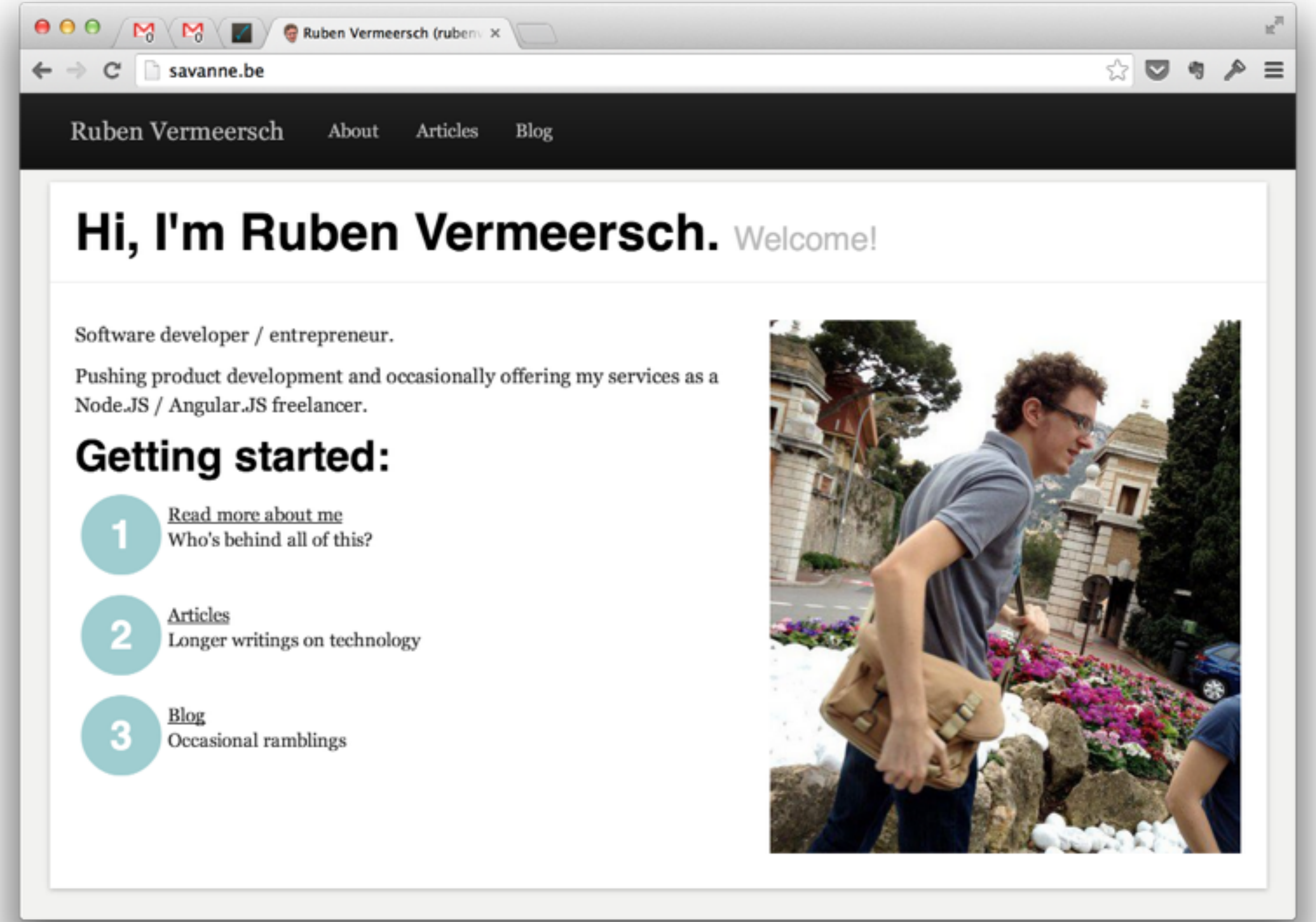
Testing with Angular.JS

Ruben Vermeersch
@rubenv
ruben@rocketeer.be

Ruben Vermeersch (@rubenv)

ruben@rocketeer.be



Hi, I'm Ruben Vermeersch. Welcome!

Software developer / entrepreneur.

Pushing product development and occasionally offering my services as a Node.JS / Angular.JS freelancer.

Getting started:

1 Read more about me
Who's behind all of this?

2 Articles
Longer writings on technology

3 Blog
Occasional ramblings

Two assumptions

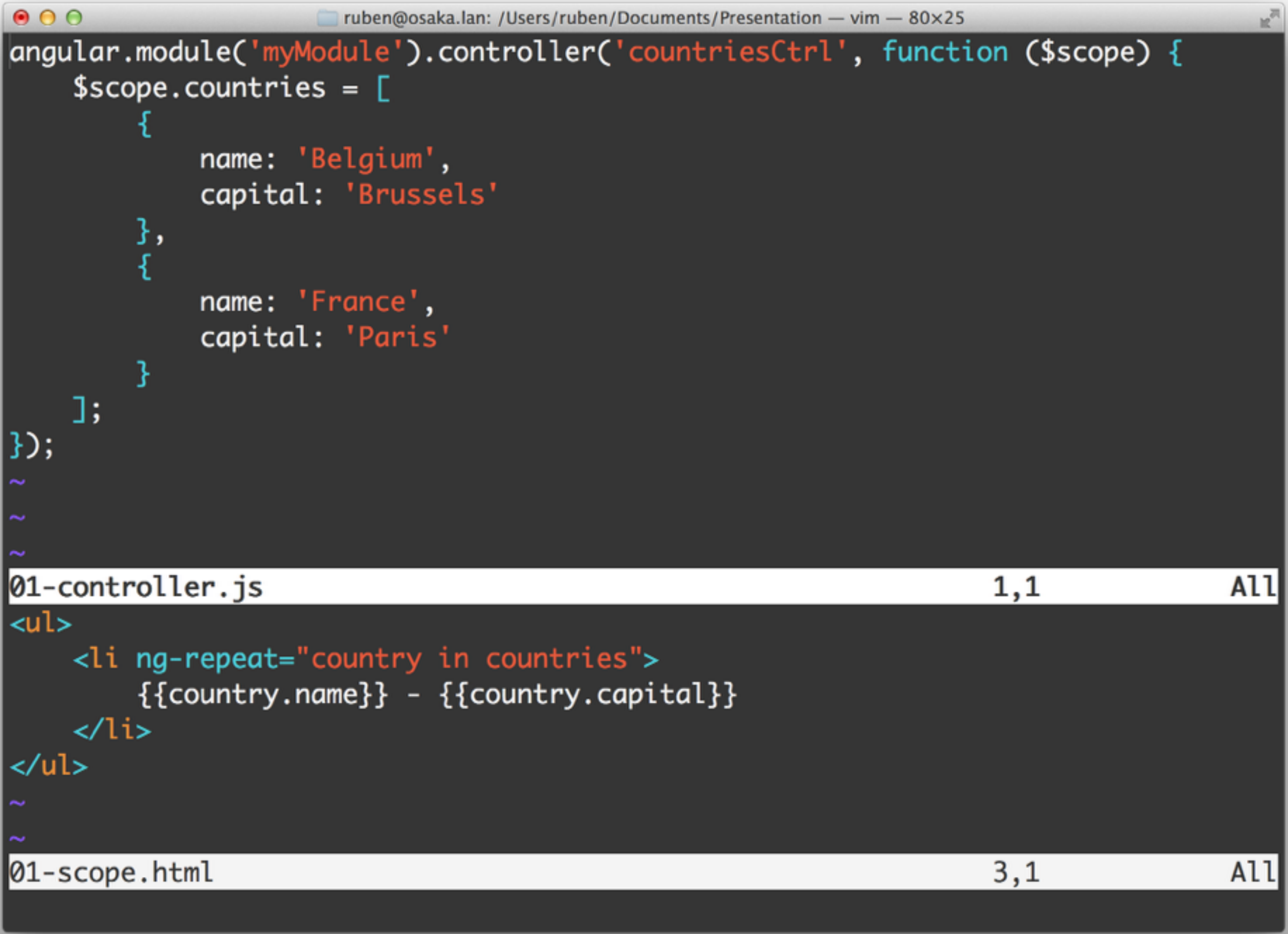
Two assumptions

- You should test more

99 little bugs in the code.
99 little bugs in the code.
Take one down, patch it around.
127 little bugs in the code...

