Protractor Examples:

**Controller Definition:**

var myApp = angular.module('myApp',[]);

myApp.controller('MyController', function($scope) {

$scope.spices = [{"name":"pasilla", "spiciness":"mild"},

{"name":"jalapeno", "spiciness":"hot hot hot!"},

{"name":"habanero", "spiciness":"LAVA HOT!!"}];

$scope.spice = "habanero";

});

**Controller Test:**

describe('myController function', function() {

describe('myController', function() {

var $scope;

beforeEach(module('myApp'));

beforeEach(inject(function($rootScope, $controller) {

$scope = $rootScope.$new();

$controller('MyController', {$scope: $scope});

}));

it('should create "spices" model with 3 spices', function() {

expect($scope.spices.length).toBe(3);

});

it('should set the default value of spice', function() {

expect($scope.spice).toBe('habanero');

});

});

});

If you need to test a nested Controller you need to create the same scope hierarchy in your test that exists in the DOM:

describe('state', function() {

var mainScope, childScope, grandChildScope;

beforeEach(module('myApp'));

beforeEach(inject(function($rootScope, $controller) {

mainScope = $rootScope.$new();

$controller('MainController', {$scope: mainScope});

childScope = mainScope.$new();

$controller('ChildController', {$scope: childScope});

grandChildScope = childScope.$new();

$controller('GrandChildController', {$scope: grandChildScope});

}));

it('should have over and selected', function() {

expect(mainScope.timeOfDay).toBe('morning');

expect(mainScope.name).toBe('Nikki');

expect(childScope.timeOfDay).toBe('morning');

expect(childScope.name).toBe('Mattie');

expect(grandChildScope.timeOfDay).toBe('evening');

expect(grandChildScope.name).toBe('Gingerbread Baby');

});

});

Using a Service

To use an Angular service, you add it as a dependency for the component (controller, service, filter or directive) that depends on the service. Angular's [dependency injection](https://docs.angularjs.org/guide/di) subsystem takes care of the rest.

angular.

module('myServiceModule', []).

controller('MyController', ['$scope','notify', function ($scope, notify) {

$scope.callNotify = function(msg) {

notify(msg);

};

}]).

factory('notify', ['$window', function(win) {

var msgs = [];

return function(msg) {

msgs.push(msg);

if (msgs.length == 3) {

win.alert(msgs.join("\n"));

msgs = [];

}

};

}]);

<div id="simple" ng-controller="MyController">

<p>Let's try this simple notify service, injected into the controller...</p>

<input ng-init="message='test'" ng-model="message" >

<button ng-click="callNotify(message);">NOTIFY</button>

<p>(you have to click 3 times to see an alert)</p>

</div>

TEST:

it('should test service', function() {

expect(element(by.id('simple')).element(by.model('message')).getAttribute('value'))

.toEqual('test');

});

angular.module('scopeExample', [])

.controller('MyController', ['$scope', function($scope) {

$scope.username = 'World';

$scope.sayHello = function() {

$scope.greeting = 'Hello ' + $scope.username + '!';

};

}]);

<div ng-controller="MyController">

Your name:

<input type="text" ng-model="username">

<button ng-click='sayHello()'>greet</button>

<hr>

{{greeting}}

</div>

it('should say hello', function() {

var scopeMock = {};

var cntl = new MyController(scopeMock);

// Assert that username is pre-filled

expect(scopeMock.username).toEqual('World');

// Assert that we read new username and greet

scopeMock.username = 'angular';

scopeMock.sayHello();

expect(scopeMock.greeting).toEqual('Hello angular!');

});

angular.module('expressionExample', [])

.controller('ExampleController', ['$window', '$scope', function($window, $scope) {

$scope.name = 'World';

$scope.greet = function() {

$window.alert('Hello ' + $scope.name);

};

}]);

it('should calculate expression in binding', function() {

if (browser.params.browser == 'safari') {

// Safari can't handle dialogs.

return;

