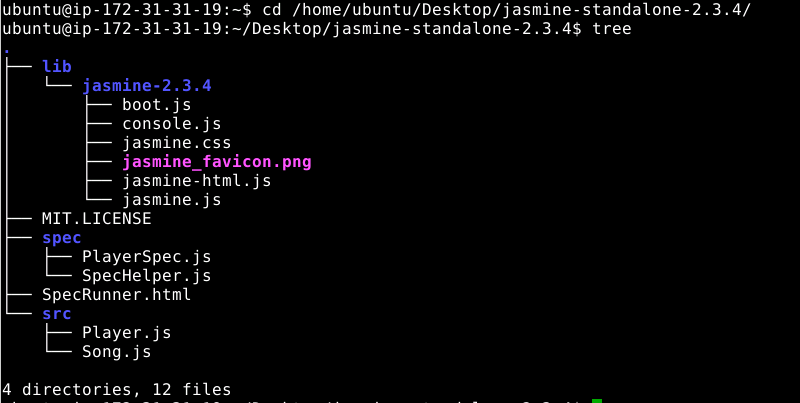
* Go to <https://github.com/jasmine/jasmine>
* Click on releases
* Select the 2.3.4 release
* Click on jasmine-standalone-2.3.4.zip



* Unzip the downloaded folder at ***/home/ubuntu/Desktop/jasmine-standalone-2.3.4/*** location
* Open command prompt
* Go to the directory where the unzipped folder is present using ***cd***

***/home/ubuntu/Desktop/jasmine-standalone-2.3.4/*** command

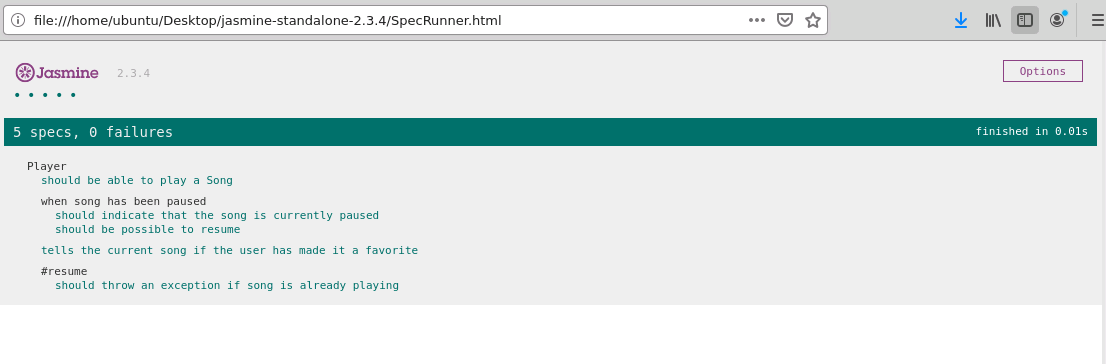
* Type **tree**



* The **spec** and **src** directories consist of some sample codes and tests. This helps you get an idea on how to use Jasmine
* The sample code is in the src directory and the sample tests are in spec directory.
* Open your default browser and type the path where the SpecRunner.html file is present

*/home/ubuntu/Desktop/jasmine-standalone-2.3.4/SpecRunner.html*

* It will display the current tests present and the current status of these tests



* You can just refresh this page to get the updated results

**Step 2.1.2:** Writing a test using Jasmine

* Open Visual Studio Code and open the jasmine-standalone-2.3.4 folder in it
* Go to **spec** folder
* Create a new file named **getNthElementSpec.js**
* Write the code given below in the command prompt:

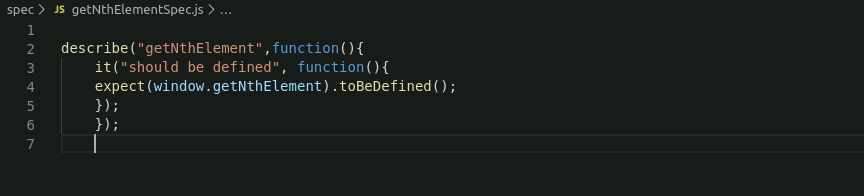
*describe("getNthElement",function(){*

*it("should be defined", function(){*

*expect(window.getNthElement).toBeDefined();*

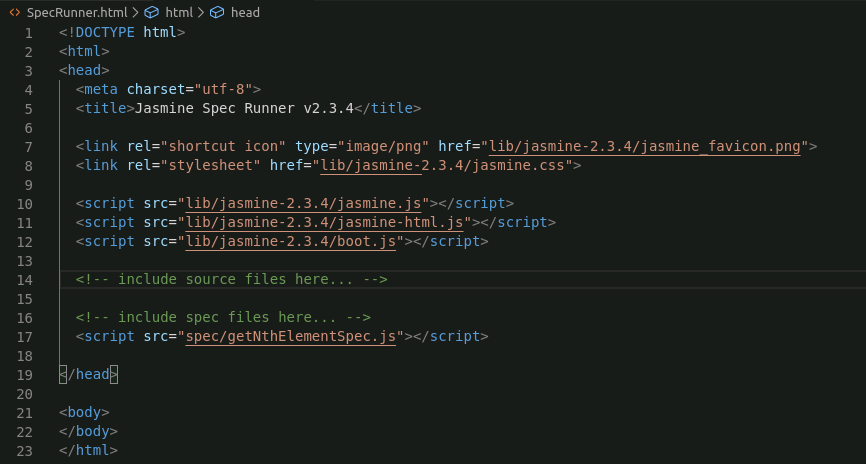
*});*

*});*



* Save the created file
* To make sure that the SpecRunner knows about the newly added spec file, open SpecRunner.html file
* Remove the source files and spec files from the SpecRunner.html file
* Copy and paste the command given below:

<script src="spec/getNthElementSpec.js"></script>

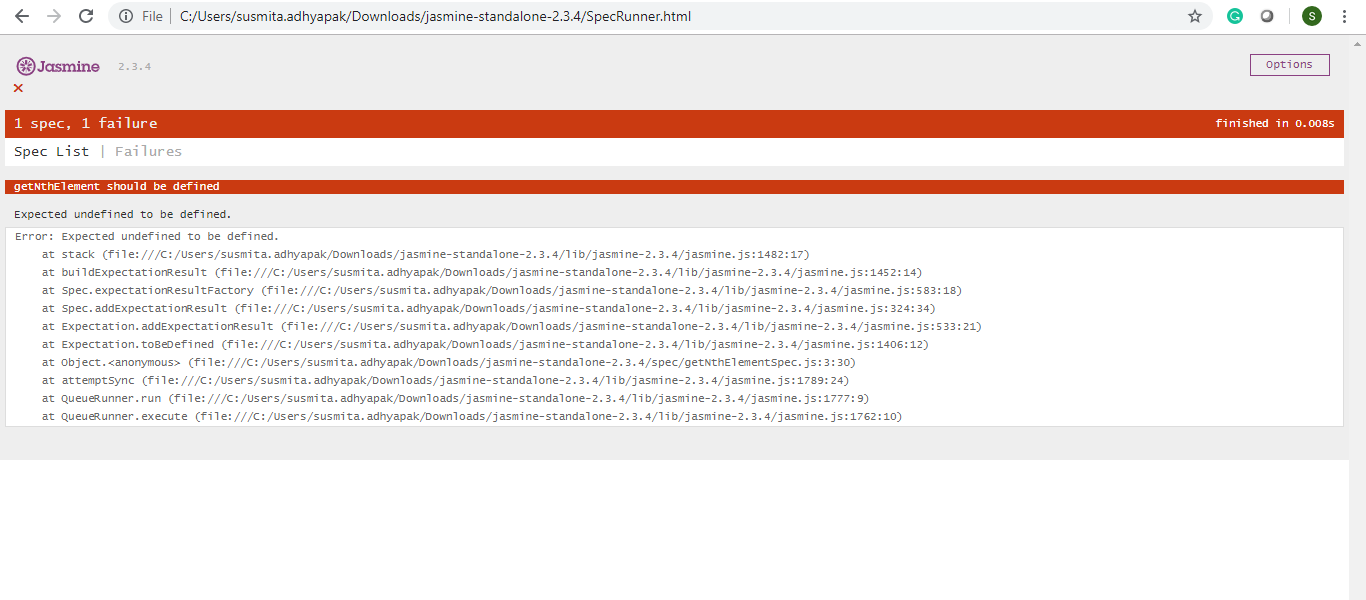


* Save the file

**Step 2.1.3:** Running the test and watching it fail

* Open SpecRunner.html file in the browser by copy-pasting the path of SpecRunner.html in the browser

*/home/ubuntu/Desktop/jasmine-standalone-2.3.4/SpecRunner.html*



* When we click on the Spec List, we can check which test suite is failing and the reason for it

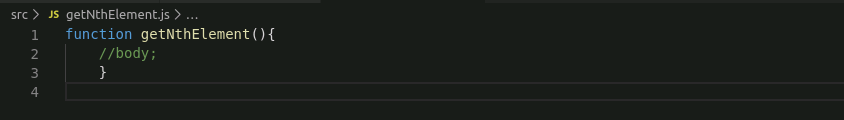
**Step 2.1.4:** Writing code to pass the tests

* Go to jasmine-standalone-2.3.4\src directory and create a file named **getNthElement.js**.
* Write the following code in the file:

*function getNthElement(){*

*//body;*

*}*



* Save the created file
* Open SpecRunner.html file
* Copy and paste the command given below:

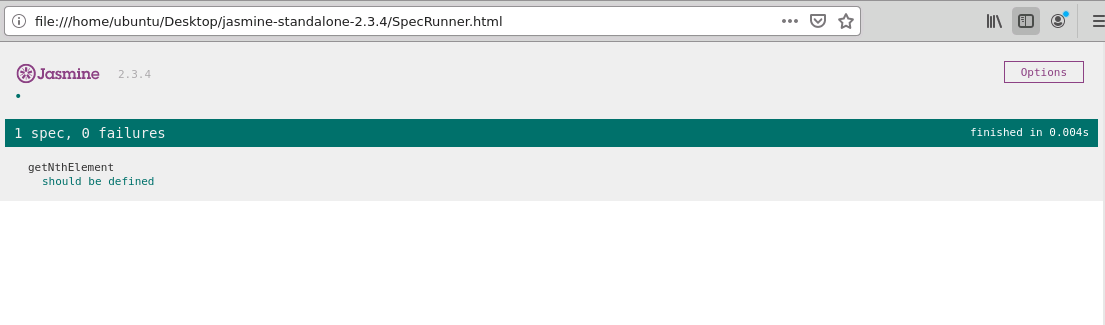
<script src="src/getNthElement.js"></script>



* Open the SpecRunner.html in the browser

*/home/ubuntu/Desktop/jasmine-standalone-2.3.4/SpecRunner.html*

* The test is passing now, but you have to add tests to find out the Nth element



**Step 2.1.5:** Refactoring the code

* Go to the **spec** folder
* Open **getNthElementSpec.js** file
* Write the code given below in the file:

*describe("getNthElement",function(){*

*it("should be defined", function(){*

*expect(window.getNthElement).toBeDefined();*

*});*

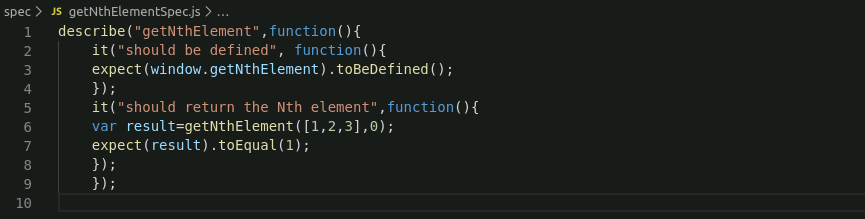
*it("should return the Nth element",function(){*

*var result=getNthElement([1,2,3],0);*

*expect(result).toEqual(1);*

*});*

*});*

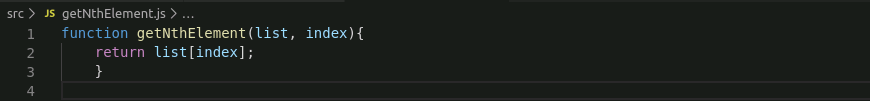


* Save the created file
* Go to the **src** folder
* Update the **getNthElement.js** file
* Write the following code in the file:

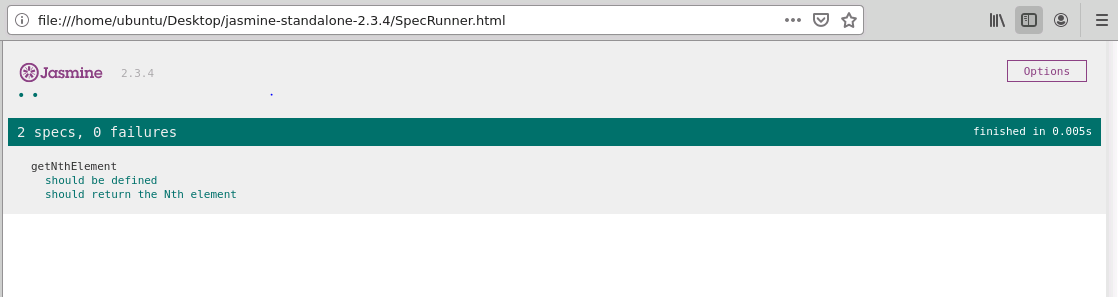
*function getNthElement(list, index){*

*return list[index];*

*}*

**

* Save the created file
* Open SpecRunner.html file in the browser by copy-pasting the path of SpecRunner.html in the browser



* It will show that the tests are running successfully

**Step 2.1.6:** Using Backbone.js in Jasmine tests

* Go to **spec** folder
* Create a new file named **backboneSpec.js**
* Write the code given below in the file:

*describe("A page with Backbone",function(){*

*it("should have access to underscore", function(){*

*expect(window.\_).toBeDefined();*

*expect(window.\_.VERSION).toBeDefined();*

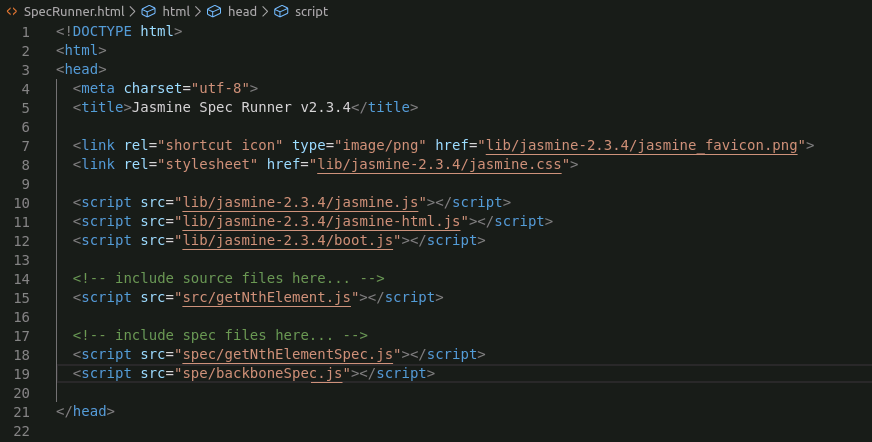
*});*

*});*



* Save the created file
* Open SpecRunner.html file with Notepad
* Copy and paste the command given below:

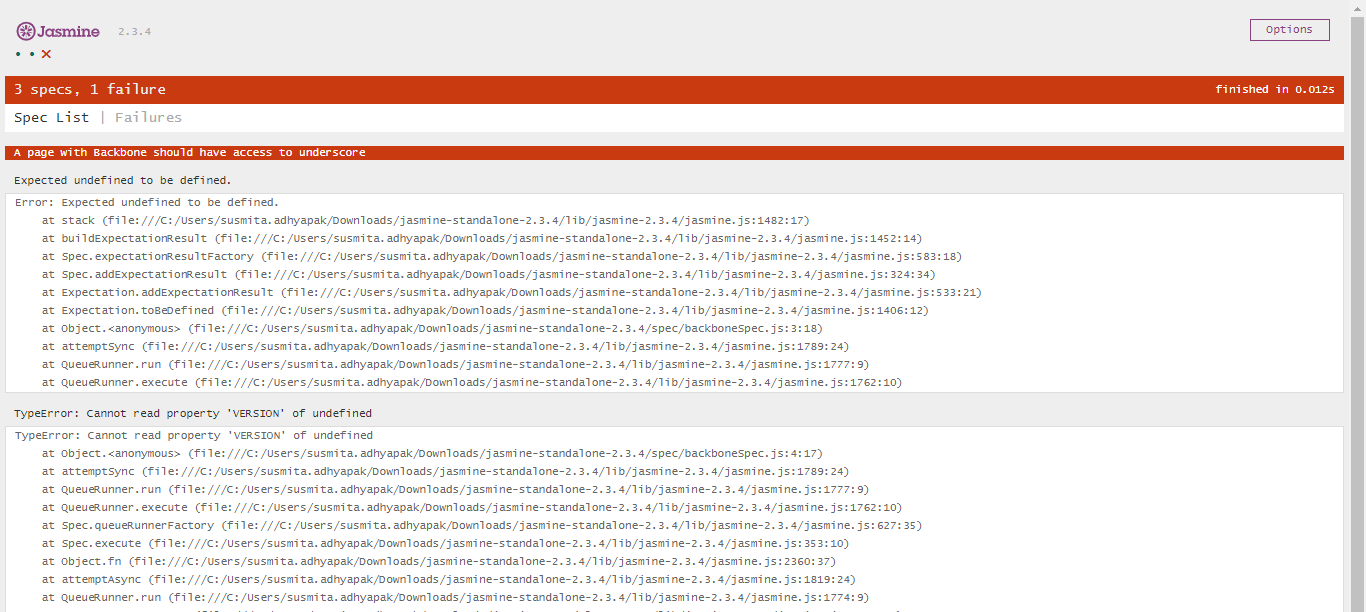
<script src="spec/backboneSpec.js"></script>