Two assumptions

- You should test more
- You know (some) Angular.JS



ruben@osaka.lan: /Users/ruben/Documents/Presentation — vim — 80×25

```javascript
angular.module('myModule').controller('countriesCtrl', function ($scope) {
    $scope.countries = [
        {
            name: 'Belgium',
            capital: 'Brussels'
        },
        {
            name: 'France',
            capital: 'Paris'
        }
    ];
});
~
~
~
01-controller.js                                                  1,1           All
```

```html
<ul>
    <li ng-repeat="country in countries">
        {{country.name}} - {{country.capital}}
    </li>
</ul>
~
~
01-scope.html                                                     3,1           All
```

Part one: Unit testing

Part two: E2E testing

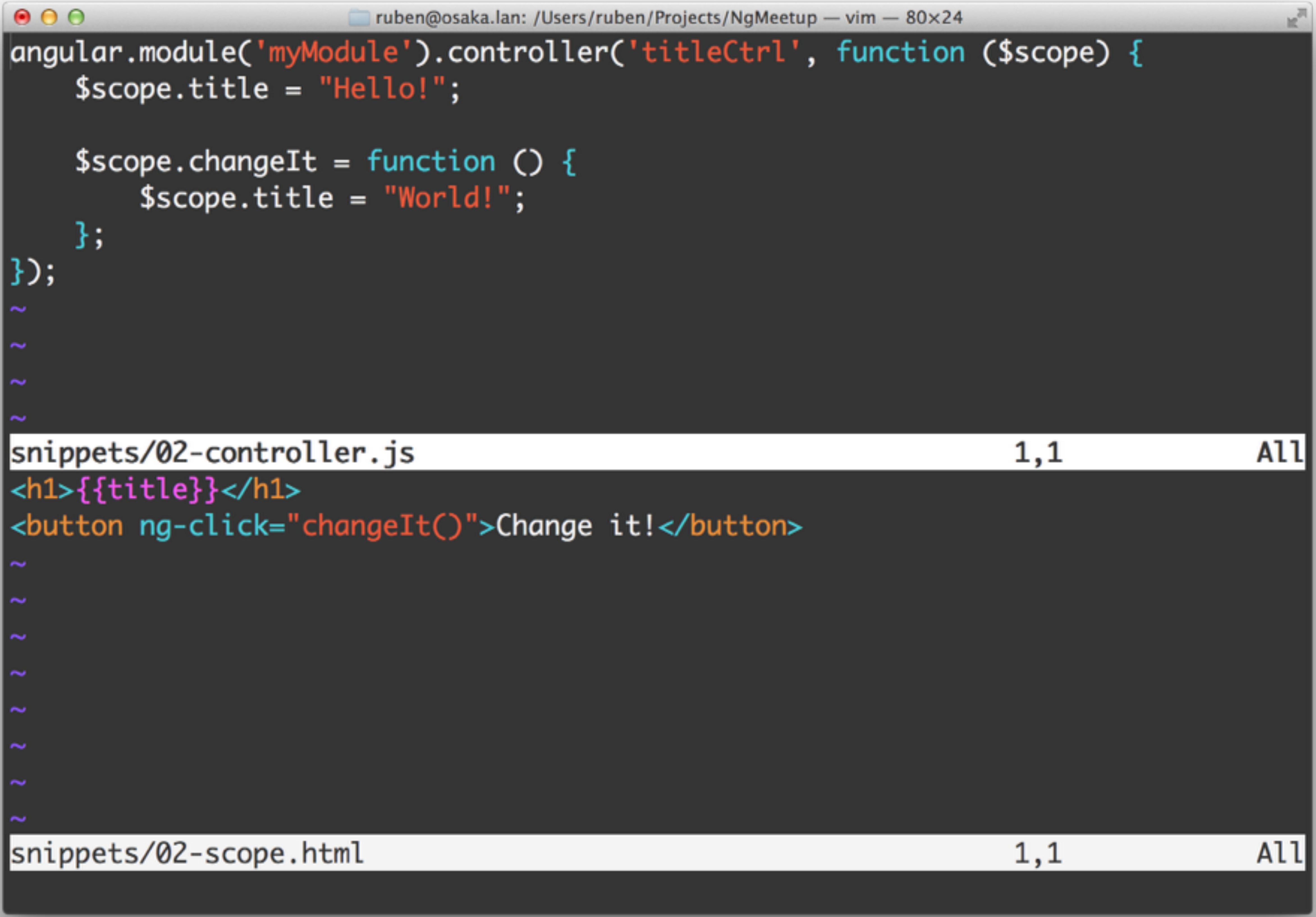
Unit vs E2E

Unit vs E2E

Unit test	E2E test
APIs	UIs
Tiny	Large
White-box	Black-box
Isolated from the world	Make the pieces fit together

