RICE®

Web Development

COMP 431 / COMP 531

Front-End Unit Testing

Scott E Pollack, PhD

February 18, 2016

Front-End Recap

- HTML and HTML5
- JavaScript and JS Libraries
- Forms and Events
- CSS and Style Frameworks
- MVC and HTML Templates
- Homework Assignment 4 (JavaScript Game)
  - Due TONIGHT

Unit Testing
Angular Services
Headless Browsing
SPRING BREAK

Homework Assignment 5
(Front-End App)
Due Thursday 3/10

COMP 531
Draft Front-End Review
Due Tuesday 2/23

HW 5 Front-End Web Application

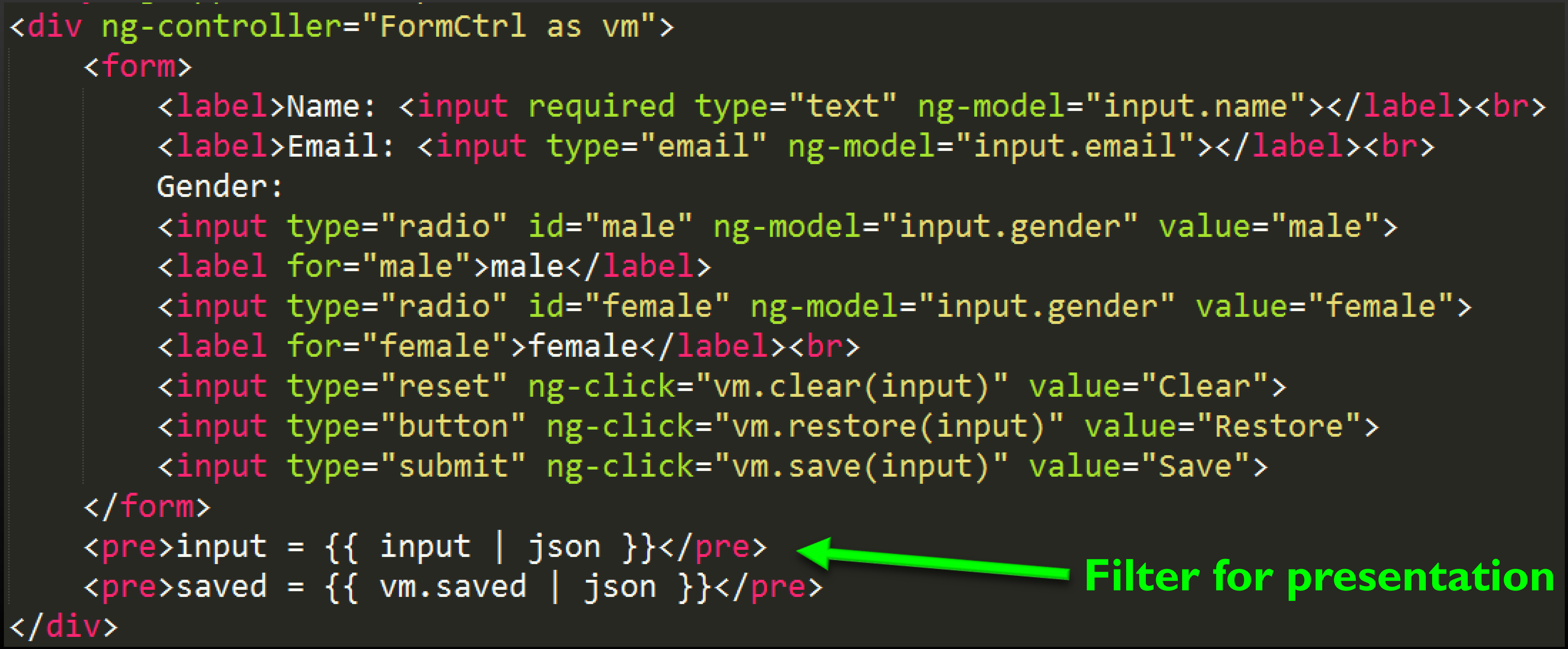
- Implement:
  - User login / logout – routing from landing to main to profile views
  - Status headline update
  - Filter the posts by author/body (not id)
  - Add a post
  - Edit a post
  - Comment on a post
  - Edit a comment
- TDD = *Write tests first!*
- Using the Dummy Server (next week)

https://www.clear.rice.edu/comp431/sample/angular-form.html

Form Validation

DEMO

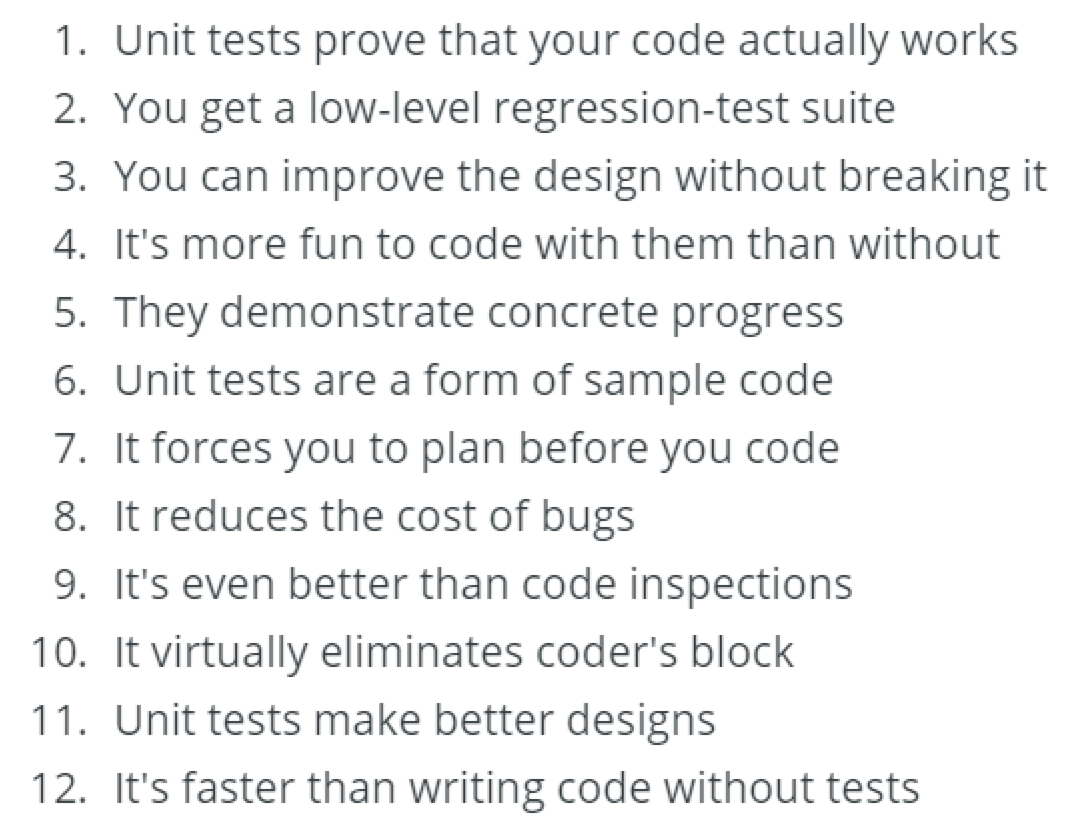
- Angular binding is “smart” about forms



```
<div ng-controller="FormCtrl as vm">
    <form>
        <label>Name: <input required type="text" ng-model="input.name"></label><br>
        <label>Email: <input type="email" ng-model="input.email"></label><br>
        Gender:
        <input type="radio" id="male" ng-model="input.gender" value="male">
        <label for="male">male</label>
        <input type="radio" id="female" ng-model="input.gender" value="female">
        <label for="female">female</label><br>
        <input type="reset" ng-click="vm.clear(input)" value="Clear">
        <input type="button" ng-click="vm.restore(input)" value="Restore">
        <input type="submit" ng-click="vm.save(input)" value="Save">
    </form>
    <pre>input = {{ input | json }}</pre>
    <pre>saved = {{ vm.saved | json }}</pre>
</div>
```

Filter for presentation

Testing



1. Unit tests prove that your code actually works
2. You get a low-level regression-test suite
3. You can improve the design without breaking it
4. It's more fun to code with them than without
5. They demonstrate concrete progress
6. Unit tests are a form of sample code
7. It forces you to plan before you code
8. It reduces the cost of bugs
9. It's even better than code inspections
10. It virtually eliminates coder's block
11. Unit tests make better designs
12. It's faster than writing code without tests