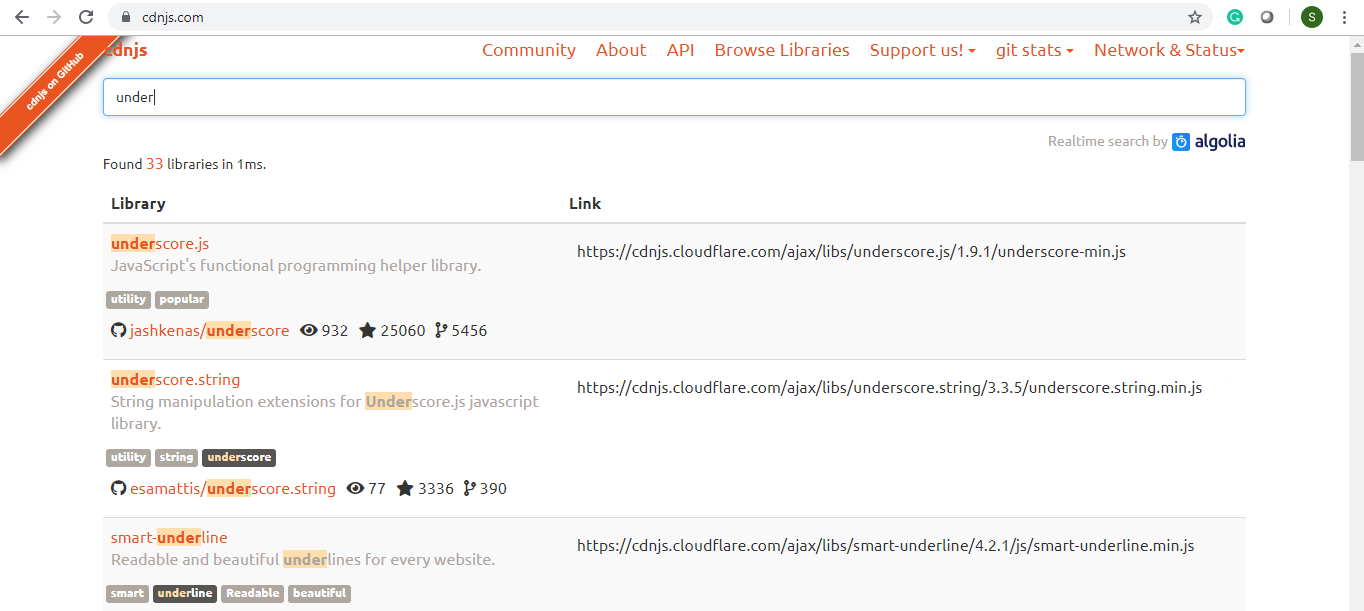
* Open SpecRunner.html file in the browser by copy-pasting the path of SpecRunner.html in the browser.

*/home/ubuntu/Desktop/jasmine-standalone-2.3.4/SpecRunner.html*

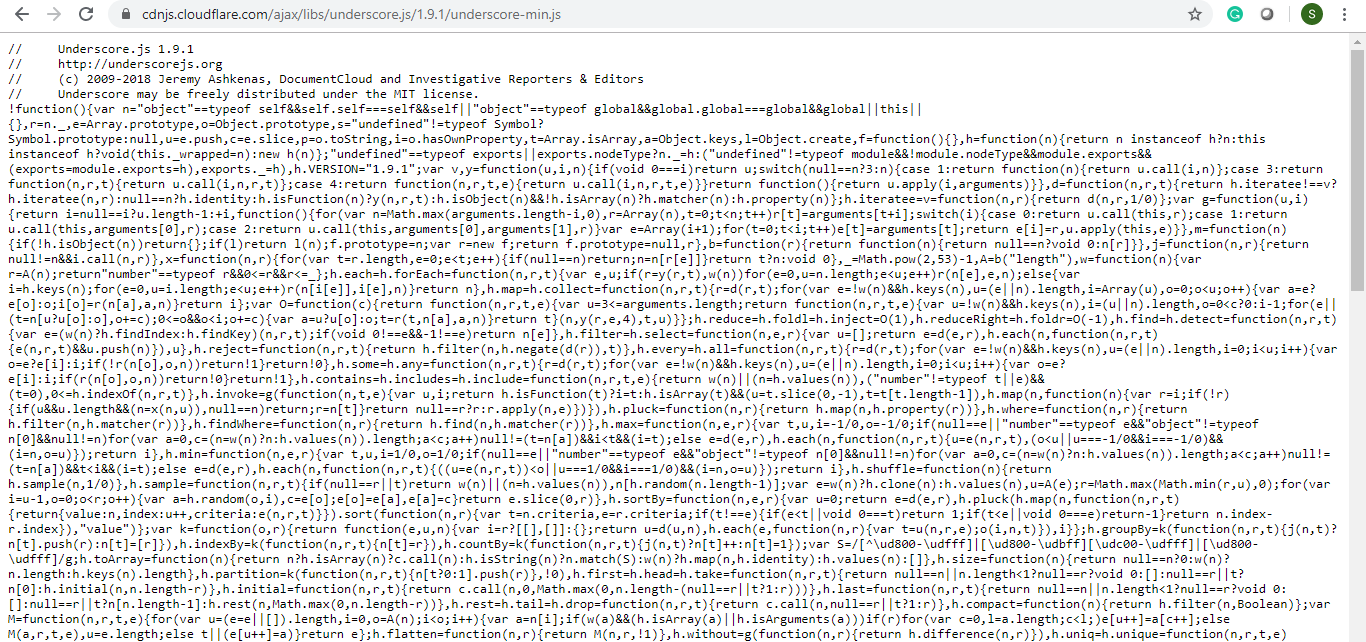
* The test will fail.



