2.3 Automating the Process

This section will guide you to:

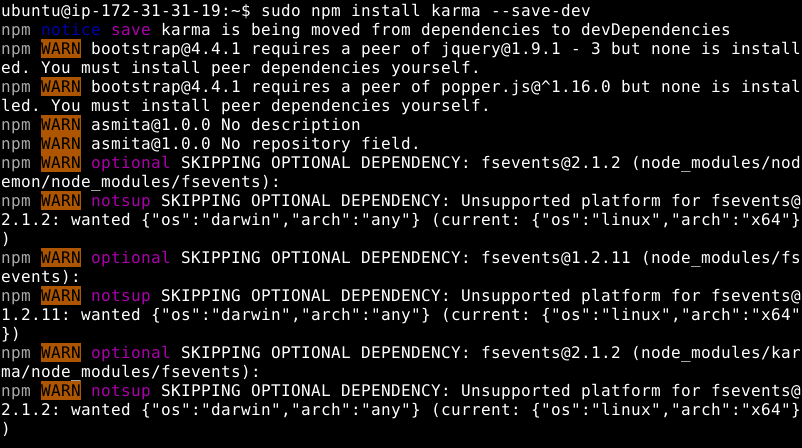
* Automate the testing using Karma

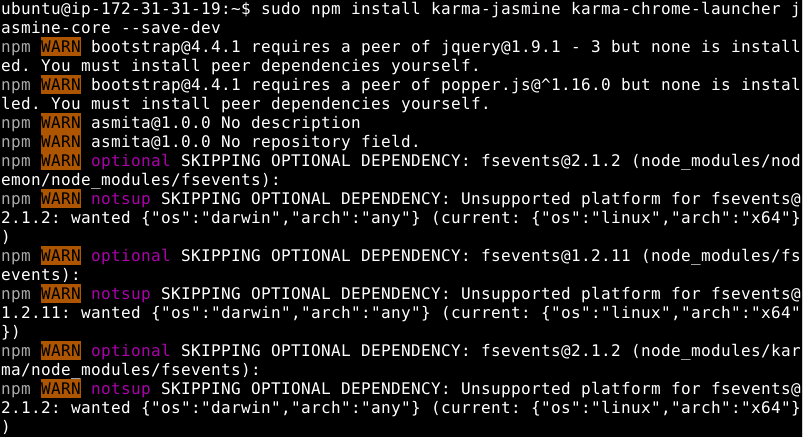
This lab has four subsections, namely:

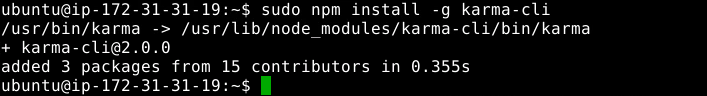
* + 1. Installing Karma
    2. Running Karma in browser
    3. Running tests with Karma
    4. Pushing the code to GitHub repositories
* *Jasmine and Karma are already installed in your lab. (Refer MEAN: Lab Guide - Phase 4)*

**Step 2.3.1:** Installing Karma

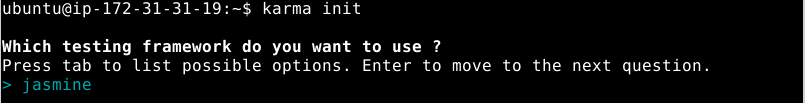
* Follow the given steps to install Karma on your local system
* Before installing Karma, make sure that Node.js and npm is installed on your system
* Open **command-line interface**
* Type **npm install karma --save-dev**
* Type **npm install karma-jasmine karma-chrome-launcher jasmine-core --save-dev.**
* This will install karma, karma-jasmine, karma-chrome-launcher, and jasmine-core packages into node\_modules in your current working directory, and also save these as devDependencies in package.json
* To run Karma on windows, type **npm install -g karma-cli**