CODING HORROR

programming and human factors

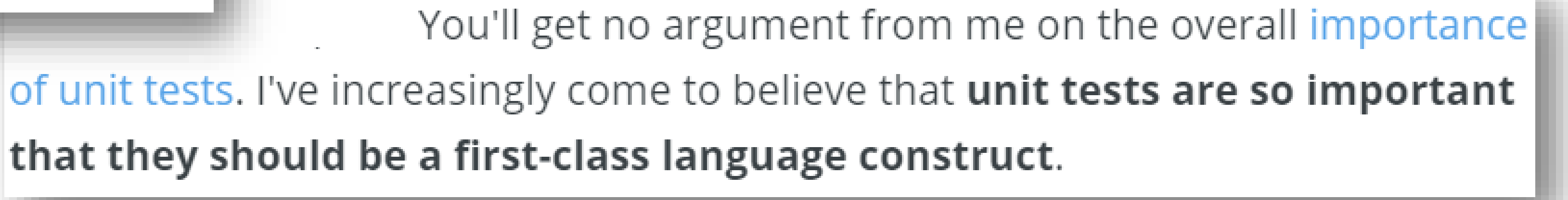
Copyright Jeff Atwood © 2015

Logo image © 1993 Steven C. McConnell

20 Jul 2006

I Pity The Fool Who Doesn't Write Unit Tests

J. Timothy King has a nice piece on the twelve benefits of writing unit tests first.



You'll get no argument from me on the overall importance of unit tests. I've increasingly come to believe that **unit tests are so important that they should be a first-class language construct**.

http://blog.codinghorror.com/i-pity-the-fool-who-doesnt-write-unit-tests/

http://www.jtse.com/blog/2006/07/11/twelve-benefits-of-writing-unit-tests-first

Pieces of Testing

- Assertion
  - A single condition expected to be true
- Test Case
  - Executes an atomic component of the larger program
  - May contain multiple assertions
- Test Suite
  - Collection of Test Cases
- Test Runner
  - Executes the test suite

Testing Frameworks

- Jasmine
  - Assertion
  - "Describe it" for cases
  - Suite = all in file
  - Runner in page

state of the art

Mocha (test)
Chai (assertion)
Sinon (mock)

- QUnit (jQuery)
- Jest
- JsUnit
- Buster.js
- TestSwarm
- BrowserSwarm
- Intern
- Chutzpah
- Dojo Object Harness

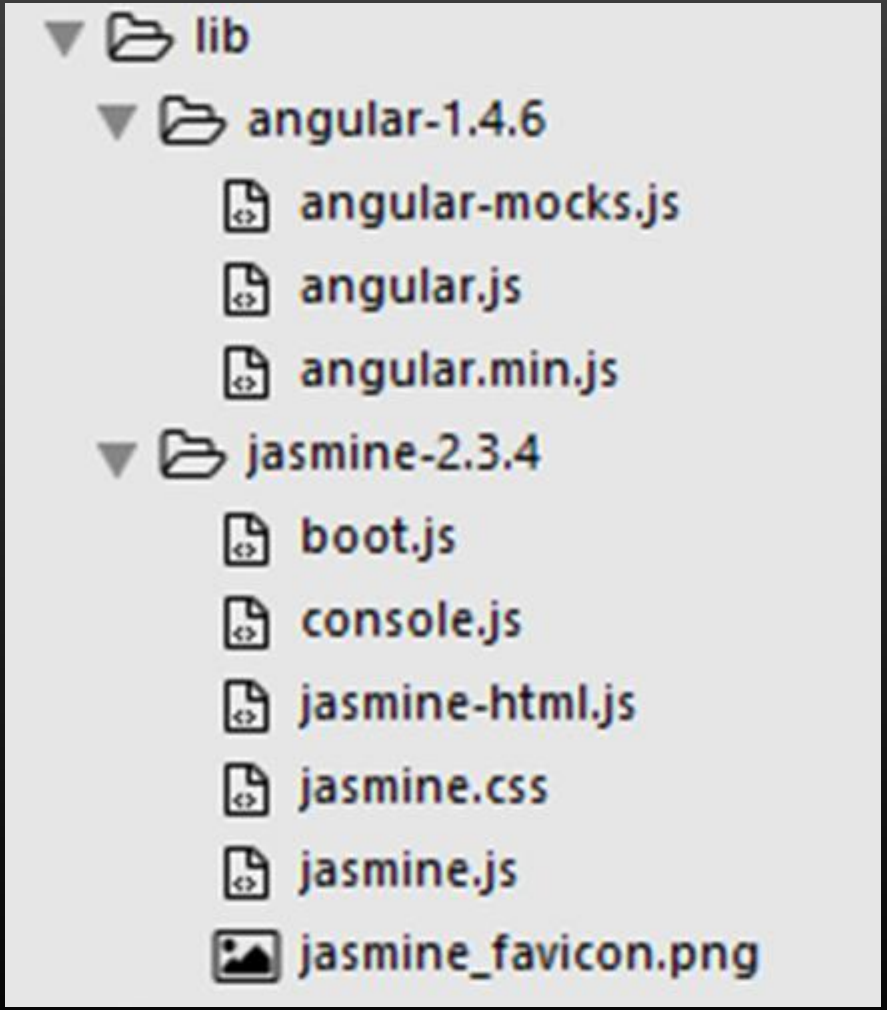
- Pavlov
- jsTestDriver
- HtmlUnit
- Celerity
- Schnell
- Screw.Unit

Put libraries in lib

- Angular v1.4.6 (or 1.5.0)
  https://code.angularjs.org/1.4.6/angular.js

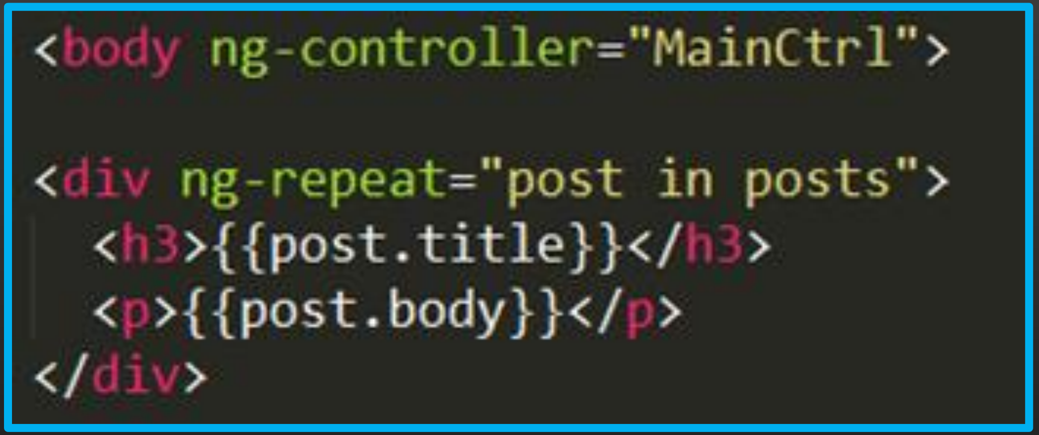
- Test Driven Development with Jasmine v2.3.4
  https://github.com/jasmine/jasmine/releases

- For mocking services next week
  https://code.angularjs.org/1.4.6/angular-mocks.js

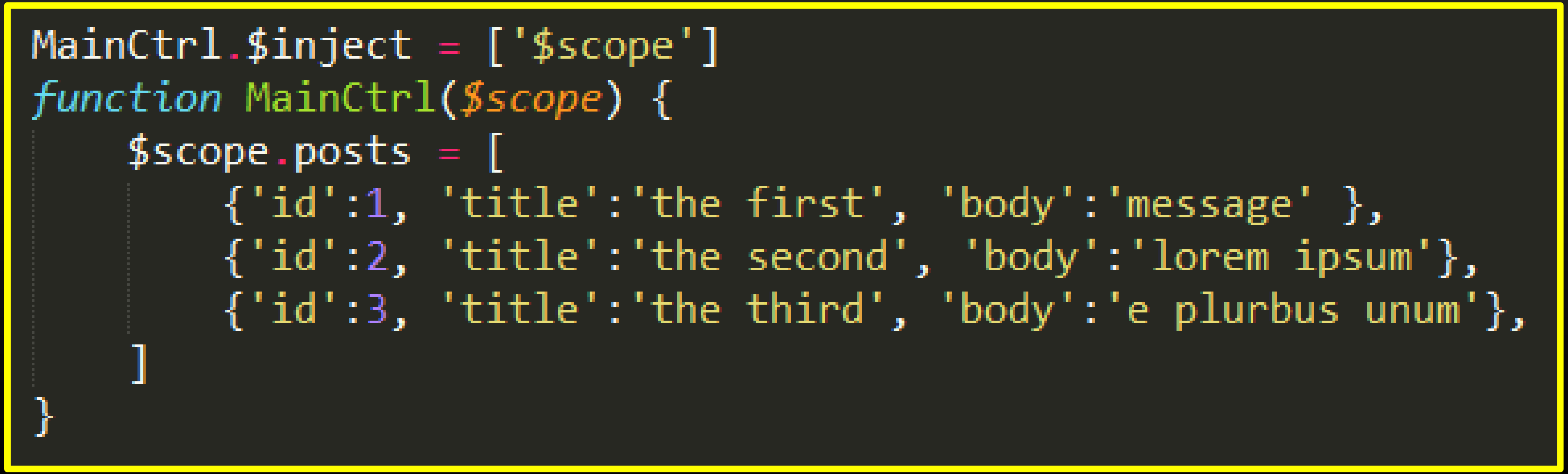


- lib
  - angular-1.4.6
    - angular-mocks.js
    - angular.js
    - angular.min.js
  - jasmine-2.3.4
    - boot.js
    - console.js
    - jasmine-html.js
    - jasmine.css
    - jasmine.js
    - jasmine_favicon.png