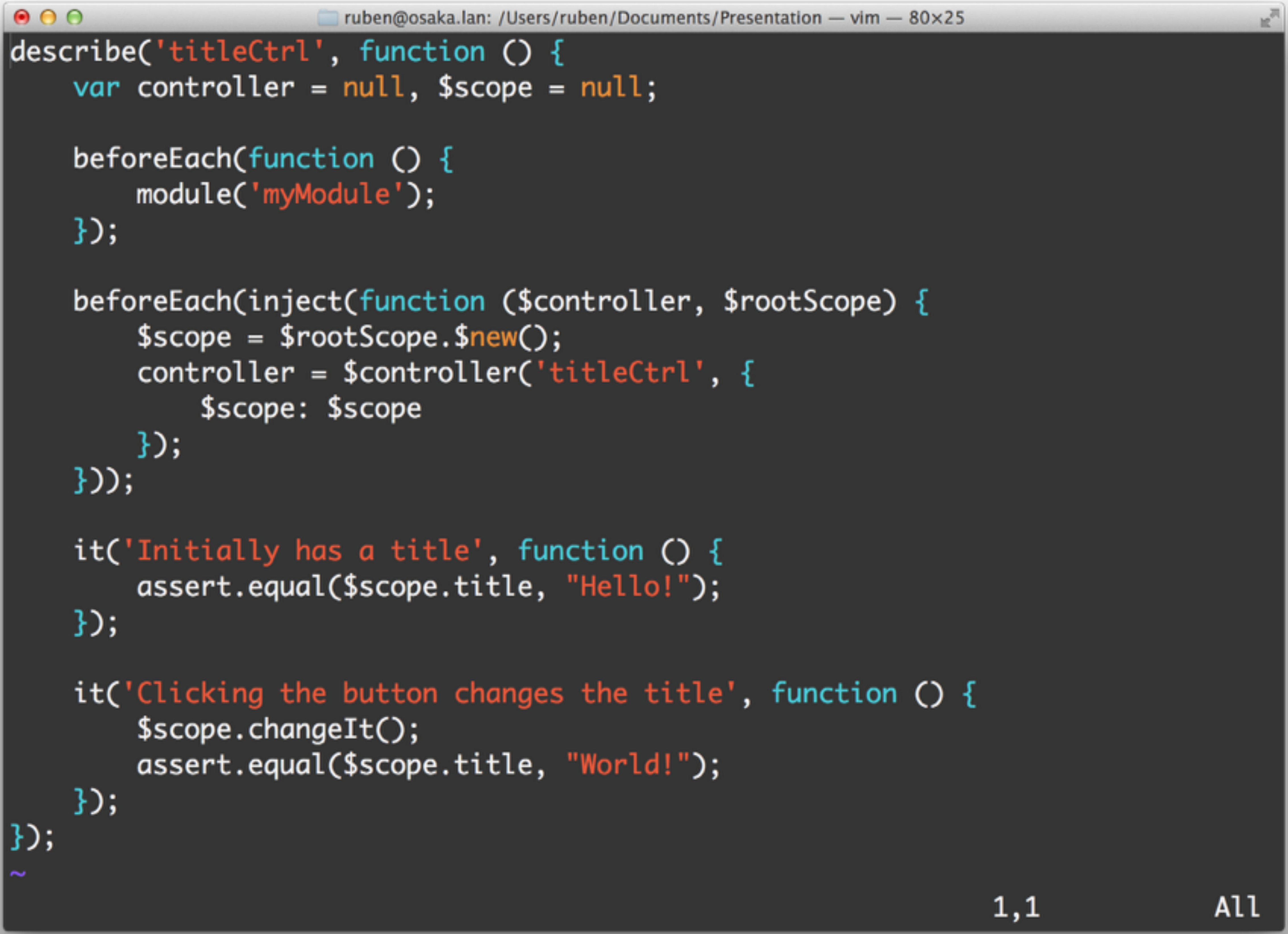
Unit testing

$scope is a wonderful thing

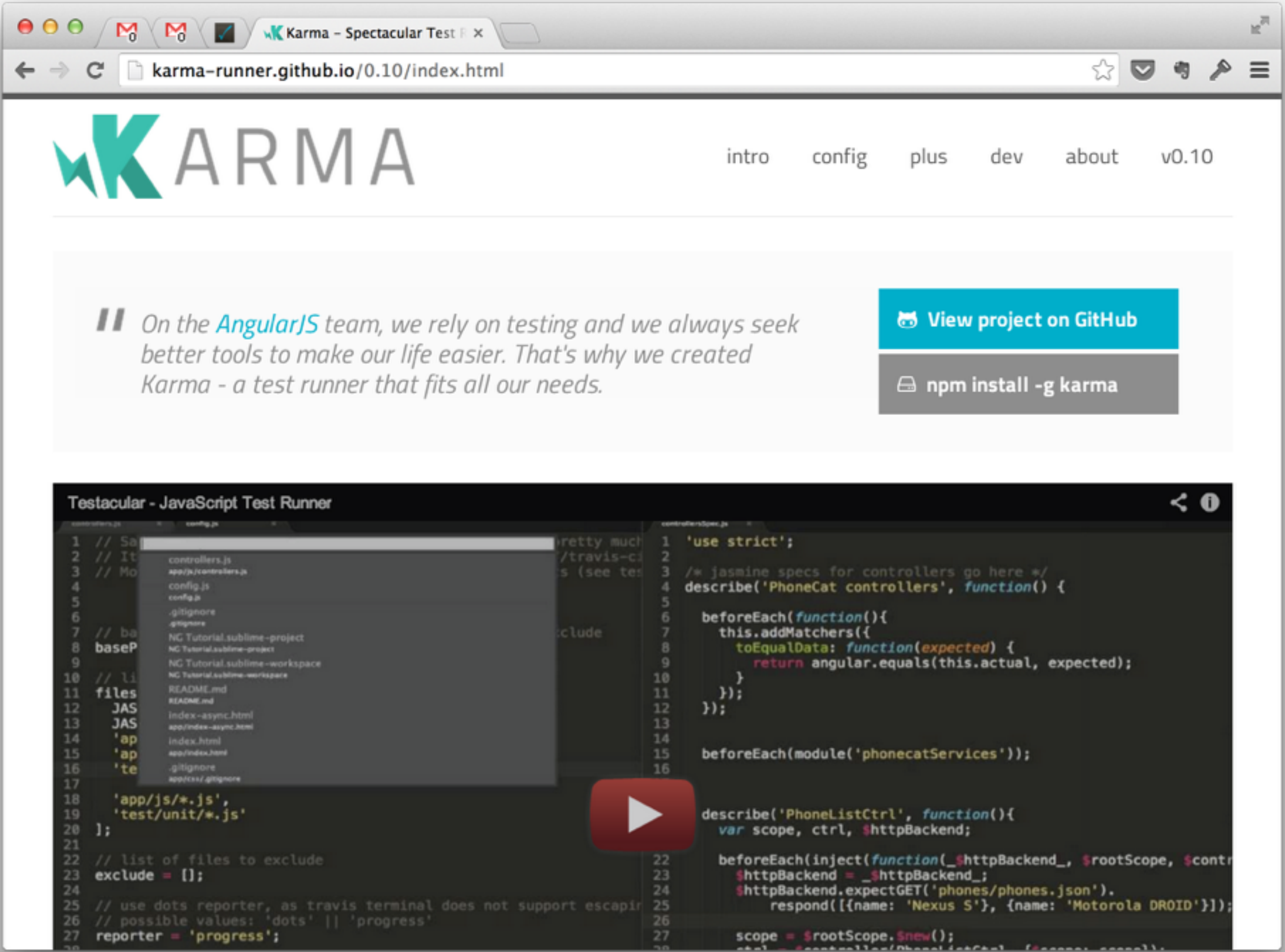

ruben@osaka.lan: /Users/ruben/Projects/NgMeetup — vim — 80×24

```
angular.module('myModule').controller('titleCtrl', function ($scope) {
    $scope.title = "Hello!";

    $scope.changeIt = function () {
        $scope.title = "World!";
    };
});
~
~
~
~
snippets/02-controller.js                                         1,1           All
<h1>{{title}}</h1>
<button ng-click="changeIt()">Change it!</button>
~
~
~
~
~
~
~
~
snippets/02-scope.html                                            1,1           All
```

Simple test case


ruben@osaka.lan: /Users/ruben/Documents/Presentation — vim — 80×25

```
describe('titleCtrl', function () {
    var controller = null, $scope = null;

    beforeEach(function () {
        module('myModule');
    });

    beforeEach(inject(function ($controller, $rootScope) {
        $scope = $rootScope.$new();
        controller = $controller('titleCtrl', {
            $scope: $scope
        });
    }));

    it('Initially has a title', function () {
        assert.equal($scope.title, "Hello!");
    });

    it('Clicking the button changes the title', function () {
        $scope.changeIt();
        assert.equal($scope.title, "World!");
    });
});
~
                                                        1,1           All
```


Karma – Spectacular Test R
karma-runner.github.io/0.10/index.html
KARMA
intro
config
plus
dev
about
v0.10
View project on GitHub
npm install -g karma
Testacular - JavaScript Test Runner
controllers.js
config.js
.gitignore
NG Tutorial.sublime-project
NG Tutorial.sublime-workspace
README.md
index-async.html
index.html
.gitignore