* In order to get support for underscore, add a script tag on SpecRunner.html:
  + Go to <https://cdnjs.com/>
  + Type Underscore

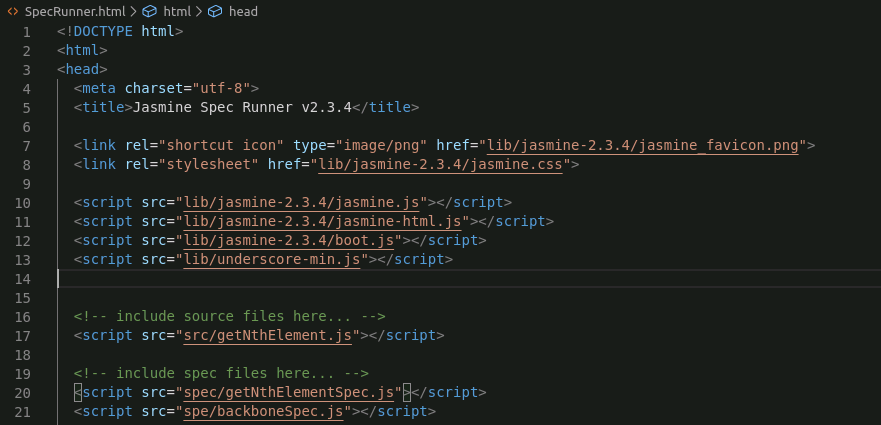


* Copy and paste the copied link to the browser
* Save the underscore-min.js file to the lib directory of Jasmine-standalone-2.3.4.



* Open SpecRunner.html.
* Copy-paste the command given below:

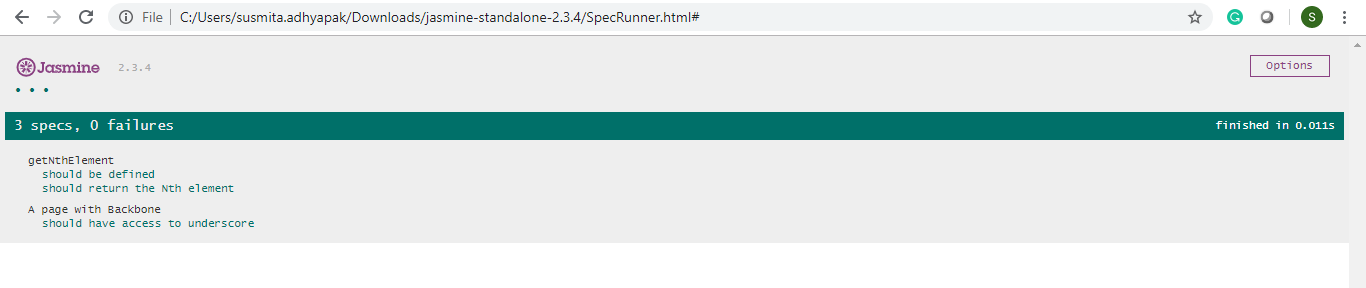
<script src="lib/underscore-min.js"></script>



* Open SpecRunner.html file in the browser by copy-pasting the path of SpecRunner.html in the browser

*/home/ubuntu/Desktop/jasmine-standalone-2.3.4/SpecRunner.html*

* The test will pass



* Refactor the code
* Go to the **spec** folder
* Modify the **backboneSpec.js** file
* Write the code given below in the file:

*describe("A page with Backbone",function(){*

*it("should have access to underscore", function(){*

*expect(window.\_).toBeDefined();*

*expect(window.\_.VERSION).toBeDefined();*

*});*

*it("should have access to Backbone",function(){*

*expect(window.Backbone).toBeDefined();*

*expect(window.Backbone.Model).toBeDefined();*

*expect(window.Backbone.Collection).toBeDefined();*

*expect(window.Backbone.View).toBeDefined();*

*});*

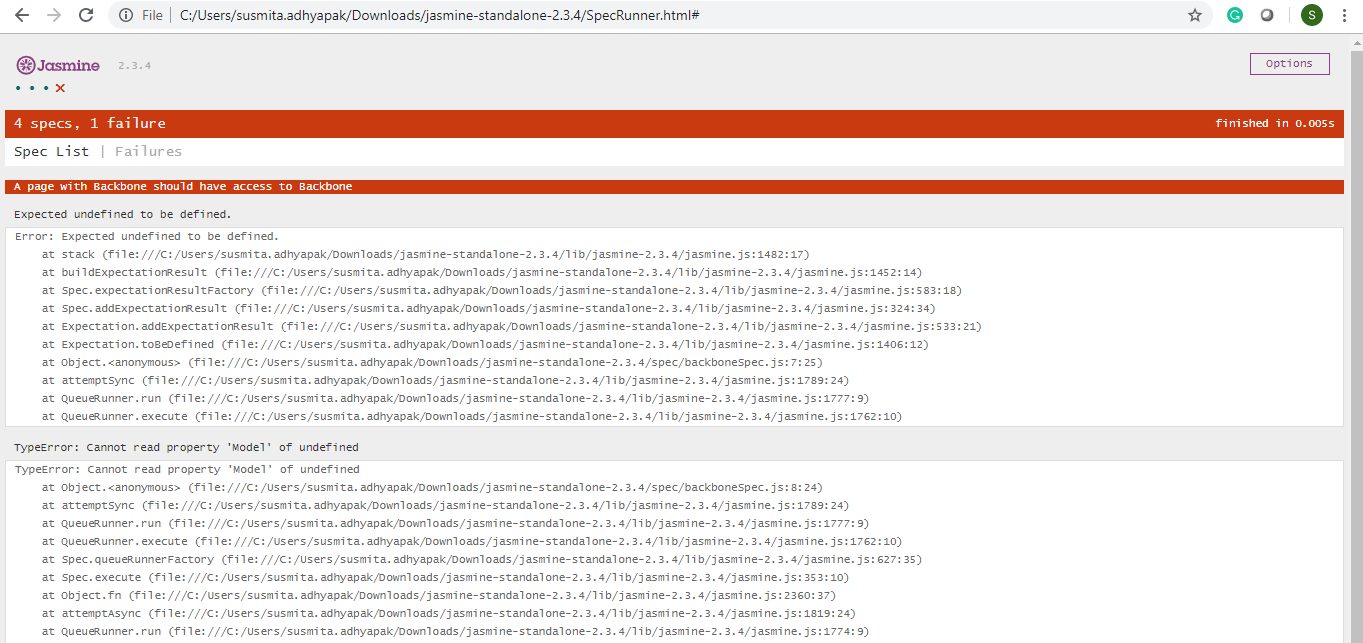
*});*

**

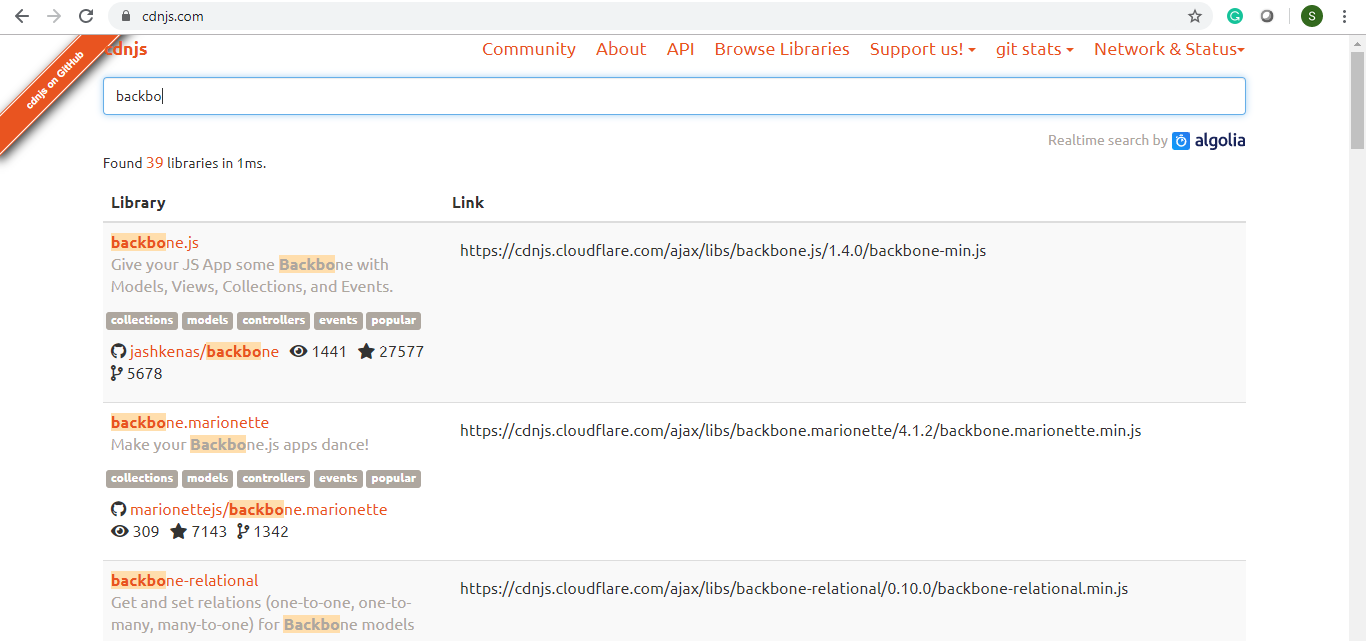
* Save the created file
* Open SpecRunner.html in browser