hello-angular

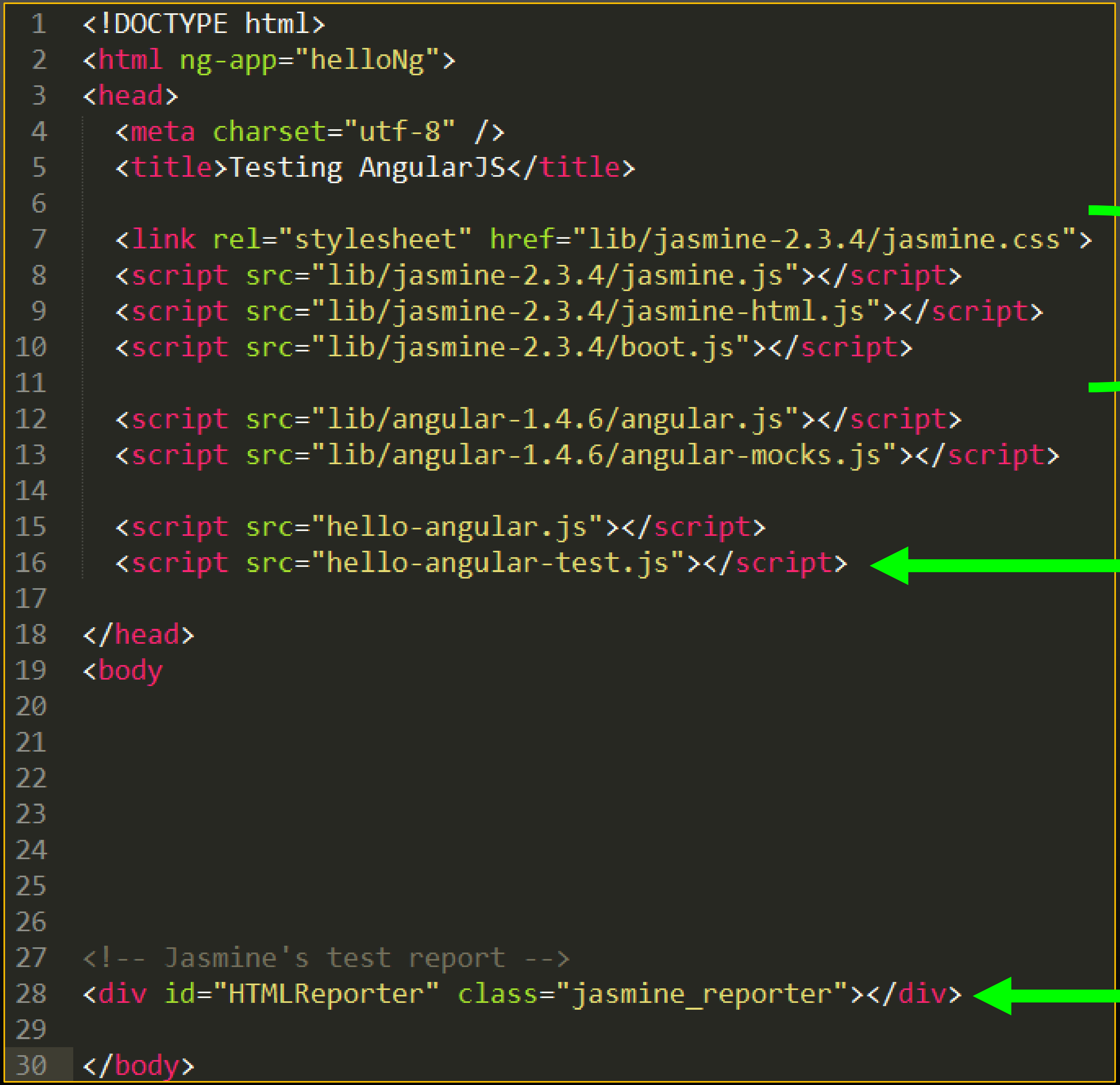


```
<body ng-controller="MainCtrl">

<div ng-repeat="post in posts">
  <h3>{{post.title}}</h3>
  <p>{{post.body}}</p>
</div>
```



```
MainCtrl.$inject = ['$scope']
function MainCtrl($scope) {
    $scope.posts = [
        {'id':1, 'title':'the first', 'body':'message' },
        {'id':2, 'title':'the second', 'body':'lorem ipsum'},
        {'id':3, 'title':'the third', 'body':'e plurbus unum'},
    ]
}
```

Testing with Jasmine and ngMocks



```html
<!DOCTYPE html>
<html ng-app="helloNg">
<head>
  <meta charset="utf-8" />
  <title>Testing AngularJS</title>

  <link rel="stylesheet" href="lib/jasmine-2.3.4/jasmine.css">
  <script src="lib/jasmine-2.3.4/jasmine.js"></script>
  <script src="lib/jasmine-2.3.4/jasmine-html.js"></script>
  <script src="lib/jasmine-2.3.4/boot.js"></script>

  <script src="lib/angular-1.4.6/angular.js"></script>
  <script src="lib/angular-1.4.6/angular-mocks.js"></script>

  <script src="hello-angular.js"></script>
  <script src="hello-angular-test.js"></script>

</head>
<body









<!-- Jasmine's test report -->
<div id="HTMLReporter" class="jasmine_reporter"></div>

</body>
```