On the [AngularJS] team, we rely on testing and we always seek better tools to make our life easier. That's why we created Karma - a test runner that fits all our needs.

```
// Sa                                          retty much
// It                                          /travis-ci
// Mo                                          s (see tes


// ba                                          clude
baseP

// li
files
  JAS
  JAS
  'ap
  'ap
  'te

  'app/js/*.js',
  'test/unit/*.js'
];

// list of files to exclude
exclude = [];

// use dots reporter, as travis terminal does not support escapir
// possible values: 'dots' || 'progress'
reporter = 'progress';
```

```
'use strict';

/* jasmine specs for controllers go here */
describe('PhoneCat controllers', function() {

  beforeEach(function(){
    this.addMatchers({
      toEqualData: function(expected) {
        return angular.equals(this.actual, expected);
      }
    });
  });


  beforeEach(module('phonecatServices'));



  describe('PhoneListCtrl', function(){
    var scope, ctrl, $httpBackend;

    beforeEach(inject(function(_$httpBackend_, $rootScope, $contr
      $httpBackend = _$httpBackend_;
      $httpBackend.expectGET('phones/phones.json').
          respond([{name: 'Nexus S'}, {name: 'Motorola DROID'}]);

      scope = $rootScope.$new();
```

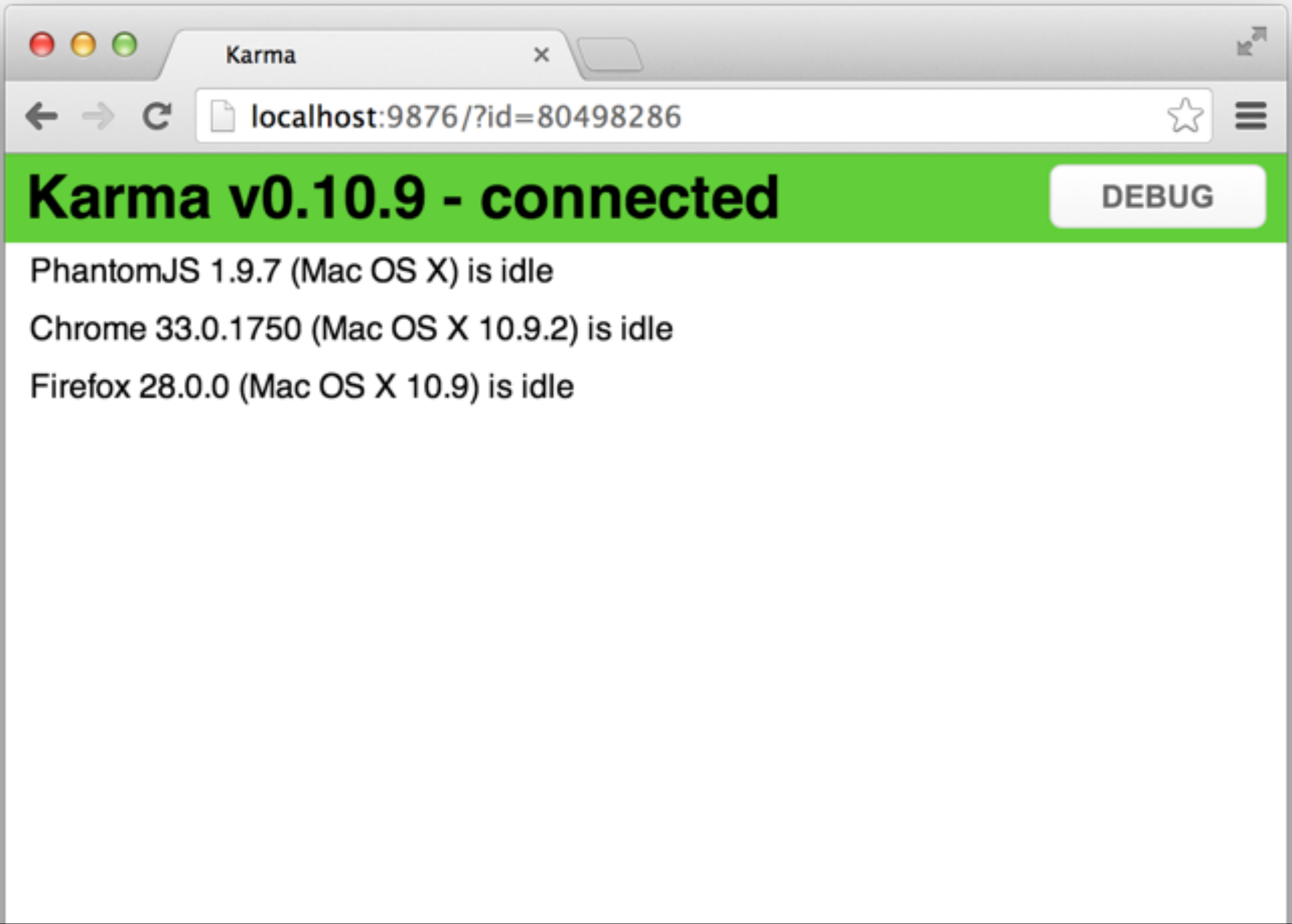
Install:

```
npm -g install karma

npm install --save-dev karma-mocha karma-chai
```

Configure & run:

```
karma init

karma start karma.conf.js
```

Debugging tests



Karma v0.10.9 - connected

PhantomJS 1.9.7 (Mac OS X) is idle

Chrome 33.0.1750 (Mac OS X 10.9.2) is idle

Firefox 28.0.0 (Mac OS X 10.9) is idle

Faking HTTP

There’s no HTTP in unit tests!
Because we like to control the environment.

$httpBackend is your friend.

Faking time

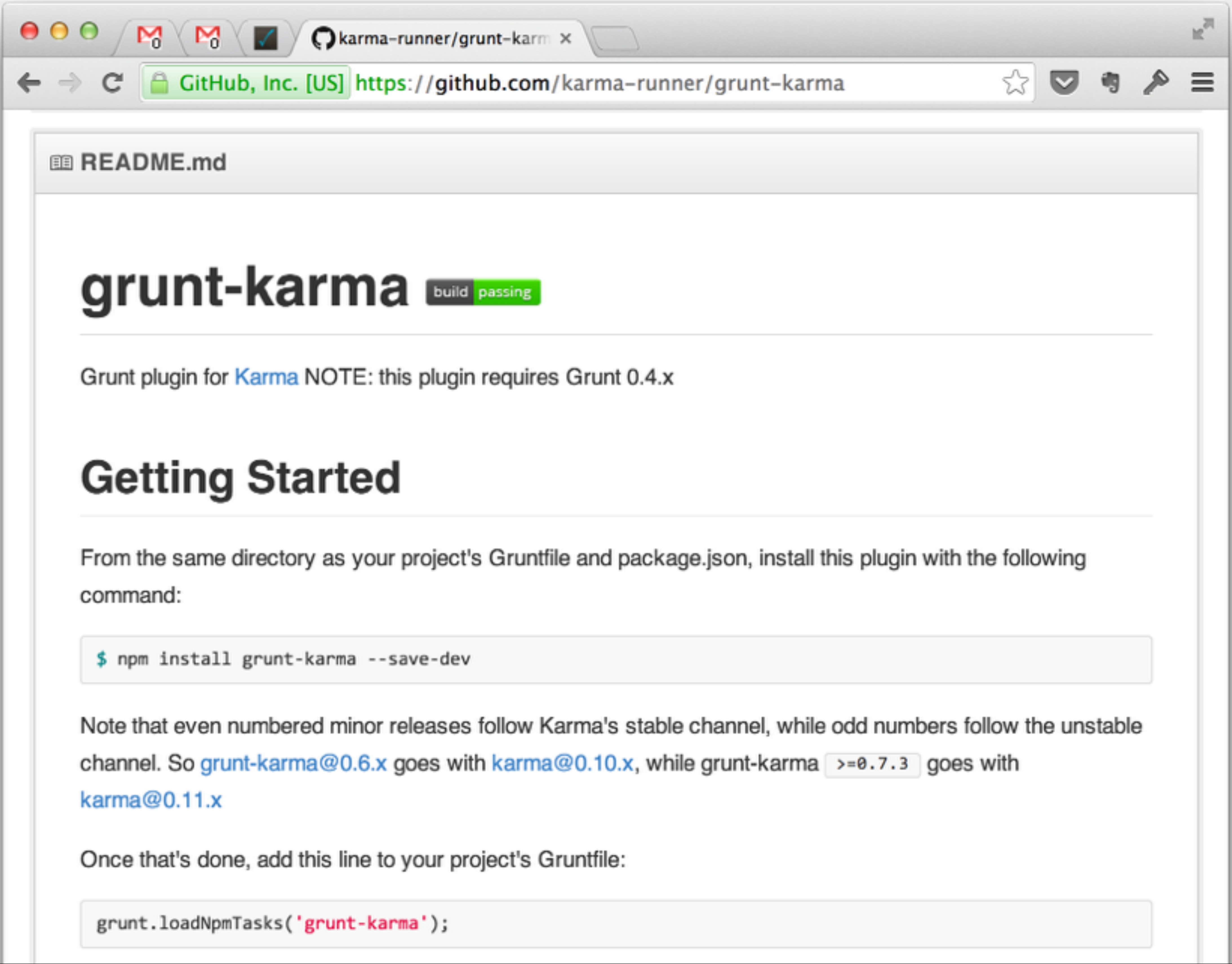
There’s no time in unit tests!
Has to be fast!

