



# Testing in Sproutcore

[joe@matygo.com](mailto:joe@matygo.com)

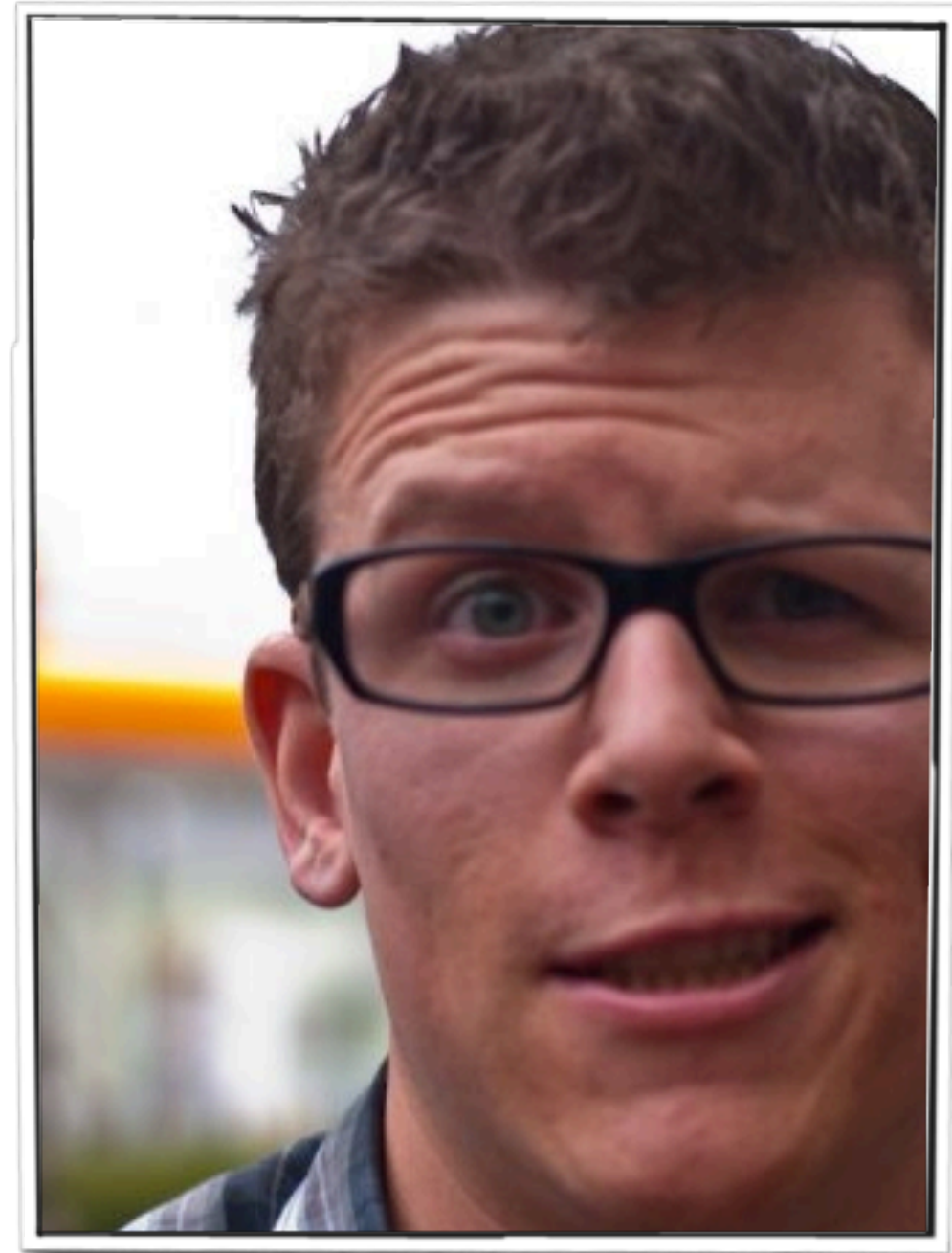
# Overview

- Who am I?
- Why Test?
- Examples of Unit Testing
- Headless Execution
- Integration Tests
- Jenkins
- Q/A

**Who Am I ?**

# I'm Joe.

- Technical Lead of Matygo
- BSc/MASC Engineering
- Long time geek
- Minor Contributor to Sproutcore



# Why Test?

# To err is Human

- Everyone makes mistakes
- Side effects cause regressions [read: rework]
- Interpreted languages don't have compilers
- Event driven systems particularly difficult to do correctly

# Why not?

- Time
- Ego
- Difficult to do correctly
- My system is too simple

# **Unit Testing**

**in**

# **Sproutcore**



# xUnit Pattern

- Setup - put the SUT into the desired state
- Exercise - call functions on the SUT
- Assert - verify appropriate reactions
- Teardown - destroy any created objects