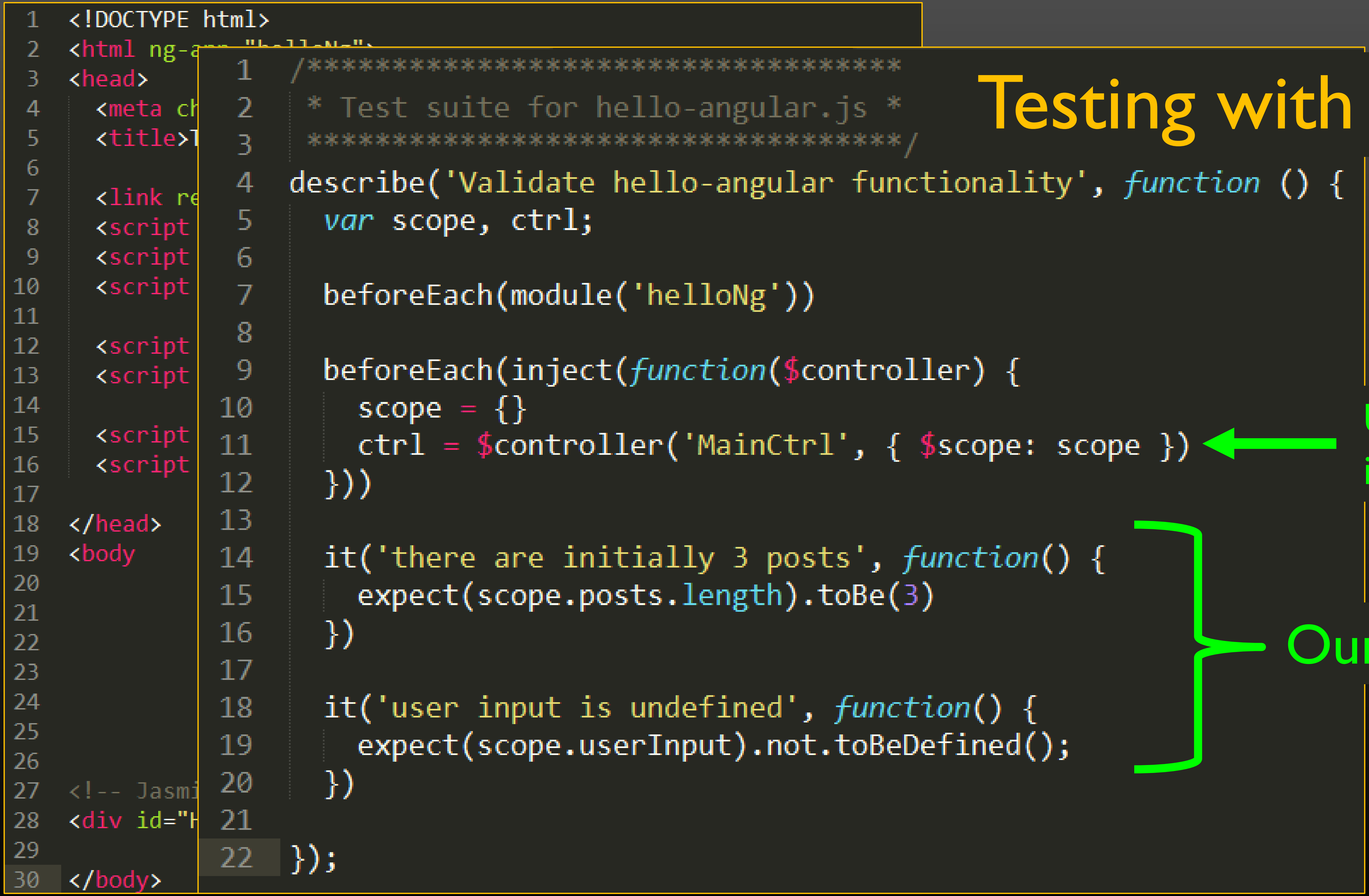
Jasmine libraries

Our test definitions

hello-angular.spec.js

Jasmine output

Testing with Jasmine



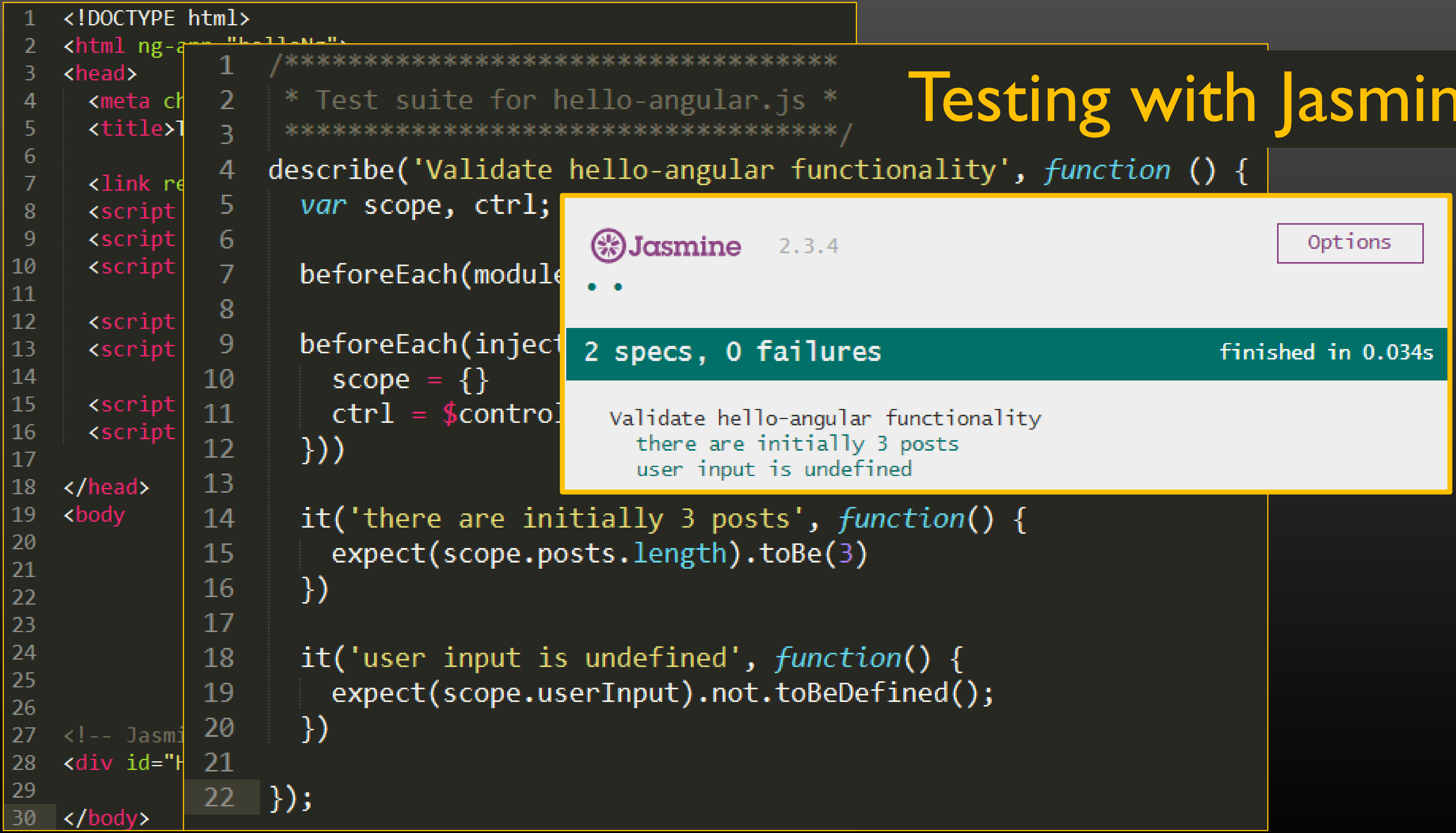
```
<!DOCTYPE html>
<html ng-a
<head>
  <meta ch
  <title>T

  <link re
  <script
  <script
  <script

  <script
  <script

  <script
  <script

</head>
<body>







<!-- Jasmi
<div id="H

</body>
```

```
/***********************************
 * Test suite for hello-angular.js *
 ***********************************/
describe('Validate hello-angular functionality', function () {
  var scope, ctrl;

  beforeEach(module('helloNg'))

  beforeEach(inject(function($controller) {
    scope = {}
    ctrl = $controller('MainCtrl', { $scope: scope })
  }))

  it('there are initially 3 posts', function() {
    expect(scope.posts.length).toBe(3)
  })

  it('user input is undefined', function() {
    expect(scope.userInput).not.toBeDefined();
  })

});
```

Using the $scope injected version

Our tests

Testing with Jasmine

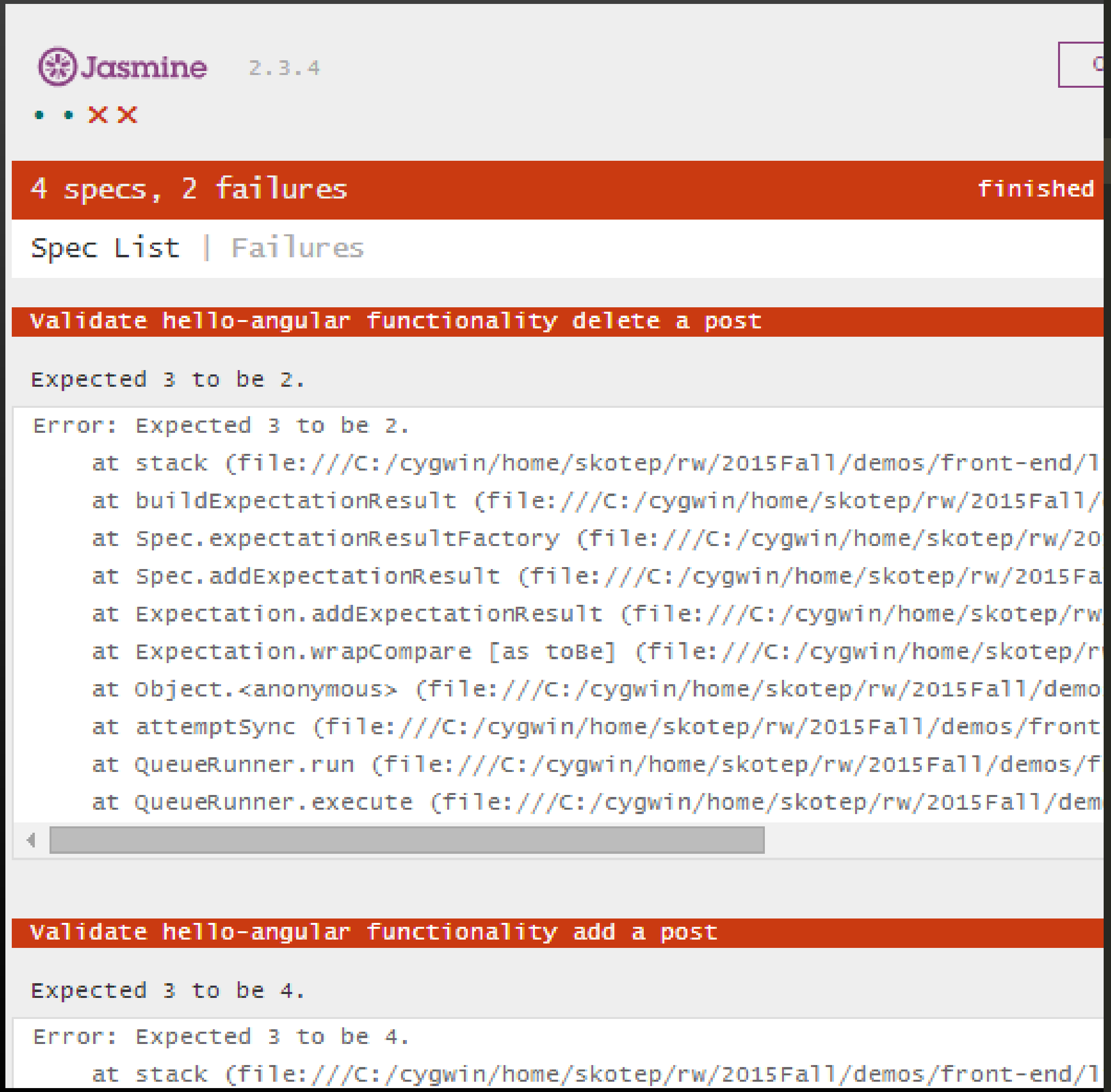


```
<!DOCTYPE html>
<html ng-a
<head>
  <meta ch
  <title>T

  <link re
  <script
  <script
  <script

  <script
  <script

  <script
  <script

</head>
<body







<!-- Jasmi
<div id="H

</body>
```