$timeout is your friend.

Learn when to $digest

Especially when testing promises and directives.

Use grunt!


karma-runner/grunt-karm
GitHub, Inc. [US] https://github.com/karma-runner/grunt-karma
README.md
build passing

grunt-karma

Grunt plugin for Karma NOTE: this plugin requires Grunt 0.4.x

Getting Started

From the same directory as your project's Gruntfile and package.json, install this plugin with the following command:

```
$ npm install grunt-karma --save-dev
```

Note that even numbered minor releases follow Karma's stable channel, while odd numbers follow the unstable channel. So grunt-karma@0.6.x goes with karma@0.10.x, while grunt-karma `>=0.7.3` goes with karma@0.11.x

Once that's done, add this line to your project's Gruntfile:

```
grunt.loadNpmTasks('grunt-karma');
```

Dependencies

Angular.JS is like an onion.

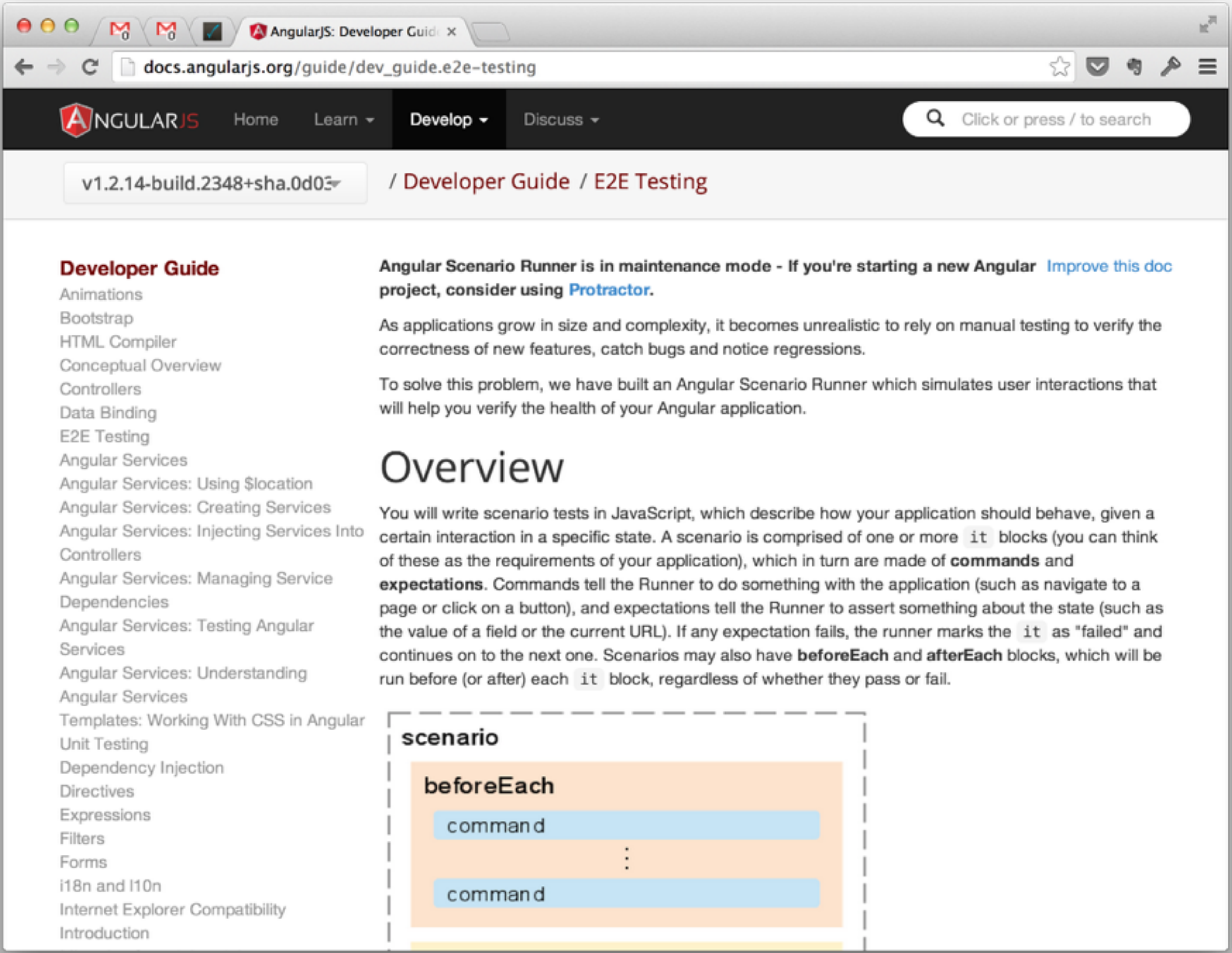
Questions?

E2E testing

Unit vs E2E

Unit test	E2E test
APIs	UIs
Tiny	Large
White-box	Black-box
Isolated from the world	Make the pieces fit together

Browser automation!



docs.angularjs.org/guide/dev_guide.e2e-testing

AngularJS: Developer Guid
ANGULARJS
Home
Learn
Develop
Discuss
Click or press / to search
v1.2.14-build.2348+sha.0d03

