2.5 Testing Views

This section will guide you to:

* Set up and test views

This lab has four subsections, namely:

* + 1. Setting up views
    2. Setting DOM expectations
    3. Rendering views
    4. Pushing the code to GitHub repositories

**Step 2.5.1:** Setting up views

* Open Visual Studio Code
* Open the karma-recipe folder
* Open the terminal and type **npm install sinon**
* Go to the spec folder
* Create the **recipeViewSpec.js** file
* Write the code given below in the file:

describe("A recipe view", function(){

it("should exist", function(){

expect(app.Views.Recipe).toBeDefined();

});

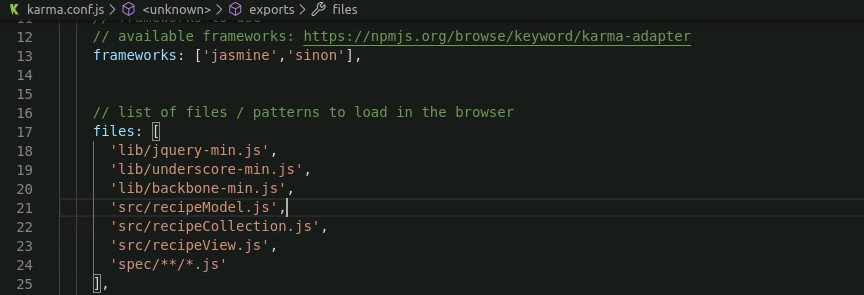
});

* Save the file
* Go to the src folder and create the **recipeView.js** file
* Write the following code in the file:

var app=window.app || {};

app.Views=app.Views || {};

* Save the file
* Open the karma.conf.js using Notepad
* Copy and paste **'src/recipeView.js'** in the list of files
* Save the file



* Open the terminal
* Type karma start, the test will fail

Go to src folder and modify the **recipeView.js** file

* Write the following code in the file:

var app=window.app || {};

app.Views=app.Views || {};

app.Views.Recipe=Backbone.View.extend({});

* Save the file
* Open the terminal
* Type karma start, this time the test will pass
* Create a new directory **templates** at src folder
* Go to templates folder
* Create a new html file named **recipeTemplate.html**
* Write the code given below:

<script type="x-handlebars" id="recipe-tmpl">

<h1 class="recipe-name">{{name}}</h1>

<h4 class="rating">{{rating}}</h4>

<ul class="ingredients">

<ul>

<ol class ="instructions">

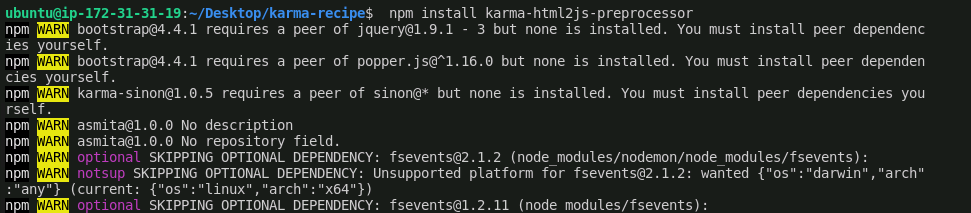
</ol>

</script>

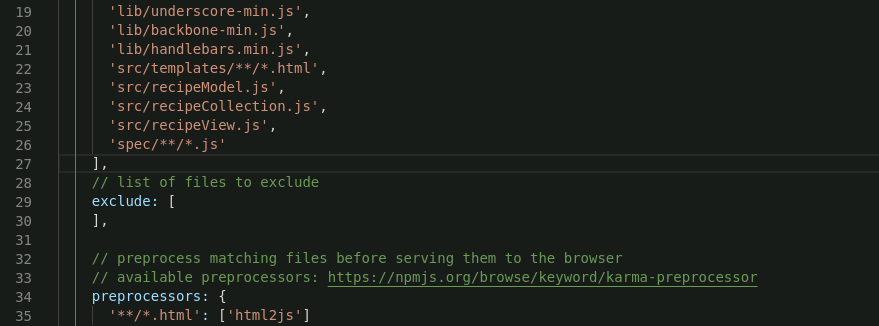
* Save the file
* Go to <https://handlebarsjs.com/>
* Go through the basics of handlebars.js