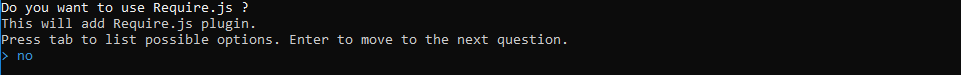




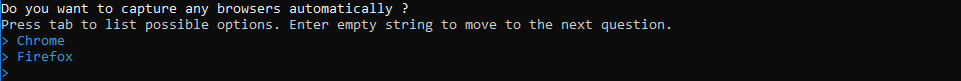
* To start the karma, type **karma init**
* Select **Jasmine** as the testing framework



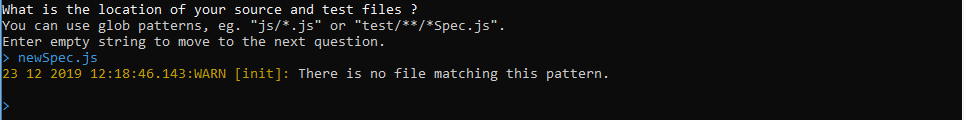
* Do not use Require.js



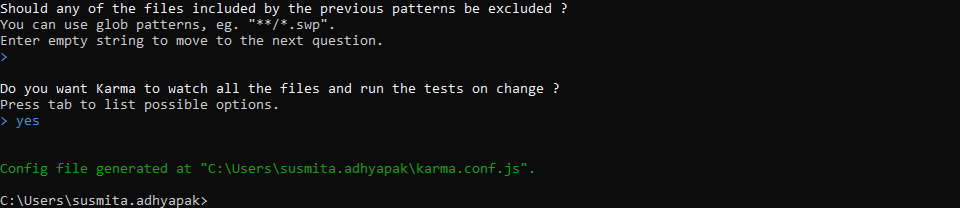
* Select Chrome and Firefox browsers



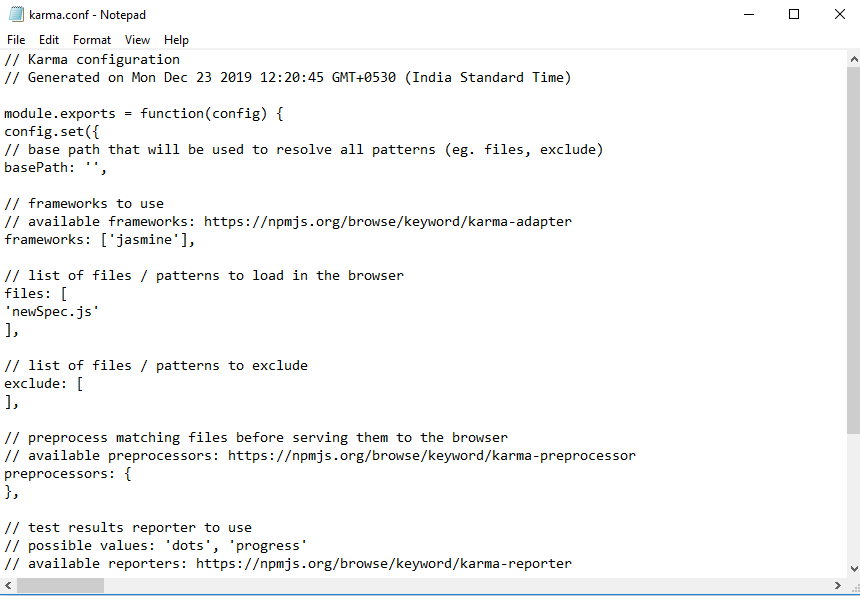
* Create a placeholder file named newSpec.js



* It will give you a warning
* Select yes to rerun the tests



* We have generated the karma.conf.js file
* Go to the directory where you have created the file
* Open the karma.conf.js file. You can see all the specifications you have declared in previous steps:



**Step 2.3.2:** Running Karma in browser

**2.3.2.1:** Create a simple Karma test

* Rename the jasmine-standalone-2.3.4 to **karma-recipe**
* Move the created **karma.cong.js file** to the karma-recipe folder
* Open Visual Studio Code and open the karma-recipe folder
* Go to the **spec** folder
* Create a  **newSpec.js** file
* Write the code given below in the command prompt:

*describe("A karma suite",function(){*

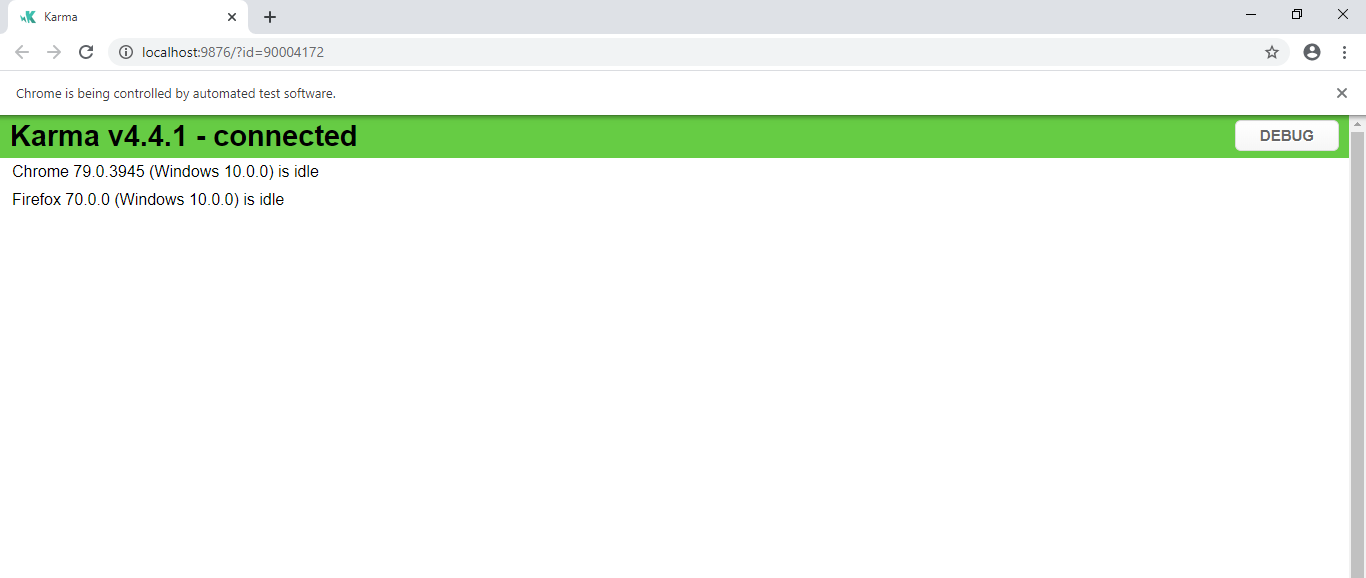
*it("should run in browsers", function(){*

*expect(true).toBeTruthy();*

*});*

*});*

* Save the created file
* Go to terminal
* Type **karma start**
* You will get some output in console
* Chrome and Firefox will load in the background.



* Karma is watching your file system for changes

2.3.2.2: Run in automatically launched browser

* Modify the newSpec.js file
* Write the code given below in the command prompt:

*describe("A karma suite",function(){*

*it("should run in browsers", function(){*

*expect(false).toBeTruthy();*

*});*

*});*

* Save the created file
* You will get an error message in the previous console
* Modify the newSpec.js file
* Write the code given below in the command prompt:

*describe("A karma suite",function(){*

*it("should run in browsers", function(){*

*expect(Object.observe).toBeTruthy();*

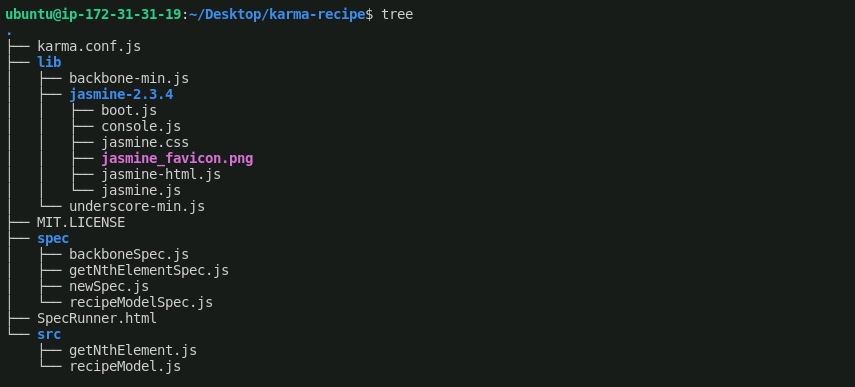
*});*

*});*

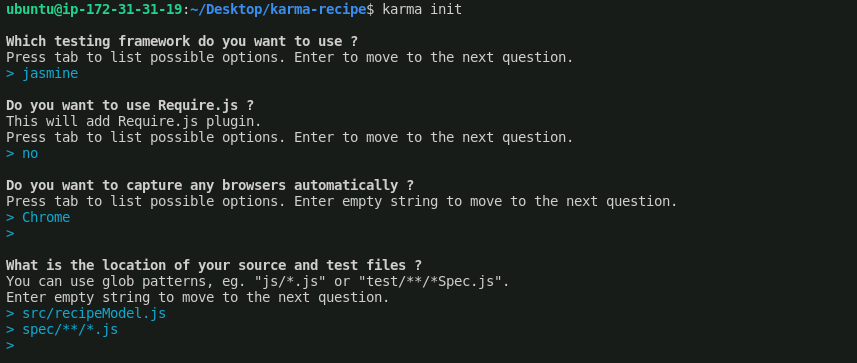
* Save the created file
* Object.observe will help you to find which browser is failing

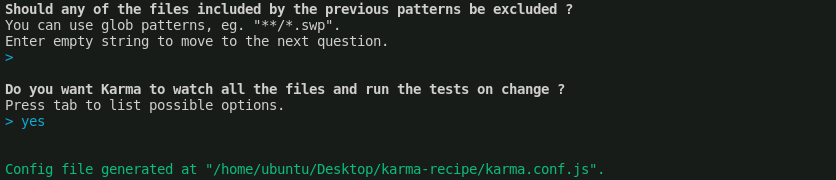
**Step 2.3.3:** Running tests with Karma

* Open karma-recipe folder
* Open the terminal
* Type **tree**

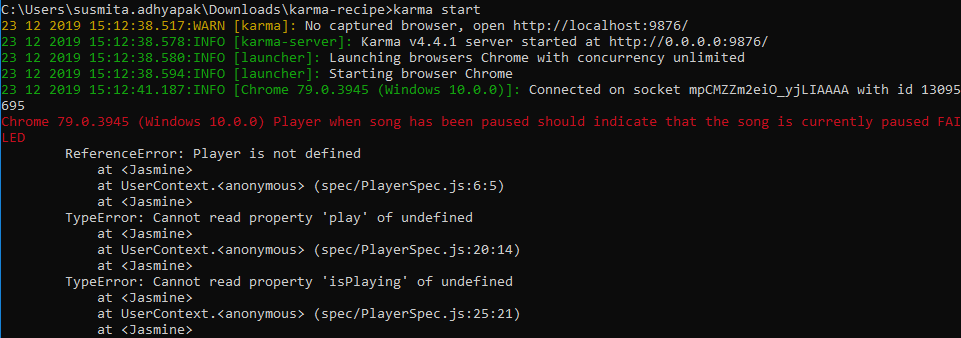


* Remove SpecRunner.html as we are going to create your own SpecRunner.html file dynamically
* Type **karma init**