/ Developer Guide / E2E Testing

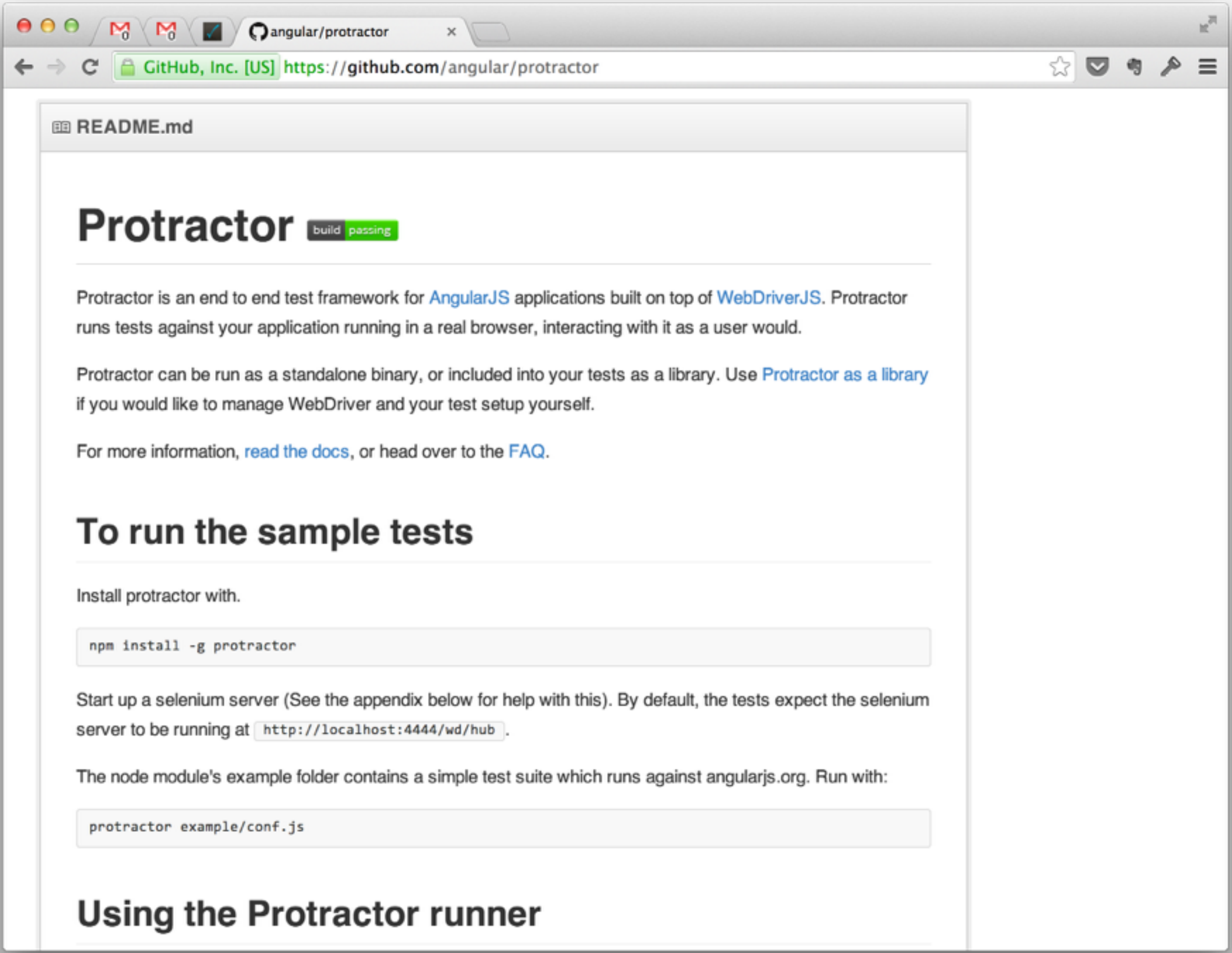
Angular Scenario Runner is in maintenance mode - If you're starting a new Angular project, consider using Protractor. Improve this doc

As applications grow in size and complexity, it becomes unrealistic to rely on manual testing to verify the correctness of new features, catch bugs and notice regressions.

To solve this problem, we have built an Angular Scenario Runner which simulates user interactions that will help you verify the health of your Angular application.

Overview

You will write scenario tests in JavaScript, which describe how your application should behave, given a certain interaction in a specific state. A scenario is comprised of one or more `it` blocks (you can think of these as the requirements of your application), which in turn are made of **commands** and **expectations**. Commands tell the Runner to do something with the application (such as navigate to a page or click on a button), and expectations tell the Runner to assert something about the state (such as the value of a field or the current URL). If any expectation fails, the runner marks the `it` as "failed" and continues on to the next one. Scenarios may also have **beforeEach** and **afterEach** blocks, which will be run before (or after) each `it` block, regardless of whether they pass or fail.


angular/protractor

GitHub, Inc. [US] https://github.com/angular/protractor

README.md

Protractor build passing

Protractor is an end to end test framework for AngularJS applications built on top of WebDriverJS. Protractor runs tests against your application running in a real browser, interacting with it as a user would.

Protractor can be run as a standalone binary, or included into your tests as a library. Use Protractor as a library if you would like to manage WebDriver and your test setup yourself.

For more information, read the docs, or head over to the FAQ.

To run the sample tests

Install protractor with.

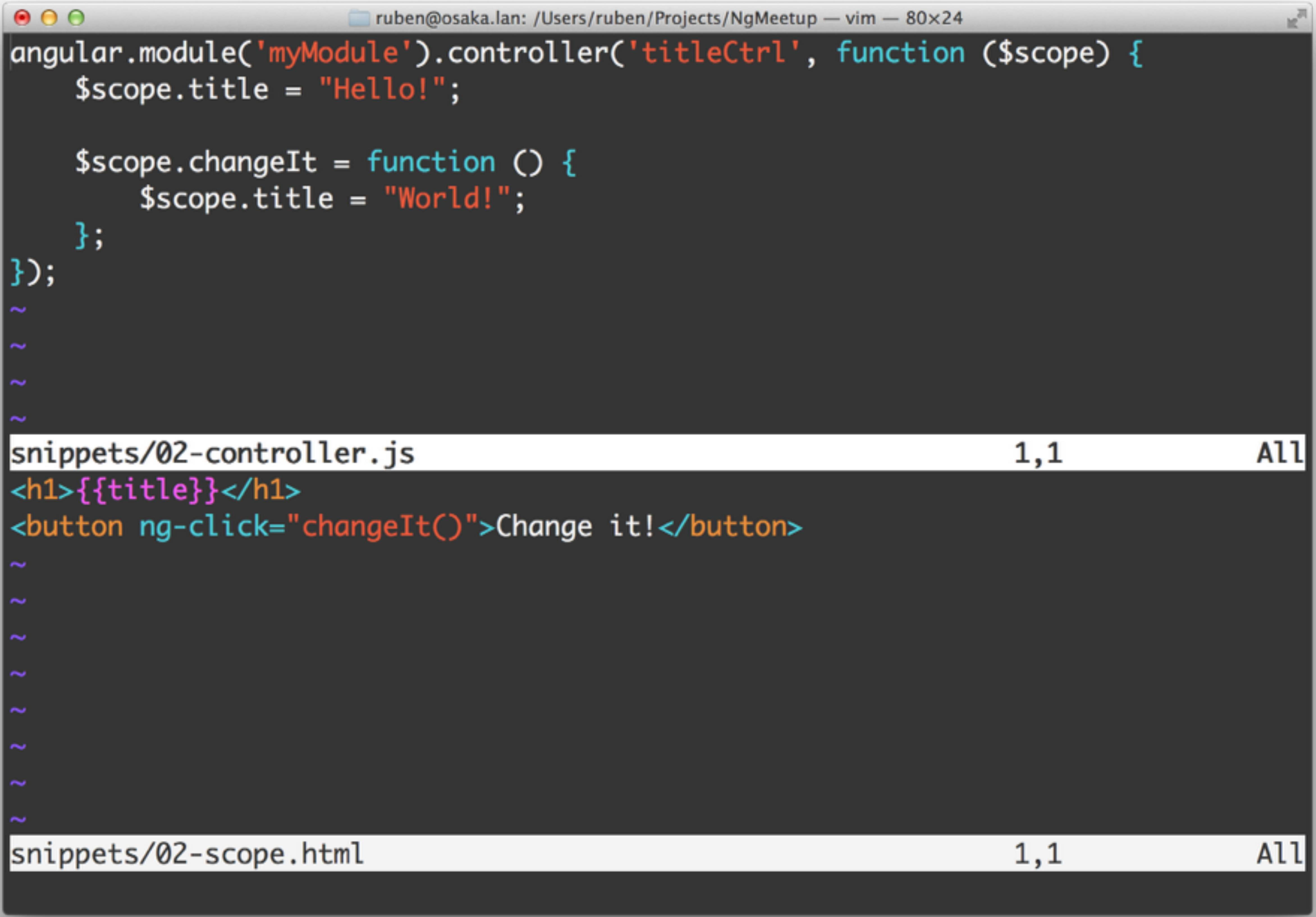
```
npm install -g protractor
```

Start up a selenium server (See the appendix below for help with this). By default, the tests expect the selenium server to be running at `http://localhost:4444/wd/hub`.