# JUnit

# Getting Started

bin — bash — 170x40

```
joe$ sc-gen app Unit
```

```
joe$ sc-gen model Foo
```

# isEven

```
Unit.Foo = SC.Record.extend({  
  isEven: function(value) {  
    return false;  
  }  
});
```

# Some Unit Tests

```
module("Unit.Foo");
var foo = Unit.Foo.create();
// even cases
// base case
test("Foo#isEven should report even, when 0.", function() {
    ok(foo.isEven(0));
});

// second concrete odd base case
test("Foo#isEven should report even, when 2.", function() {
    ok(foo.isEven(2));
});

// indefinite (anonymous) case
test("Foo#isEven should report even, when number is even.",
function() {
    ok(foo.isEven(2 * Math.round(Math.random())));
});
```

# Some Unit Tests

```
// odd cases
// base case
test("Foo#isEven should report false, when 1.", function() {
    ok(!foo.isEven(1));
});

test("Foo#isEven should report false, when 3.", function() {
    // second concrete odd base case
    ok(!foo.isEven(3));
});

test("Foo#isEven should report false, when number is odd.",
function() {
    // indefinite (anonymous) odd case
    ok(!foo.isEven(1 + 2 * Math.round(Math.random())));
});
```

# [http://localhost:4020/sproutcore/tests#unit&test=models/foo\\_test](http://localhost:4020/sproutcore/tests#unit&test=models/foo_test)

The screenshot shows the Test Runner interface for the unit test `models/foo_test`. The sidebar on the left lists the project structure, including `Apps` (greenhouse, sc\_presentation, unit) and `Frameworks` (core, designer, device\_motion, forms, localized, polymorphism, select\_view, split\_view, SproutCore). The main area displays the test results for `models/foo_test.js`.

Summary: Completed 6 tests in 3 msec. 6 total assertions: 3 passed 3 failed

| Test   | Result |
|--|--------|
| Unit.Foo module: Is even should report even, when 0. - 0msec (0, 1, 0, 0)              | FAILED |
| Unit.Foo module: Is even should report even, when 2. - 0msec (0, 1, 0, 0)              | FAILED |
| Unit.Foo module: Is even should report even, when number is even. - 0msec (0, 1, 0, 0) | FAILED |

The interface also includes a 'Run Tests' button at the bottom right and a 'Continuous Integration' checkbox at the bottom left.

# isEven

```
Unit.Foo = SC.Record.extend({  
  isEven: function(value) {  
    return value % 2 == 0;  
  }  
});
```



|  | Result |
|--|--------|
| models/foo_test.js<br>Unit.Foo module: Is even should report even, when 0. - 0msec (1, 0, 0, 0)              | PASSED |
| models/foo_test.js<br>Unit.Foo module: Is even should report even, when 2. - 0msec (1, 0, 0, 0)              | PASSED |
| models/foo_test.js<br>Unit.Foo module: Is even should report even, when number is even. - 0msec (1, 0, 0, 0) | PASSED |
| models/foo_test.js<br>Unit.Foo module: Is even should report false, when 1. - 0msec (1, 0, 0, 0)             | PASSED |
| models/foo_test.js<br>Unit.Foo module: Is even should report false, when 3. - 0msec (1, 0, 0, 0)             | PASSED |
| models/foo_test.js<br>Unit.Foo module: Is even should report false, when number is odd. - 0msec (1, 0, 0, 0) | PASSED |