*/home/ubuntu/Desktop/jasmine-standalone-2.3.4/SpecRunner.html*

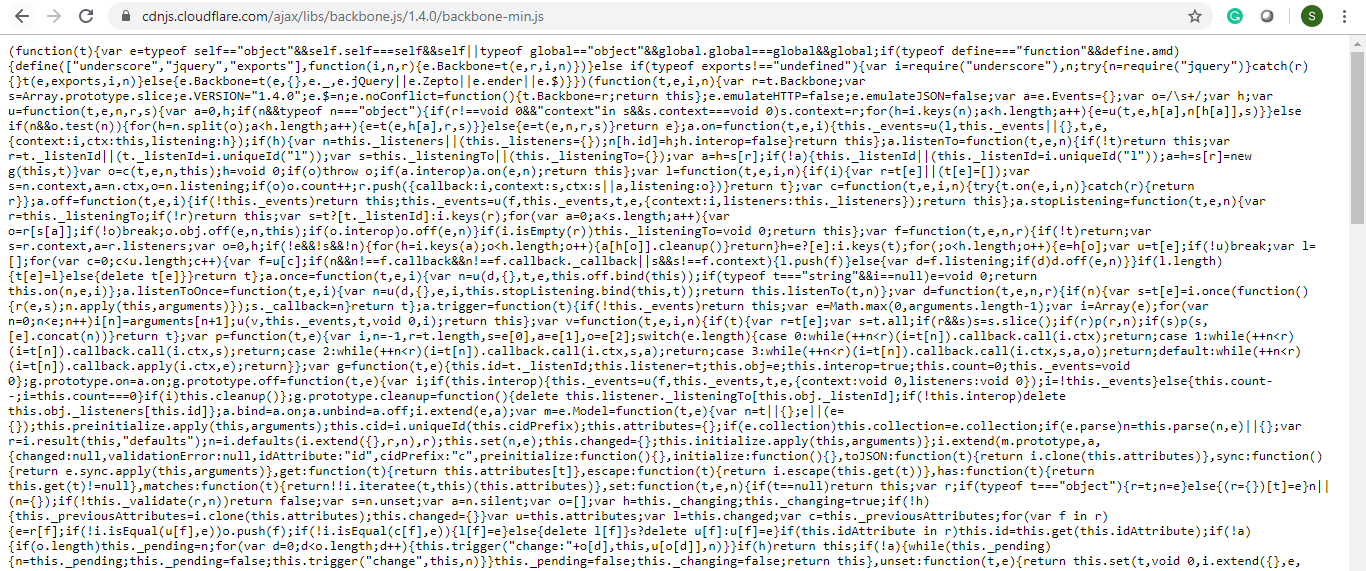
* The test will fail



* To run the test, go to <https://cdnjs.com/>
* Type backbone

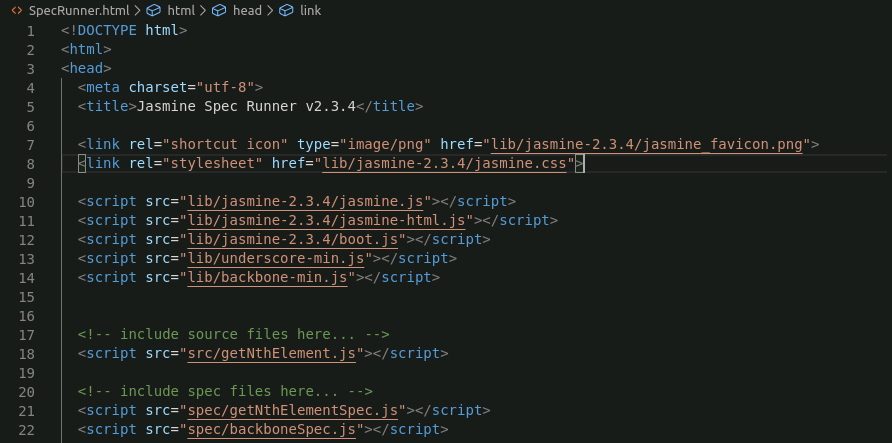


* Copy and paste the link of backbone in the new browser
* Save the file to the lib directory of jasmine



* Open SpecRunner.html
* Copy-paste the command given below:

<script src="lib/backbone-min.js"></script>

****

* Open SpecRunner.html in the browser

*/home/ubuntu/Desktop/jasmine-standalone-2.3.4/SpecRunner.html*

* All the tests are passing

**Step 2.1.7:** Pushing the code to GitHub repositories

Open your command prompt and navigate to the folder where you have created your files.

cd <folder path>

Initialize your repository using the following command:

git init

Add all the files to your git repository using the following command:

git add . 

Commit the changes using the following command:

git commit . -m “Changes have been committed.”

Push the files to the folder you created initially using the following command:

git push -u origin master