**Step 2.5.2:** Setting DOM expectations

* Go to <http://karma-runner.github.io/4.0/config/preprocessors.html>
* Go through the instructions given
* Open the terminal
* Type **npm install karma-html2js-preprocessor**



* Open the karma.conf.js
* Modify the file as shown in the image
* Save the file



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

describe("A recipe view", function(){

it("should exist", function(){

console.log(window.\_\_html\_\_);

expect(app.Views.Recipe).toBeDefined();

});

});

* Save the file
* Create a new directory **helpers** at spec folder
* Go to helpers folder
* Create a new file named **addTemplates.js**
* Write the code given below:

(function() {

$body = $('body');

for(var path in window.\_\_html\_\_) {

$body.append(window.\_\_html\_\_[path]);

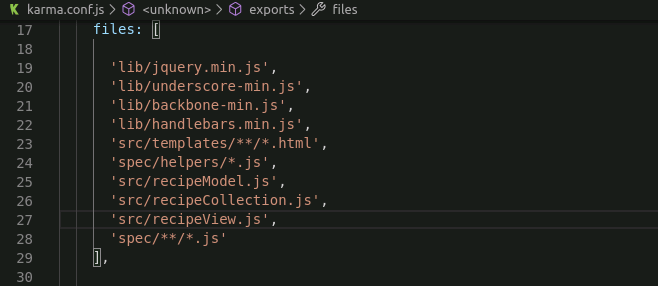
}

}());

* Save the file



* Open karma.conf.js using Notepad
* Modify the file as shown in the image
* Save the file



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

describe("A recipe view", function(){

it("should exist", function(){

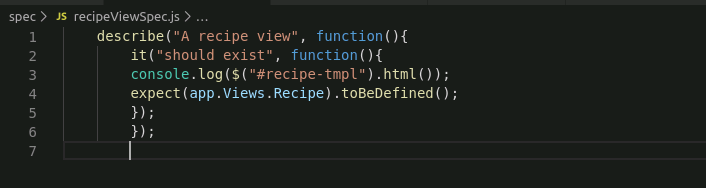
console.log($("#recipe-tmpl").html());

expect(app.Views.Recipe).toBeDefined();

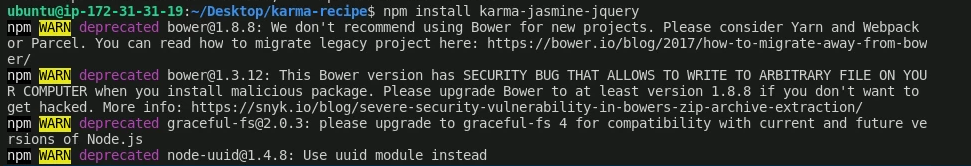
});

});

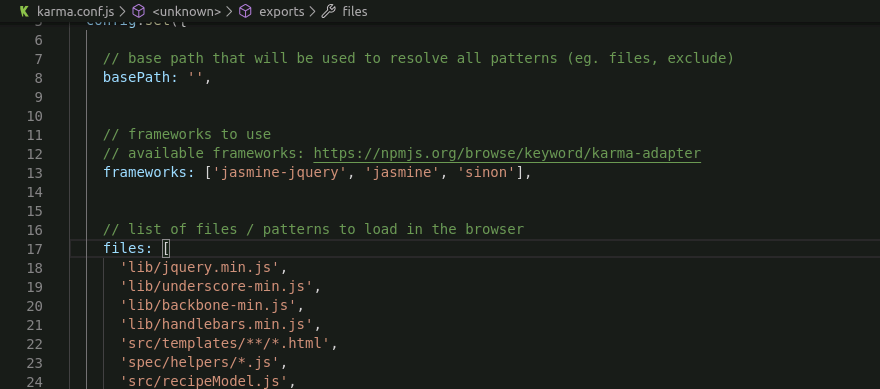
* Save the file



* Open the terminal
* Type **npm install karma-jasmine-jquery**



* Open karma.conf.js using Notepad
* Modify the file as shown in the image
* Save the file