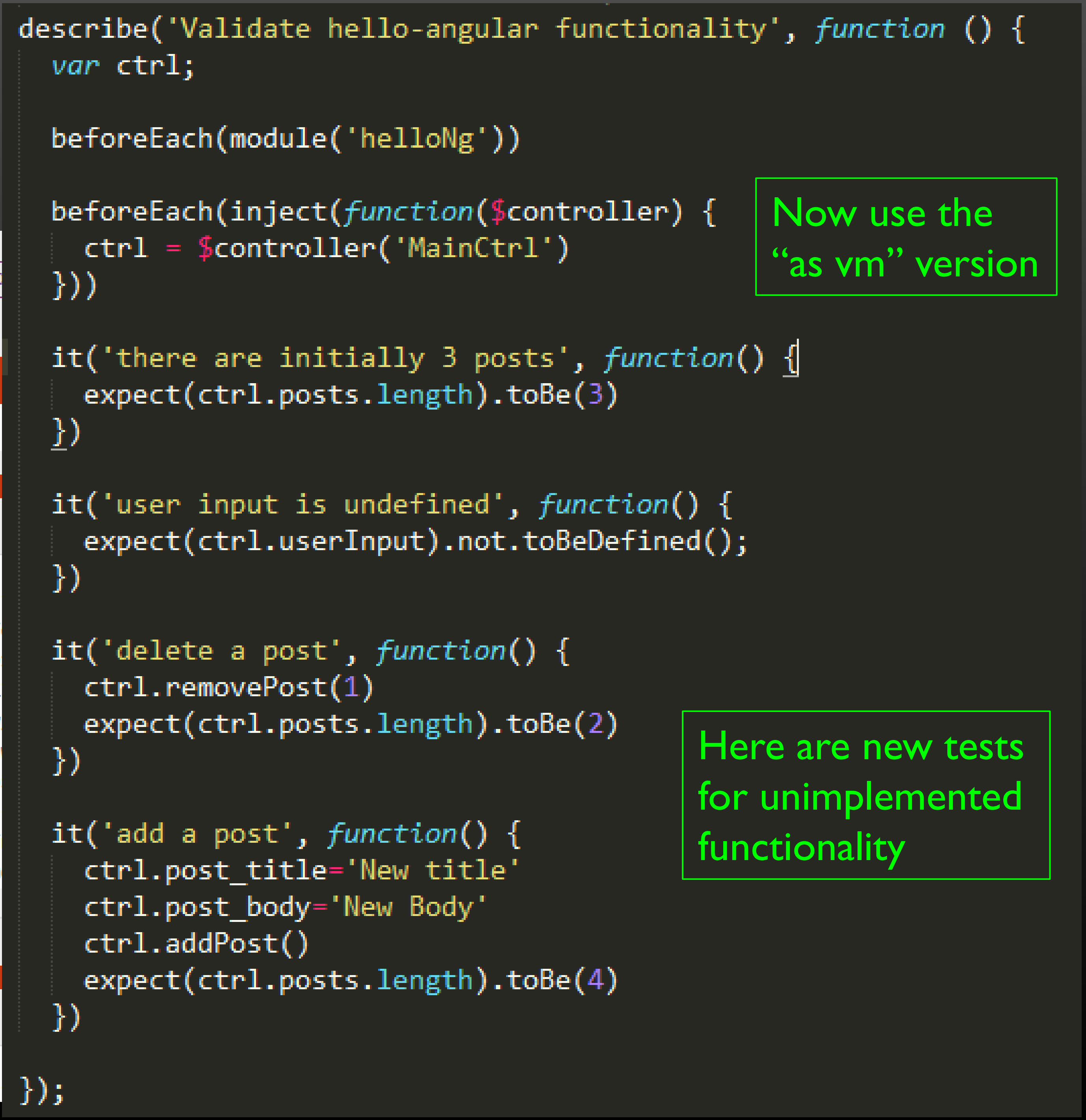
```
/************************************
 * Test suite for hello-angular.js *
 ************************************/
describe('Validate hello-angular functionality', function () {
  var scope, ctrl;

  beforeEach(module

  beforeEach(inject
    scope = {}
    ctrl = $control
  }))

  it('there are initially 3 posts', function() {
    expect(scope.posts.length).toBe(3)
  })

  it('user input is undefined', function() {
    expect(scope.userInput).not.toBeDefined();
  })

});
```

```
• •
2 specs, 0 failures                    finished in 0.034s

  Validate hello-angular functionality
    there are initially 3 posts
    user input is undefined
```

Let’s do some TDD



```
4 specs, 2 failures                                          finished

Spec List | Failures

Validate hello-angular functionality delete a post

Expected 3 to be 2.

Error: Expected 3 to be 2.
    at stack (file:///C:/cygwin/home/skotep/rw/2015Fall/demos/front-end/l
    at buildExpectationResult (file:///C:/cygwin/home/skotep/rw/2015Fall/
    at Spec.expectationResultFactory (file:///C:/cygwin/home/skotep/rw/20
    at Spec.addExpectationResult (file:///C:/cygwin/home/skotep/rw/2015Fa
    at Expectation.addExpectationResult (file:///C:/cygwin/home/skotep/rw
    at Expectation.wrapCompare [as toBe] (file:///C:/cygwin/home/skotep/r
    at Object.<anonymous> (file:///C:/cygwin/home/skotep/rw/2015Fall/demo
    at attemptSync (file:///C:/cygwin/home/skotep/rw/2015Fall/demos/front
    at QueueRunner.run (file:///C:/cygwin/home/skotep/rw/2015Fall/demos/f
    at QueueRunner.execute (file:///C:/cygwin/home/skotep/rw/2015Fall/dem

Validate hello-angular functionality add a post

Expected 3 to be 4.

Error: Expected 3 to be 4.
    at stack (file:///C:/cygwin/home/skotep/rw/2015Fall/demos/front-end/l
```

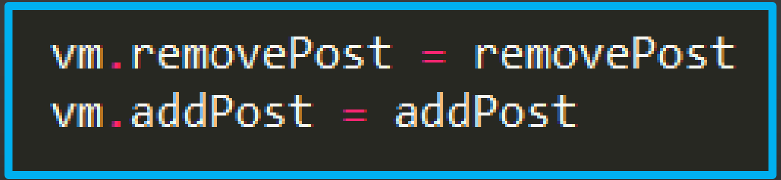


```javascript
describe('Validate hello-angular functionality', function () {
  var ctrl;

  beforeEach(module('helloNg'))

  beforeEach(inject(function($controller) {
    ctrl = $controller('MainCtrl')
  }))

  it('there are initially 3 posts', function() {
    expect(ctrl.posts.length).toBe(3)
  })

  it('user input is undefined', function() {
    expect(ctrl.userInput).not.toBeDefined();
  })

  it('delete a post', function() {
    ctrl.removePost(1)
    expect(ctrl.posts.length).toBe(2)
  })

  it('add a post', function() {
    ctrl.post_title='New title'
    ctrl.post_body='New Body'
    ctrl.addPost()
    expect(ctrl.posts.length).toBe(4)
  })

});
```