}

element(by.css('[ng-click="greet()"]')).click();

var alertDialog = browser.switchTo().alert();

expect(alertDialog.getText()).toEqual('Hello World');

alertDialog.accept();

});

angular.module('oneTimeBidingExampleApp', []).

controller('EventController', ['$scope', function($scope) {

var counter = 0;

var names = ['Igor', 'Misko', 'Chirayu', 'Lucas'];

/\*

\* expose the event object to the scope

\*/

$scope.clickMe = function(clickEvent) {

$scope.name = names[counter % names.length];

counter++;

};

}]);

it('should freeze binding after its value has stabilized', function() {

var oneTimeBiding = element(by.id('one-time-binding-example'));

var normalBinding = element(by.id('normal-binding-example'));

expect(oneTimeBiding.getText()).toEqual('One time binding:');

expect(normalBinding.getText()).toEqual('Normal binding:');

element(by.buttonText('Click Me')).click();

expect(oneTimeBiding.getText()).toEqual('One time binding: Igor');

expect(normalBinding.getText()).toEqual('Normal binding: Igor');

element(by.buttonText('Click Me')).click();

expect(oneTimeBiding.getText()).toEqual('One time binding: Igor');

expect(normalBinding.getText()).toEqual('Normal binding: Misko');

element(by.buttonText('Click Me')).click();

element(by.buttonText('Click Me')).click();

expect(oneTimeBiding.getText()).toEqual('One time binding: Igor');

expect(normalBinding.getText()).toEqual('Normal binding: Lucas');

});

angular.module('docsBindExample', [])

.controller('Controller', ['$scope', function($scope) {

$scope.name = 'Max Karl Ernst Ludwig Planck (April 23, 1858 – October 4, 1947)';

}]);

<div ng-controller="Controller">

Hello <input ng-model='name'> <hr/>

<span ng-bind="name"></span> <br/>

<span ng:bind="name"></span> <br/>

<span ng\_bind="name"></span> <br/>

<span data-ng-bind="name"></span> <br/>

<span x-ng-bind="name"></span> <br/>

</div>

it('should show off bindings', function() {

expect(element(by.css('div[ng-controller="Controller"] span[ng-bind]')).getText())

.toBe('Max Karl Ernst Ludwig Planck (April 23, 1858 – October 4, 1947)');

});

describe('sorting the list of users', function() {

it('sorts in descending order by default', function() {

var users = ['jack', 'igor', 'jeff'];

var sorted = sortUsers(users);

expect(sorted).toEqual(['jeff', 'jack', 'igor']);

});

});

angular.module('app', [])

.controller('PasswordController', function PasswordController($scope) {

$scope.password = '';

$scope.grade = function() {

var size = $scope.password.length;

if (size > 8) {

$scope.strength = 'strong';

} else if (size > 3) {

$scope.strength = 'medium';

} else {

$scope.strength = 'weak';

}

};

});

describe('PasswordController', function() {

beforeEach(module('app'));

var $controller;

beforeEach(inject(function(\_$controller\_){

// The injector unwraps the underscores (\_) from around the parameter names when matching

$controller = \_$controller\_;

}));

describe('$scope.grade', function() {

it('sets the strength to "strong" if the password length is >8 chars', function() {

var $scope = {};

var controller = $controller('PasswordController', { $scope: $scope });

$scope.password = 'longerthaneightchars';

$scope.grade();

expect($scope.strength).toEqual('strong');

});

});

});

describe('TODO list', function() {

it('should filter results', function() {

// Find the element with ng-model="user" and type "jacksparrow" into it

element(by.model('user')).sendKeys('jacksparrow');

// Find the first (and only) button on the page and click it

element(by.css(':button')).click();

// Verify that there are 10 tasks

expect(element.all(by.repeater('task in tasks')).count()).toEqual(10);

// Enter 'groceries' into the element with ng-model="filterText"

element(by.model('filterText')).sendKeys('groceries');

// Verify that now there is only one item in the task list

expect(element.all(by.repeater('task in tasks')).count()).toEqual(1);

});

});

baseUrl: 'http://localhost:8000/app/',