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

describe("A recipe view", function(){

it("should exist", function(){

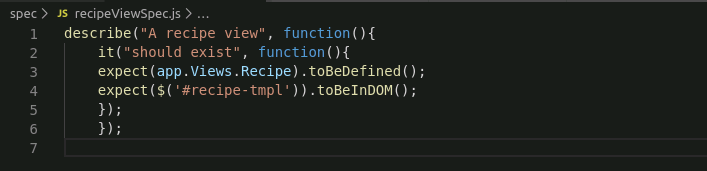
expect(app.Views.Recipe).toBeDefined();

expect($('#recipe-tmpl')).toBeInDOM();

});

});

* Save the file

****

**Step 2.5.3:** Rendering views

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

describe("A recipe view", function(){

it("should exist", function(){

expect(app.Views.Recipe).toBeDefined();

expect($('#recipe-tmpl')).toBeInDOM();

});

it('should have a template', function(){

var recipeModel=new app.Models.Recipes(FIXTURES.recipes.noRestrictions);

var recipeView=new app.Views.Recipes({

model:recipeModel

});

expect(recipeView.template).toBeDefined();

});

});

* Save the file

****

* Go to the src folder
* Modify the **recipeView.js** file
* Write the code given below in the file:

var app=window.app || {};

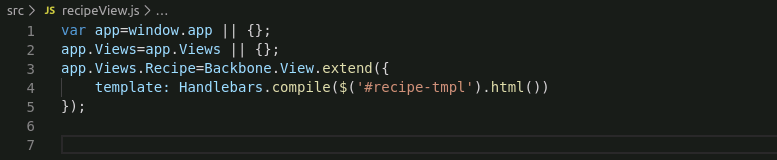
app.Views=app.Views || {};

app.Views.Recipe=Backbone.View.extend({

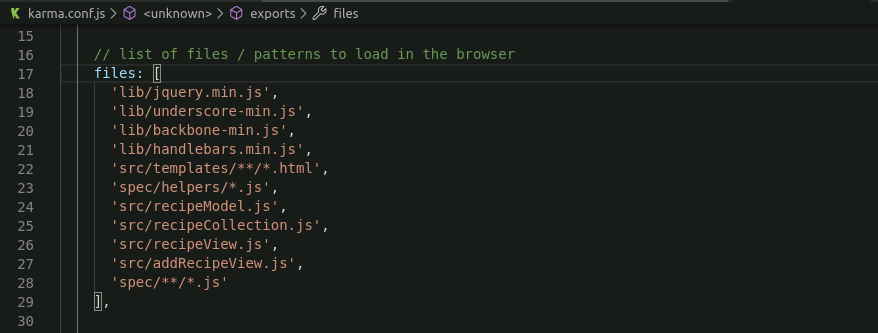
template: Handlebars.compile($('#recipe-tmpl').html())

});

* Save the file

****

* Go to <https://cdnjs.com/>
* Search for handlebars
* Download the copy of handlebars.min.js at lib directory of karma-recipe
* Open karma.conf.js using Notepad
* Modify the file as shown in the image
* Save the file



* Open the terminal
* Type karma start, the test will pass
* Now you have to render the view
* Go to the spec folder
* Modify the **recipeViewSpec.js** file
* Write the code given below in the file:

*describe('A recipe view', function() {*

*it('should exist', function() {*

*expect(app.Views.Recipe).toBeDefined();*

*expect($('#recipe-tmpl')).toBeInDOM();*

*});*

*it('should have a template', function() {*

*var recipeModel = new app.Models.Recipe(FIXTURES.recipes.noRestrictions);*

*var recipeView = new app.Views.Recipe({*

*model: recipeModel*

*});*

*expect(recipeView.template).toBeDefined();*

*});*

*describe('A rendered view', function() {*

*var recipeView, recipeModel;*

*beforeEach(function() {*

*recipeModel = new app.Models.Recipe(FIXTURES.recipes.noRestrictions);*

*recipeView = new app.Views.Recipe({*

*model: recipeModel*

*});*

*recipeView.render();*

*});*

*it('should have a title', function() {*

*expect(recipeView.$('.recipe-name')).toHaveText(recipeModel.get('name'));*

*});*

*it('should have a rating', function() {*

*expect(recipeView.$('.rating')).toHaveText(recipeModel.get('rating'));*

*});*

*it('should list ingredients', function() {*

*var ingredientEl = recipeView.$('.ingredient').first();*

*var ingredient = recipeModel.get('ingredients')[0];*

*var expectedString = ingredient.quantity + ' ' +*

*ingredient.baseIngredient + ', ' + ingredient.modifier;*

*expect(ingredientEl).toHaveText(expectedString);*

*});*

*it('should list instructions', function() {*

*var instructionEl = recipeView.$('.instruction').first();*

*var instruction = recipeModel.get('instructions')[0];*

*expect(instructionEl).toHaveText(instruction);*

*});*

*});*

*describe('A notes section', function() {*

*var recipeModel;*

*var recipeView;*

*beforeEach(function() {*

*recipeModel = new app.Models.Recipe(FIXTURES.recipes.noRestrictions);*

*recipeView = new app.Views.Recipe({*

*model: recipeModel*

*});*

*recipeView.render();*

*recipeView.$el.appendTo('body');*

*});*

*afterEach(function() {*

*recipeView.$el.remove();*

*});*

*it('should have an add notes button', function() {*

*expect(recipeView.$('.add-note')).toBeVisible();*

*});*

*it('should reveal a notes section when the button is clicked', function() {*

*expect(recipeView.$('.note-area')).not.toBeVisible();*

*recipeView.$('.add-note').click();*

*// View after clicking add note*

*expect(recipeView.$('.note-area')).toBeVisible();*

*expect(recipeView.$('.add-note')).not.toBeVisible();*

*expect(recipeView.$('.save-note')).toBeVisible();*

*expect(recipeView.$('.cancel-note')).toBeVisible();*

*});*

*it('should return to it\'s normal state when cancelled', function() {*

*recipeView.$('.add-note').click();*

*recipeView.$('.cancel-note').click();*

*// View after cancelling*

*expect(recipeView.$('.note-area')).not.toBeVisible();*

*expect(recipeView.$('.save-note')).not.toBeVisible();*

*expect(recipeView.$('.cancel-note')).not.toBeVisible();*

*expect(recipeView.$('.add-note')).toBeVisible();*

*});*

*it('should show a saved note', function() {*

*var noteStr = 'Needs some spice. Maybe red pepper flakes when adding tomatoes?';*

*// Simulate UI interactions*

*recipeView.$('.add-note').click();*

*recipeView.$('.note-area').val(noteStr);*

*recipeView.$('.save-note').click();*

*// View after saving*

*expect(recipeView.$('.note-area')).not.toBeVisible();*

*expect(recipeView.$('.save-note')).not.toBeVisible();*

*expect(recipeView.$('.cancel-note')).not.toBeVisible();*

*expect(recipeView.$('.add-note')).not.toBeVisible();*

*expect(recipeView.$('.note')).toBeVisible();*

*// Verify model has data*

*expect(recipeModel.get('note')).toBe(noteStr);*

*// Verify the view*

*expect(recipeView.$('.note')).toHaveText(noteStr);*

*});*

*});*

*});*

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

*describe('The add recipe view', function() {*

*var addRecipeView;*

*beforeEach(function() {*

*addRecipeView = new app.Views.AddRecipe();*

*addRecipeView.render();*

*addRecipeView.$el.appendTo('body');*

*});*

*afterEach(function() {*

*addRecipeView.$el.remove();*

*});*

*it('should have an editable field for name', function() {*

*expect(addRecipeView.$el).toContainElement('input.recipe-name');*

*});*

*it('should allow a user to add ingredients', function() {*

*expect(addRecipeView.$el).toContainElement('button.add-ingredient');*

*addRecipeView.$('.add-ingredient').click();*

*expect(addRecipeView.$el).toContainElement('.ingredient');*

*expect(addRecipeView.$('.ingredient')).toContainElement('input.quantity')*

*expect(addRecipeView.$('.ingredient')).toContainElement('input.base-ingredient')*