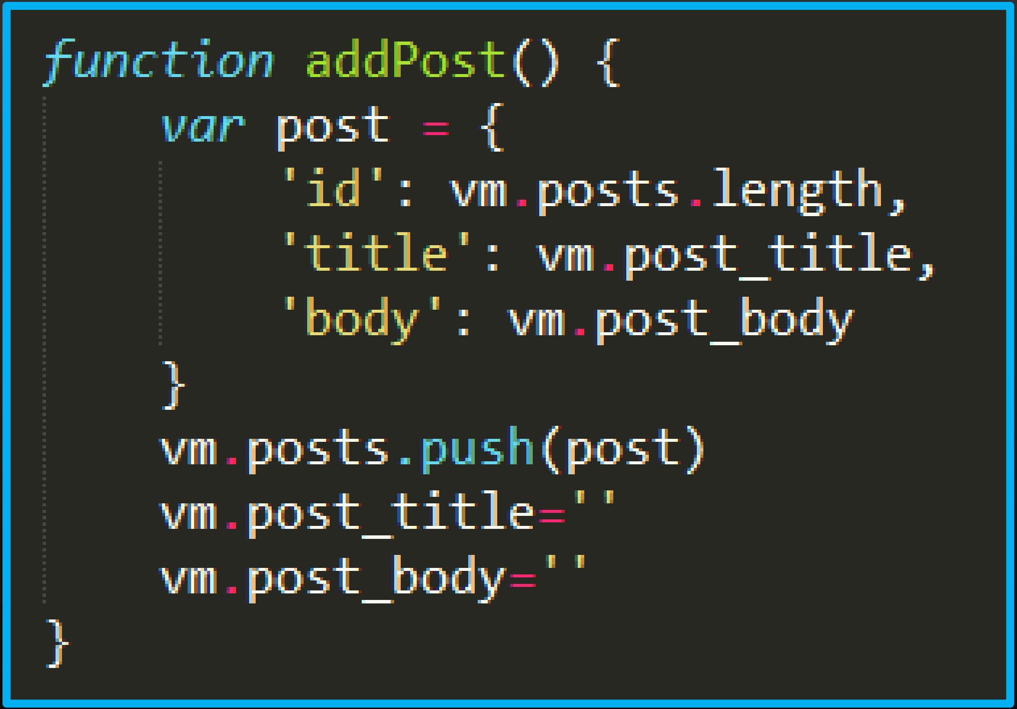
Now use the “as vm” version

Here are new tests for unimplemented functionality

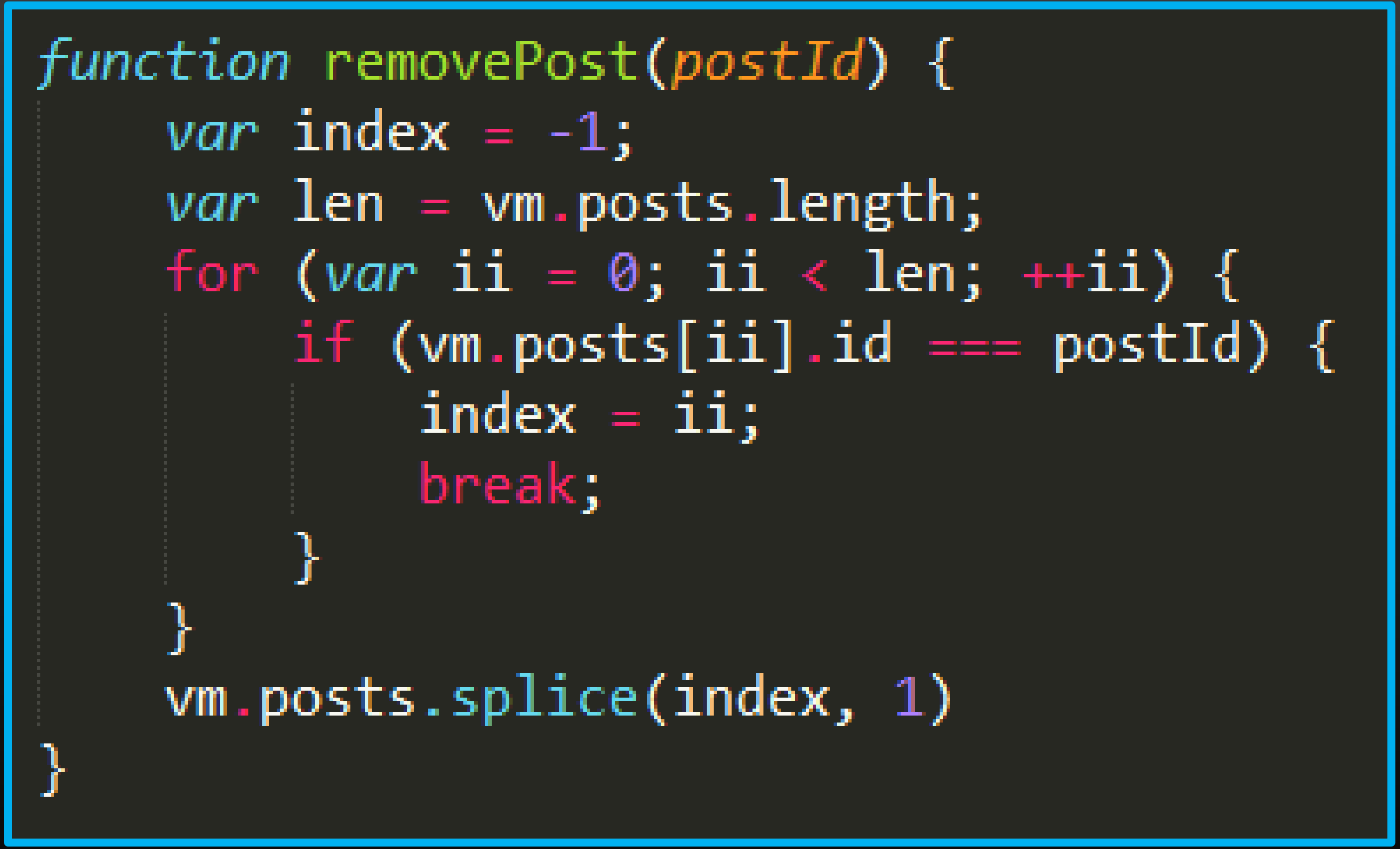
Implement the functionality



```
vm.removePost = removePost
vm.addPost = addPost
```

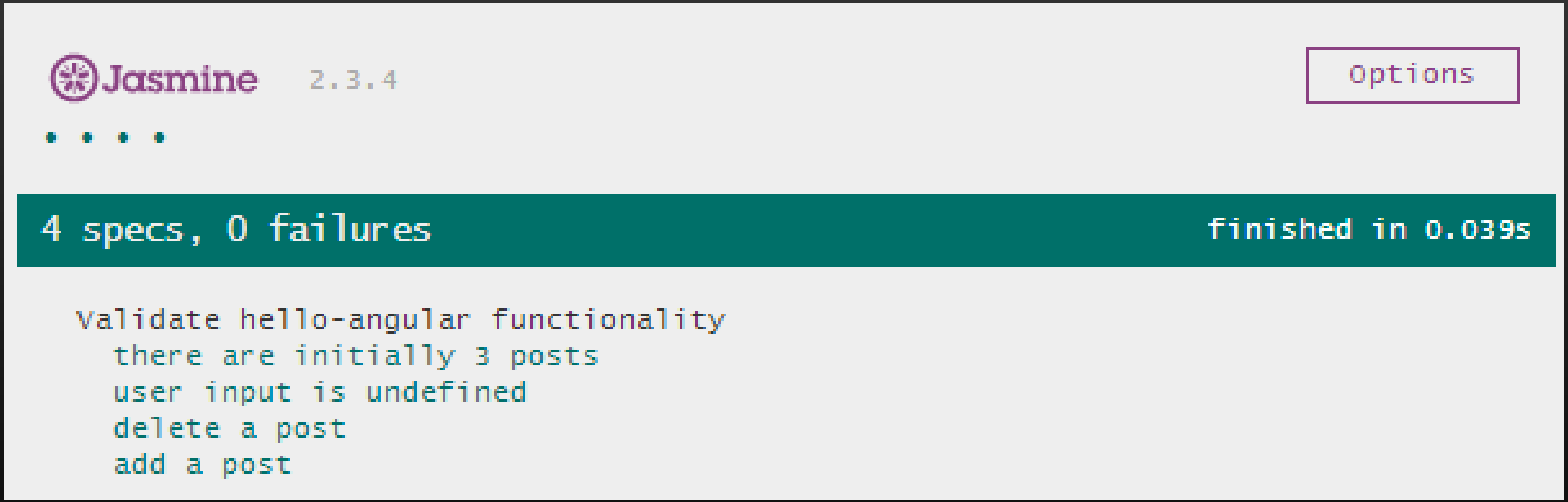


```
function addPost() {
    var post = {
        'id': vm.posts.length,
        'title': vm.post_title,
        'body': vm.post_body
    }
    vm.posts.push(post)
    vm.post_title=''
    vm.post_body=''
}
```



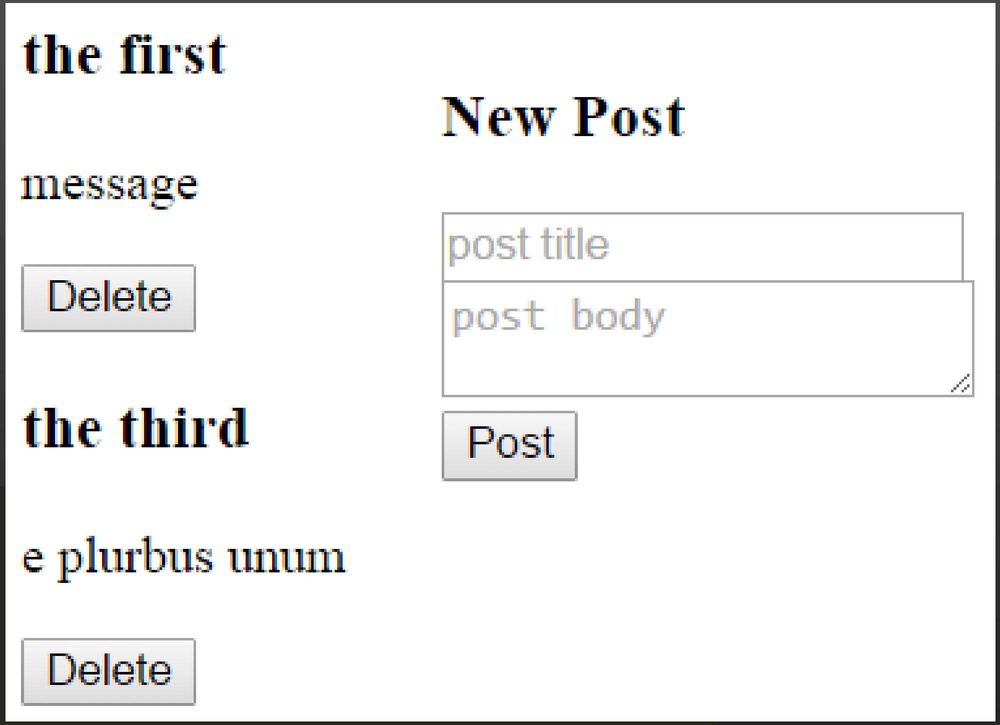
```
function removePost(postId) {
    var index = -1;
    var len = vm.posts.length;
    for (var ii = 0; ii < len; ++ii) {
        if (vm.posts[ii].id === postId) {
            index = ii;
            break;
        }
    }
    vm.posts.splice(index, 1)
}
```