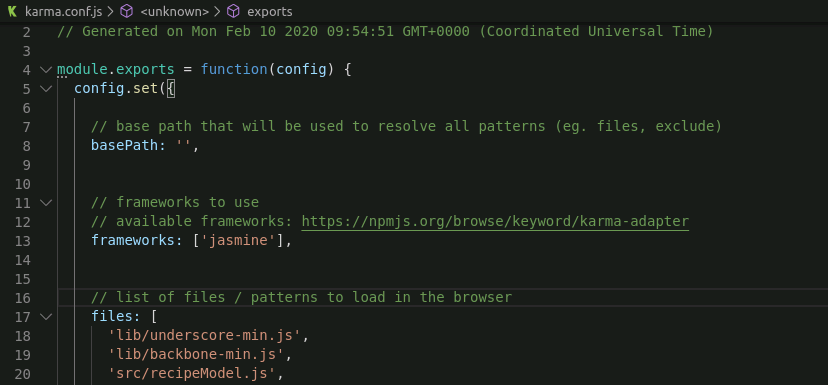


* Type **karma start** to start the karma

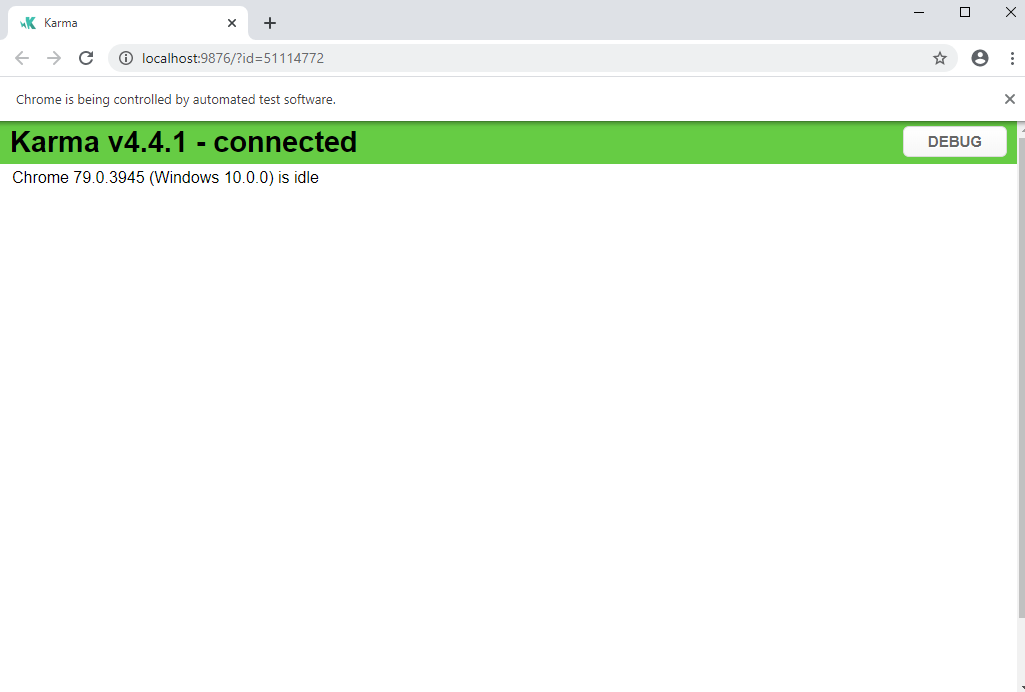


*Note: Make sure you have deleted all the files except recipeModel.js from* ***src*** *folder and deleted all the other files, except backboneSpec.js and recipeModelSpec.js from* ***spec*** *folder.*

* Open **karma.conf.js** file
* Copy and paste **'lib/underscore-min.js'** and **'lib/backbone-min.js'** in karma.conf.js before the other two file paths
* Save the file



* Open the terminal
* Type **karma start**, and it will run the Karma



**Step 2.3.4:** 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