*expect(addRecipeView.$('.ingredient')).toContainElement('input.modifier')*

*});*

*it('should allow a user to remove ingredients', function() {*

*addRecipeView.$('.add-ingredient').click();*

*addRecipeView.$('.remove-ingredient').click();*

*expect(addRecipeView.$el).not.toContainElement('.ingredient');*

*});*

*it('should allow a user to add instruction steps', function() {*

*expect(addRecipeView.$el).toContainElement('button.add-instruction-step');*

*addRecipeView.$('.add-instruction-step').click();*

*expect(addRecipeView.$el).toContainElement('.instruction-step');*

*expect(addRecipeView.$('.instruction-step')).toContainElement('input.instruction-line');*

*});*

*it('should allow a user to remove instruction steps', function() {*

*addRecipeView.$('.add-instruction-step').click();*

*addRecipeView.$('.remove-instruction-step').click();*

*expect(addRecipeView.$el).not.toContainElement('.instruction-step');*

*});*

*it('should update the model when clicking save', function() {*

*// Define recipe*

*var recipeName = 'Beurre Monte';*

*var firstIngredient = {*

*quantity: '4 tablespoons',*

*baseIngredient: 'water',*

*modifier: ''*

*};*

*var secondIngredient = {*

*quantity: '1 pound',*

*baseIngredient: 'butter',*

*modifier: 'cut into chunks'*

*};*

*var instructionOne = 'Boil water';*

*var instructionTwo = 'Reduce heat and whisk butter in slowly';*

*addRecipeView.$('.recipe-name').val(recipeName);*

*// Add two ingredients*

*addRecipeView.$('.add-ingredient').click();*

*addRecipeView.$('.add-ingredient').click();*

*var firstIngredientEl = addRecipeView.$('.ingredient').first();*

*var secondIngredientEl = addRecipeView.$('.ingredient').last();*

*firstIngredientEl.find('.quantity').val(firstIngredient.quantity);*

*firstIngredientEl.find('.base-ingredient').val(firstIngredient.baseIngredient);*

*secondIngredientEl.find('.quantity').val(secondIngredient.quantity);*

*secondIngredientEl.find('.base-ingredient').val(secondIngredient.baseIngredient);*

*secondIngredientEl.find('.modifier').val(secondIngredient.modifier);*

*// Add two steps*

*addRecipeView.$('.add-instruction-step').click();*

*addRecipeView.$('.add-instruction-step').click();*

*var firstInstructionEl = addRecipeView.$('.instruction-step').first();*

*var secondInstructionEl = addRecipeView.$('.instruction-step').last();*

*firstInstructionEl.find('.instruction-line').val(instructionOne);*

*secondInstructionEl.find('.instruction-line').val(instructionTwo);*

*addRecipeView.$('.save-recipe').click();*

*expect(addRecipeView.model.get('name')).toEqual(recipeName);*

*expect(addRecipeView.model.get('ingredients')).toEqual([firstIngredient, secondIngredient]);*

*expect(addRecipeView.model.get('instructions')).toEqual([instructionOne, instructionTwo]);*

*});*

*});*

* Save the file
* Go to the src folder
* Modify the **recipeView.js** view
* Write the code given below in the file:

*var app = window.app || {};*

*app.Views = app.Views || {};*

*app.Views.Recipe = Backbone.View.extend({*

*template: Handlebars.compile($('#recipe-tmpl').html()),*

*initialize: function() {*

*this.listenTo(this.model, 'change:note', this.updateNoteDisplay);*

*return this;*

*},*

*render: function() {*

*var data = this.model.toJSON();*

*this.$el.html(this.template(data));*

*},*

*events: {*

*'click .add-note': 'triggerAddNoteView',*

*'click .cancel-note': 'triggerNormalView',*

*'click .save-note': 'saveNote'*

*},*

*triggerAddNoteView: function() {*

*this.$('.note-area, .save-note, .cancel-note').removeClass('hidden');*

*this.$('.add-note').addClass('hidden');*

*},*

*triggerNormalView: function() {*

*this.$('.note-area, .save-note, .cancel-note').addClass('hidden');*

*this.$('.add-note').removeClass('hidden');*

*},*

*saveNote: function() {*

*this.$('.note-area, .save-note, .cancel-note, .add-note').addClass('hidden');*

*this.$('.note').removeClass('hidden');*

*var note = this.$('.note-area').val();*

*this.model.set('note', note);*

*},*

*updateNoteDisplay: function() {*

*this.$('.note').text(this.model.get('note'));*

*}*

*});*

* Save the file
* Go to the src
* Create a **addRecipeView.js** file
* Write the code given below in the file:

*var app = window.app || {};*

*app.Views = app.Views || {};*

*app.Views.AddRecipe = Backbone.View.extend({*

*template: Handlebars.compile($('#add-recipe-tmpl').html()),*

*ingredientTemplate: Handlebars.compile($('#add-ingredient-tmpl').html()),*

*instructionStepTemplate: Handlebars.compile($('#add-instruction-step-tmpl').html()),*

*events: {*

*'click .add-ingredient': 'addEmptyIngredient',*

*'click .remove-ingredient': 'removeIngredient',*

*'click .add-instruction-step': 'addEmptyInstructionStep',*

*'click .remove-instruction-step': 'removeInstructionStep',*

*'click .save-recipe': 'saveRecipe'*

*},*

*initialize: function() {*

*if(!this.model) {*

*this.model = new app.Models.Recipe();*

*}*

*},*

*render: function() {*

*var data = this.model.toJSON();*

*this.$el.html(this.template(data));*

*return this;*

*},*

*addEmptyIngredient: function() {*

*this.$('.ingredient-list').append(this.ingredientTemplate());*

*},*

*removeIngredient: function(evt) {*

*$(evt.target).closest('.ingredient').remove();*

*},*

*addEmptyInstructionStep: function() {*

*this.$('.instruction-steps').append(this.instructionStepTemplate());*

*},*

*removeInstructionStep: function(evt) {*

*$(evt.target).closest('.instruction-step').remove();*

*},*

*saveRecipe: function() {*

*// Save the name*

*this.model.set('name', this.$('.recipe-name').val());*

*// Assemble and save ingredients*

*var ingredients = [];*

*this.$('.ingredient').each(function() {*

*var ingredient = {};*

*var $el = $(this);*

*ingredient.quantity = $el.find('.quantity').val();*

*ingredient.baseIngredient = $el.find('.base-ingredient').val();*

*ingredient.modifier = $el.find('.modifier').val();*

*ingredients.push(ingredient);*

*});*

*this.model.set('ingredients', ingredients);*

*// Assemble and save instructions*

*var instructions = [];*

*this.$('.instruction-line').each(function() {*

*instructions.push(this.value);*

*});*

*this.model.set('instructions', instructions);*

*}*

*});*

* Save the file
* Go to the src\templates folder
* Modify a **recipeTemplate.html** file
* Write the code given below in the file:

*<script type="x-handlebars" id="recipe-tmpl">*

*<h1 class="recipe-name">{{name}}</h1>*

*<h4 class="rating">{{rating}}</h4>*

*<ul class="ingredients">*

*{{#each ingredients}}*

*<li class="ingredient">{{quantity}} {{baseIngredient}}, {{modifier}}</li>*

*{{/each}}*

*</ul>*

*<ol class="instructions">*

*{{#each instructions}}*

*<li class="instruction">{{this}}</li>*

*{{/each}}*

*</ol>*

*<p class="note hidden"></p>*

*<textarea class="note-area hidden"></textarea>*

*<button class="save-note hidden">Save Note</button>*

*<button class="cancel-note hidden">Cancel</button>*

*<button class="add-note">Add a Note</button>*

*</script>*

* Save the file
* Go to src\templates folder
* Create a **addRecipeTemplate.html** file
* Write the code given below in the file:

*<script type="x-handlebars" id="add-recipe-tmpl">*

*<label>Recipe Name: <input type="text" class="recipe-name"/></label>*

*<h4>Ingredients <button class="add-ingredient">Add an ingredient</button></h4>*

*<div class="ingredient-list"></div>*

*<h4>Instructions <button class="add-instruction-step">Add a step</button></h4>*

*<div class="instruction-steps"> </div>*

*<button class="save-recipe">Save Recipe</button>*

*</script>*

*<script type="x-handlebars" id="add-ingredient-tmpl">*

*<div class="ingredient">*

*<label>Quantity: <input type="text" class="quantity"/></label>*

*<label>Base Ingredient: <input type="text" class="base-ingredient"/></label>*

*<label>Modifier: <input type="text" class="modifier"/></label>*

*<button class="remove-ingredient">Remove Ingredient</button>*

*</div>*

*</script>*

*<script type="x-handlebars" id="add-instruction-step-tmpl">*

*<div class="instruction-step">*

*<input type="text" class="instruction-line"/>*

*<button class="remove-instruction-step">Remove Step</button>*

*</div>*

*</script>*

* Save the file
* Go to the spec\fixtures folder
* Modify the **recipes.js** file
* Write the code given below in the file:

*var FIXTURES = window.FIXTURES || {};*

*FIXTURES.recipes = {*

*noRestrictions: {*

*name: 'Marinara',*

*dateAdded: 1421301600000,*

*ingredients: [*

*{*

*baseIngredient: 'pomodoros',*

*quantity: '8',*

*modifier: 'crushed'*

*},*

*{*

*baseIngredient: 'onion',*

*quantity: '1',*

*modifier: 'chopped'*

*},*

*{*

*baseIngredient: 'garlic',*

*quantity: '2 clove',*

*modifier: 'minced'*

*}*

*],*

*instructions: [*

*'Saute onion in 2 Tbsp. olive oil',*

*'Add garlic when onions begin to soften, about 4 minutes',*

*'Add tomatoes and lower heat to low, stirring occasionally',*

*'Season with salt and pepper to taste'*

*]*

*},*

*noRestrictionsRecent: {*

*name: 'Marinara',*

*dateAdded: 1421388000000,*

*ingredients: [*

*{*

*baseIngredient: 'pomodoros',*

*quantity: '10',*

*modifier: 'crushed'*

*},*

*{*

*baseIngredient: 'onion',*

*quantity: '1',*

*modifier: 'chopped'*

*},*

*{*

*baseIngredient: 'garlic',*

*quantity: '2 clove',*

*modifier: 'minced'*

*}*

*]*

*},*

*withNuts: {*

*name: 'Broccoli with Butter Sauce',*

*dateAdded: 1421388000000,*

*ingredients: [*

*{*

*baseIngredient: 'broccoli',*

*quantity: '1 head',*

*modifier: ''*

*},*

*{*

*baseIngredient: 'butter',*

*quantity: '3 tablespoons',*

*modifier: 'melted'*

*},*

*{*

*baseIngredient: 'lemon',*

*quantity: '1/4',*

*modifier: 'juiced'*

*},*

*{*

*baseIngredient: 'almonds',*

*isNut: true,*

*quantity: '1/2 cup',*

*modifier: 'slivered'*

*}*

*]*

*},*

*withEggs: {*

*name: 'Eggs Benedict',*

*dateAdded: 1421474400000,*

*ingredients: [*

*{*

*baseIngredient: 'Eggs',*

*isEggs: true,*

*quantity: '4',*

*modifier: ''*

*},*

*{*

*baseIngredient: 'English Muffin',*

*quantity: '2',*

*modifier: ''*

*},*

*{*

*baseIngredient: 'Canadian Bacon',*

*quantity: '4 slices',*

*modifier: 'julienned'*

*},*

*{*

*baseIngredient: 'Hollandaise sauce',*

*quantity: '1/2 cup',*

*modifier: ''*

*}*

*]*

*},*

*withMeat: {*

*name: 'Chicken Parmesan',*

*dateAdded: 1421560800000,*

*ingredients: [*

*{*

*baseIngredient: 'Chicken',*

*isMeat: true,*

*quantity: '1 breast',*

*modifier: ''*

*},*

*{*

*baseIngredient: 'Mozzarella Cheese',*

*quantity: '8 ounces',*

*modifier: 'shredded'*

*},*

*{*

*baseIngredient: 'Pasta Sauce',*

*quantity: '1 jar',*

*modifier: ''*

*},*

*{*

*baseIngredient: 'Breadcrumbs',*

*quantity: '1 cup',*

*modifier: ''*

*}*

*]*

*}*

*};*

* Save the file
* Create a new directory named css at src folder
* Create a new file named **style.css**
* Write the given code in the file:

.hidden {

display: none;

}

* Save the file
* Open karma.conf.js using Notepad
* Make sure that all the file names on which you have worked on are mentioned in karma.conf.js file
* Save the file

****

* Open a terminal
* Type karma start
* The tests will pass

**Step 2.5.4:** 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