The node module's example folder contains a simple test suite which runs against angularjs.org. Run with:

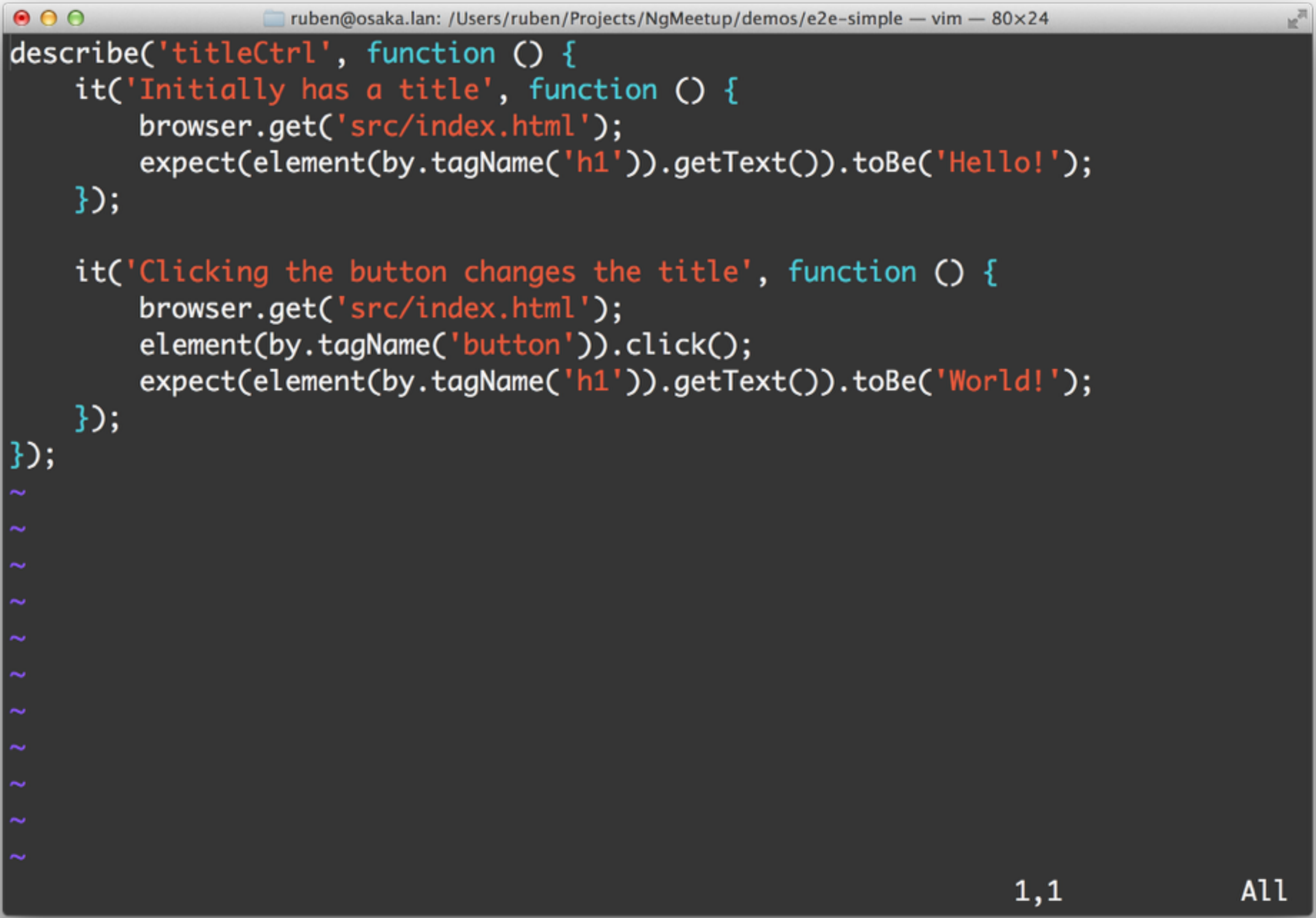
```
protractor example/conf.js
```

Using the Protractor runner



ruben@osaka.lan: /Users/ruben/Projects/NgMeetup — vim — 80×24

```
angular.module('myModule').controller('titleCtrl', function ($scope) {
    $scope.title = "Hello!";

    $scope.changeIt = function () {
        $scope.title = "World!";
    };
});
~
~
~
~
snippets/02-controller.js                                     1,1           All
<h1>{{title}}</h1>
<button ng-click="changeIt()">Change it!</button>
~
~
~
~
~
~
~
~
snippets/02-scope.html                                        1,1           All
```


ruben@osaka.lan: /Users/ruben/Projects/NgMeetup/demos/e2e-simple — vim — 80×24

```
describe('titleCtrl', function () {
    it('Initially has a title', function () {
        browser.get('src/index.html');
        expect(element(by.tagName('h1')).getText()).toBe('Hello!');
    });

    it('Clicking the button changes the title', function () {
        browser.get('src/index.html');
        element(by.tagName('button')).click();
        expect(element(by.tagName('h1')).getText()).toBe('World!');
    });
});
~
~
~
~
~
~
~
~
~
~
~
                                                              1,1           All
```

Install:

```
npm install --save-dev grunt-contrib-connect
grunt-protractor-runner
```