Tests Pass



```
4 specs, 0 failures                                   finished in 0.039s

  Validate hello-angular functionality
    there are initially 3 posts
    user input is undefined
    delete a post
    add a post
```

Now update the View



the first

message

Delete

the third

e plurbus unum

Delete

New Post

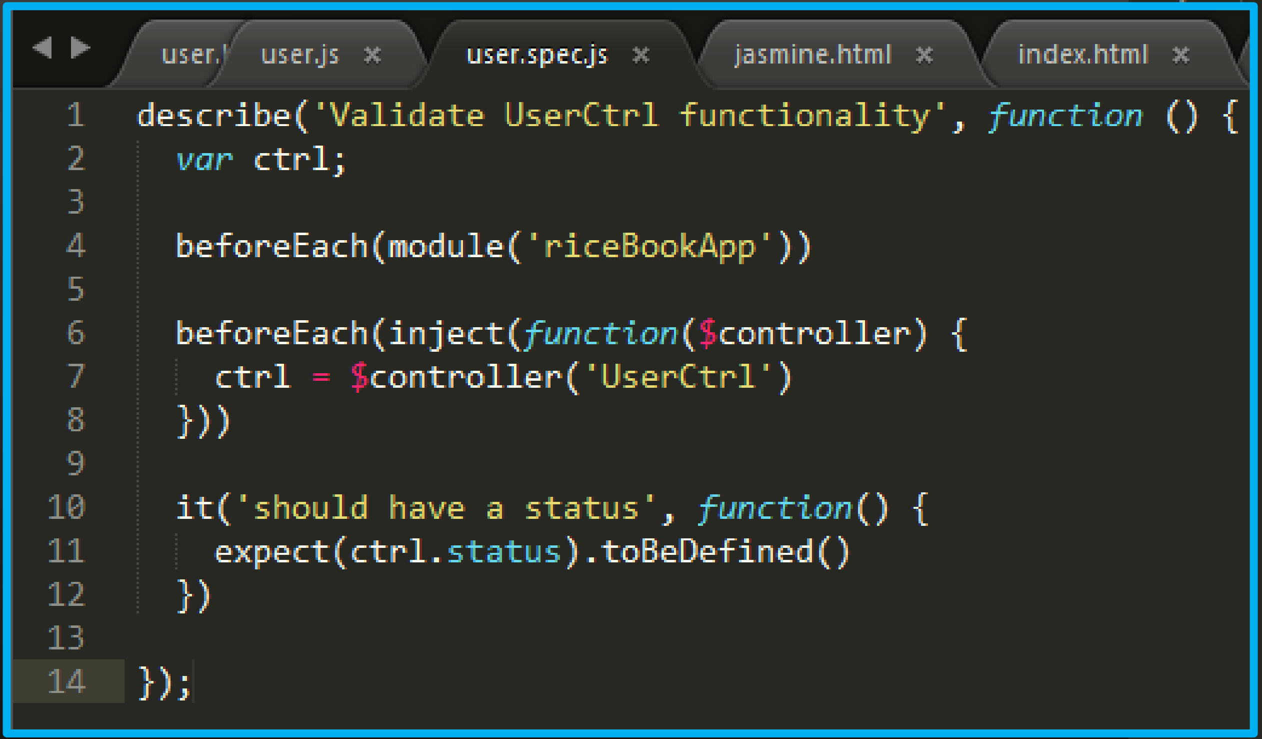
post title

post body

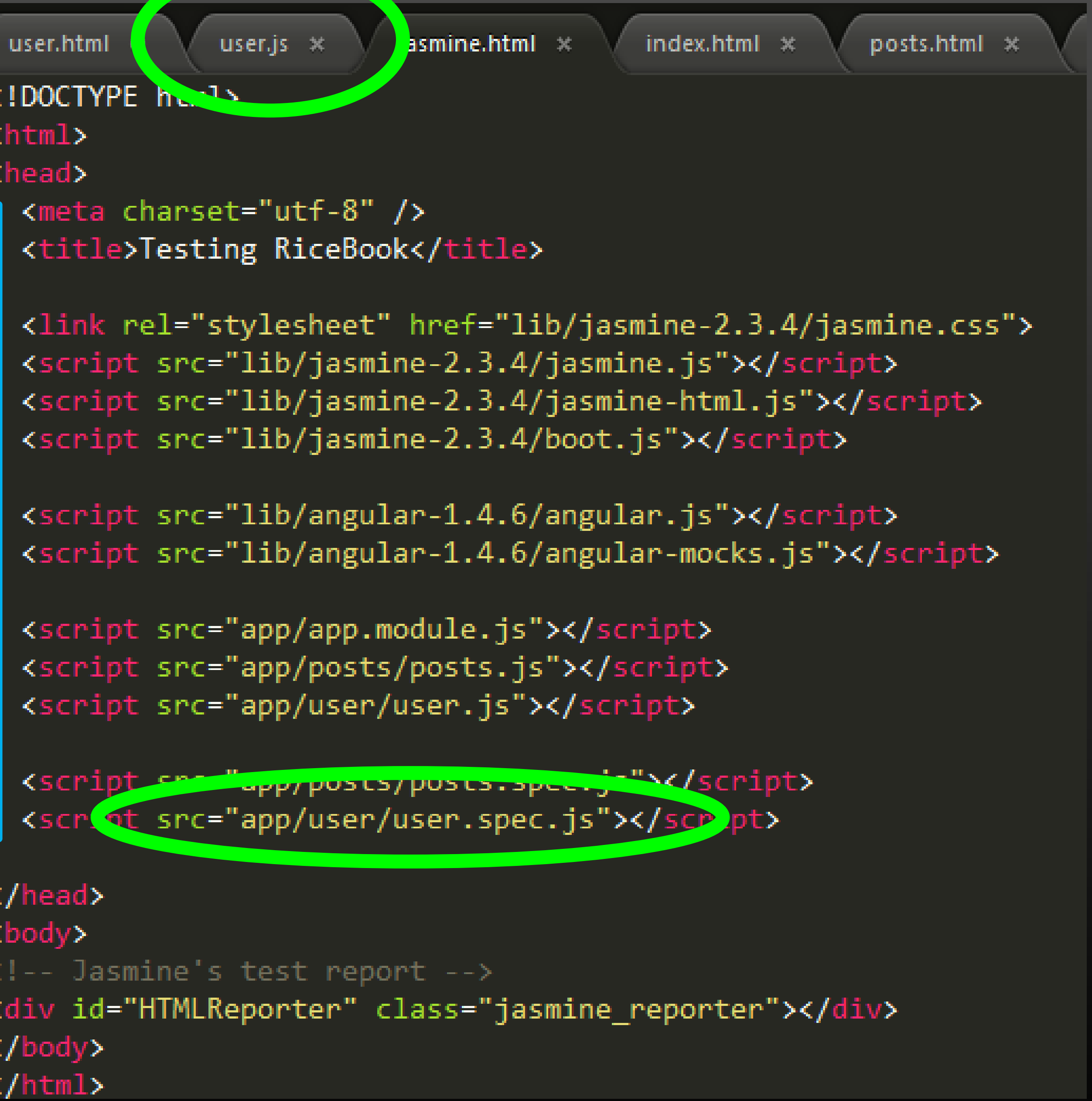
Post

```
<body ng-controller="MainCtrl as vm">

<div style="float:right">
<h3>New Post</h3>
  <input type="text" placeholder="post title" ng-model="vm.post_title"><br>
  <textarea placeholder="post body" ng-model="vm.post_body"></textarea><br>
  <input type="button" value="Post" ng-click="vm.addPost()">
</div>

<div ng-repeat="post in vm.posts">
  <h3>{{post.title}}</h3>
  <p>{{post.body}}</p>
  <input type="button" value="Delete" ng-click="vm.removePost(post.id)">
</div>
```

More Testing!



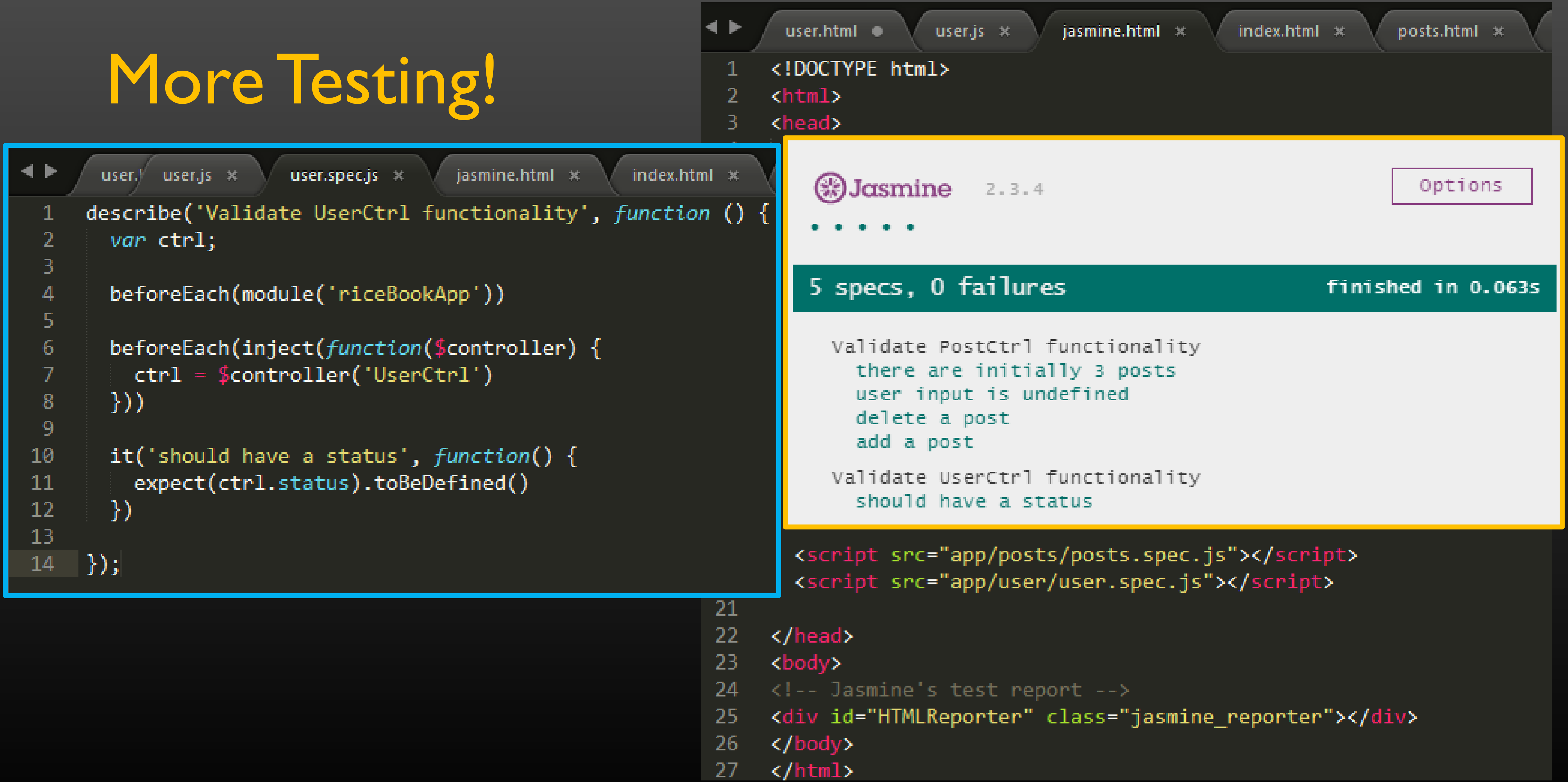
```
describe('Validate UserCtrl functionality', function () {
  var ctrl;

  beforeEach(module('riceBookApp'))

  beforeEach(inject(function($controller) {
    ctrl = $controller('UserCtrl')
  }))

  it('should have a status', function() {
    expect(ctrl.status).toBeDefined()
  })

});
```



```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <title>Testing RiceBook</title>

  <link rel="stylesheet" href="lib/jasmine-2.3.4/jasmine.css">
  <script src="lib/jasmine-2.3.4/jasmine.js"></script>
  <script src="lib/jasmine-2.3.4/jasmine-html.js"></script>
  <script src="lib/jasmine-2.3.4/boot.js"></script>

  <script src="lib/angular-1.4.6/angular.js"></script>
  <script src="lib/angular-1.4.6/angular-mocks.js"></script>

  <script src="app/app.module.js"></script>
  <script src="app/posts/posts.js"></script>
  <script src="app/user/user.js"></script>

  <script src="app/posts/posts.spec.js"></script>
  <script src="app/user/user.spec.js"></script>

</head>
<body>
<!-- Jasmine's test report -->
<div id="HTMLReporter" class="jasmine_reporter"></div>
</body>
</html>
```

Call this “StatusCtrl”
not “UserCtrl”
put in status.js, etc

More Testing!



```
describe('Validate UserCtrl functionality', function () {
  var ctrl;

  beforeEach(module('riceBookApp'))

  beforeEach(inject(function($controller) {
    ctrl = $controller('UserCtrl')
  }))

  it('should have a status', function() {
    expect(ctrl.status).toBeDefined()
  })

});
```

```
5 specs, 0 failures                                finished in 0.063s

  Validate PostCtrl functionality
    there are initially 3 posts
    user input is undefined
    delete a post
    add a post

  Validate UserCtrl functionality
    should have a status
```

```
<!DOCTYPE html>
<html>
<head>
  <script src="app/posts/posts.spec.js"></script>
  <script src="app/user/user.spec.js"></script>

</head>
<body>
<!-- Jasmine's test report -->
<div id="HTMLReporter" class="jasmine_reporter"></div>
</body>
</html>
```

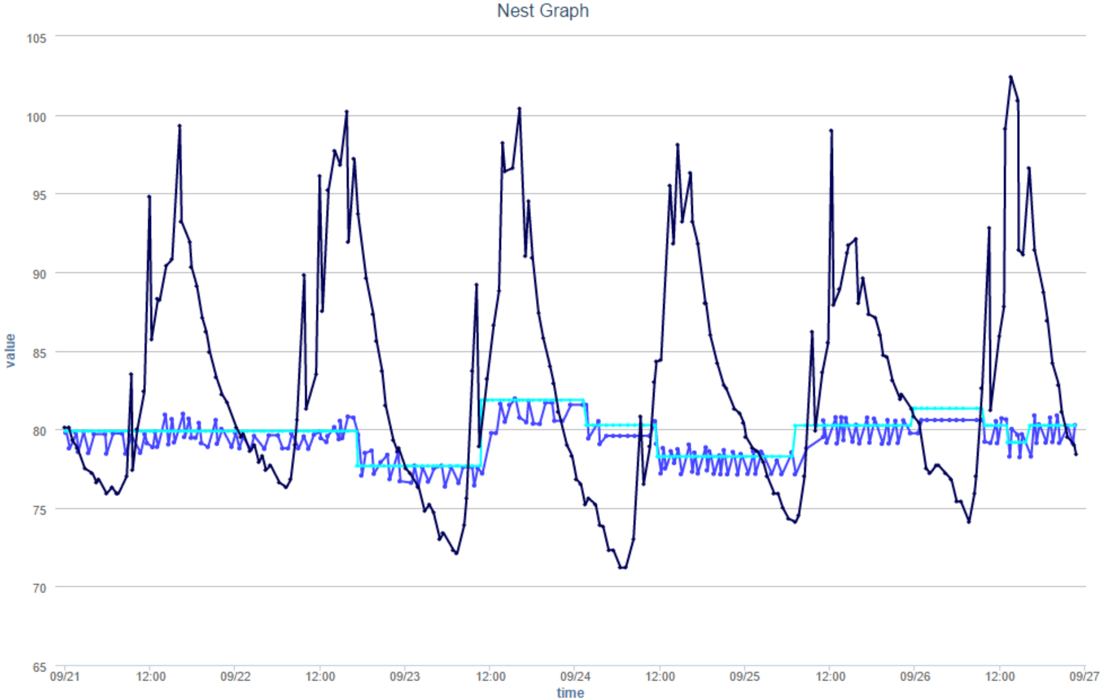
Directives

```
.directive('timeChart', function() {
  return {
    restrict: 'E',
    template: '<canvas width="{{chartWidth}}", height="{{chartHeight}}"></canvas>',
    scope: {
      chartWidth: "@",
      chartHeight: "@",
      chartOptions: '=',
      chartData: '='
    },
    replace:false,
    link: function(scope, elem, attrs) {
      var chartContext = elem.children()[0].getContext("2d");
      var chart = null;
      scope.$watch("chartData", function(v) {
        chart = d3.select(elem[0]);
        console.log(scope.chartData);
        chart.append("div").attr("class", "chart")
              .selectAll('div')
              .data(scope.chartData).enter().append("div")
              .transition().ease("elastic")
              .style("width", function(d) { return d + "%"; })
              .text(function(d) { return d + "%"; });
      });
    }
  }
})
```

Pass variables from outer scope to directive scope

This is the directive's controller

DEMO



In-Class Exercise: Testing with Jasmine

https://www.clear.rice.edu/comp431/sample/RiceBookApp-test.zip

- Test your search filter to filter by both and author, but not by date or post id
- Create a StatusCtrl and test updating the user's status headline
- If you finish these and have time, add tests for your PostCtrl (add post, edit post, add comment, etc)

http://bendetat.com/karma-and-mocha-for-angular-testing.html

Turnin* jasmine.html, spec.js, .js *files

COMP431-S16:inclass-12