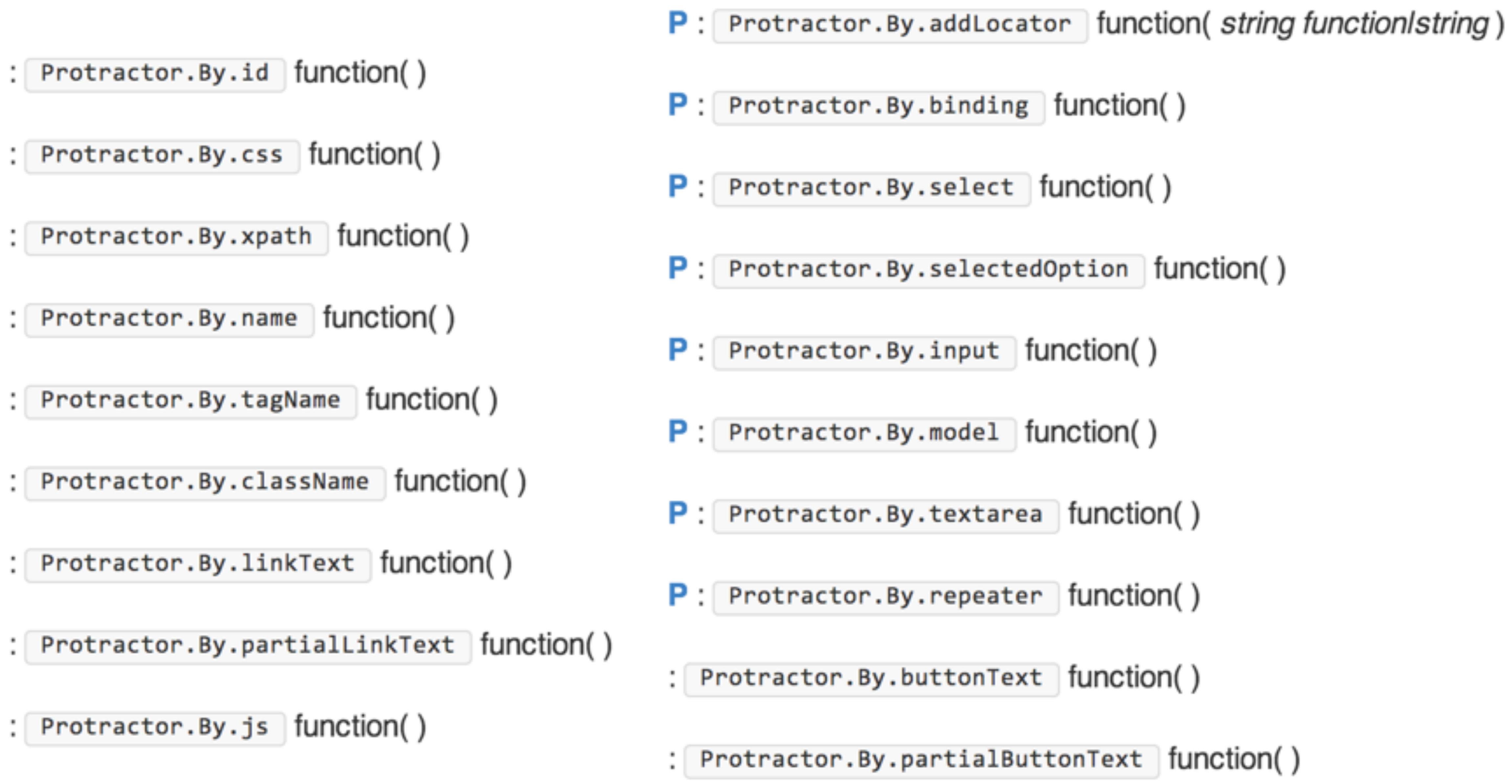
Configure & run:

```
Gruntfile.js, protractor.conf.js

grunt
```

Execution

- chromeOnly
- Local Selenium
- Sauce Labs

Locators



: `Protractor.By.id` function( )

: `Protractor.By.css` function( )

: `Protractor.By.xpath` function( )

: `Protractor.By.name` function( )

: `Protractor.By.tagName` function( )

: `Protractor.By.className` function( )

: `Protractor.By.linkText` function( )

: `Protractor.By.partialLinkText` function( )

: `Protractor.By.js` function( )

P : `Protractor.By.addLocator` function(*string function|string*)

P : `Protractor.By.binding` function( )

P : `Protractor.By.select` function( )

P : `Protractor.By.selectedOption` function( )

P : `Protractor.By.input` function( )

P : `Protractor.By.model` function( )

P : `Protractor.By.textarea` function( )

P : `Protractor.By.repeater` function( )

: `Protractor.By.buttonText` function( )

: `Protractor.By.partialButtonText` function( )

https://github.com/angular/protractor/blob/master/docs/api.md

Page objects

Reusing page structure among tests

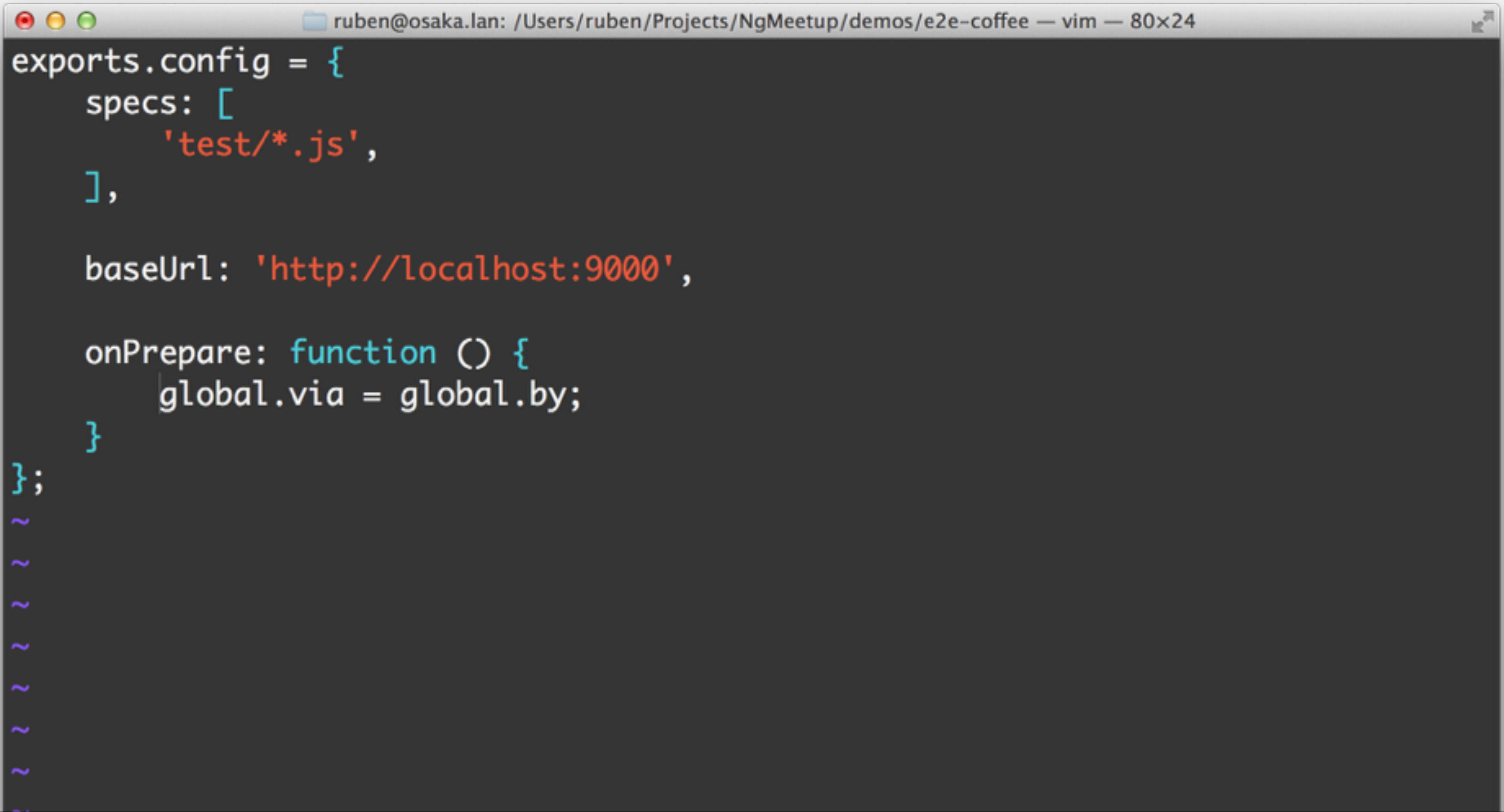
Counting things

`1 + 1 !== 3`

Authentication & Session setup

Coffeescript

ERROR ON LINE 15: UNEXPECTED BY


ruben@osaka.lan: /Users/ruben/Projects/NgMeetup/demos/e2e-coffee — vim — 80×24

```
exports.config = {
    specs: [
        'test/*.js',
    ],

    baseUrl: 'http://localhost:9000',

    onPrepare: function () {
        global.via = global.by;
    }
};
~
~
~
~
~
~
~
```

```
sendKeys(protractor.Key.RETURN)
```

vs

```
sendKeys(protractor.Key.ENTER)
```

Questions?

Testing with Angular.JS

Ruben Vermeersch
@rubenv
ruben@rocketeer.be