# Jasmine

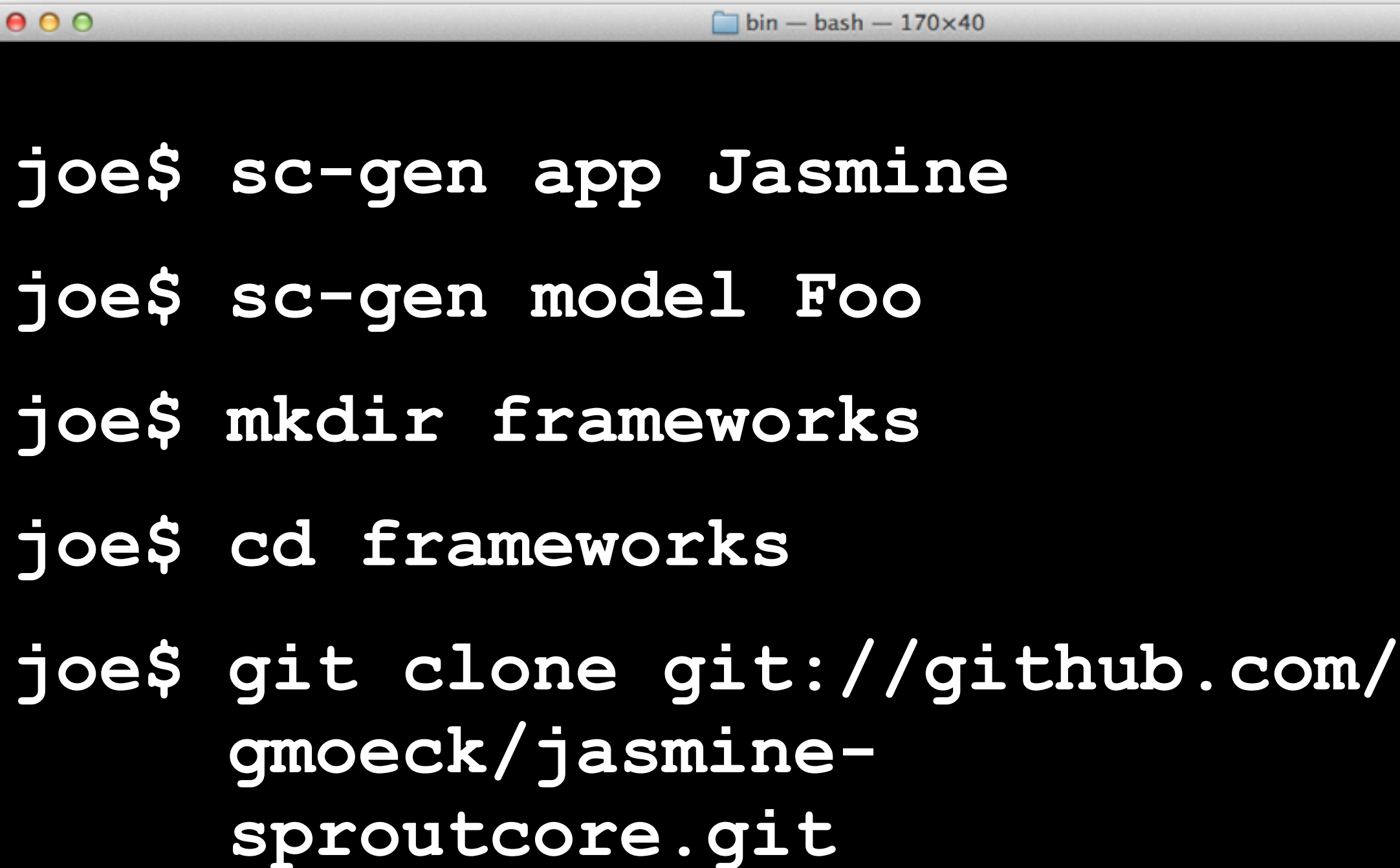
<http://pivotal.github.com/jasmine/>

# Jasmine

<https://github.com/gmoeck/jasmine-sproutcore>

- BDD style tests for JS
- Specs are easy to read / write
- Great for involving non-tech in Q/A
- Spies useful for Mocking / Spying / Faking / Stubbing etc.

# Getting Started

A terminal window with a title bar that reads "bin — bash — 170x40". The window has a dark background and contains five lines of white text, each starting with a prompt "joe\$". The commands are: "sc-gen app Jasmine", "sc-gen model Foo", "mkdir frameworks", "cd frameworks", and "git clone git://github.com/gmoeck/jasmine-sproutcore.git".

```
joe$ sc-gen app Jasmine
joe$ sc-gen model Foo
joe$ mkdir frameworks
joe$ cd frameworks
joe$ git clone git://github.com/
gmoeck/jasmine-
sproutcore.git
```

# Configuration

```
require
File.expand_path(<relativePathToJasmine>,
__FILE__)
// Your normal build config goes here
//
//
namespace :build do
  desc "builds a jasmine unit test"
  build_task :test do
    Jasmine::Builder::Test.build ENTRY, DST_PATH
  end
end
```

# isEven

```
Unit.Foo = SC.Record.extend({  
  isEven: function(value) {  
    return false;  
  }  
});
```

# Some Unit Tests

```
describe( 'Foo#isEven',  
function() {  
    var foo;  
  
    beforeEach(function() {  
        foo = Jasmine.Foo.create();  
    });  
  
    afterEach(function() {  
        foo = null;  
    });  
  
    // base case  
    it("should report even, when 0.", function() {  
        expect(foo.isEven(0)).toBe(true);  
    });  
  
    // second concrete odd base case  
    it("should report even, when 2.", function() {  
        expect(foo.isEven(2)).toBe(true);  
    });  
});
```

# Some Unit Tests

```
// indefinite (anonymous) case
it("should report even, when number is even.", function() {
    expect(foo.isEven(2 * Math.round(Math.random()))).toBe(true);
});

// base case
it("should report false, when 1.", function() {
    expect(foo.isEven(1)).toBe(false);
});

// second concrete odd base case
it("should report false, when 3.", function() {
    expect(foo.isEven(3)).toBe(false);
});

// indefinite (anonymous) odd case
it("should report false, when number is odd.", function() {
    expect(foo.isEven(1 + 2 *
Math.round(Math.random()))).toBe(false);
});
});
```



# Output

jasmine 1.0.1 revision 1286311016 Show ☐ passed ☐ skipped

6 specs, 6 failures in 0.006s Finished at Mon Dec 03 2012 13:30:51 GMT+0800 (PST) run all

Foo#isEven run

should report even, when 0.  
Expected false to be true.

Error: Expected false to be true.  
at new <anonymous> (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:94:50)  
at [object Object].toBe (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:1138:29)  
at [object Object].<anonymous> (http://localhost:4020/jasmine/en/current/tests.html:430:25)

should report even, when 2.  
Expected false to be true.

Error: Expected false to be true.  
at new <anonymous> (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:94:50)  
at [object Object].toBe (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:1138:29)  
at [object Object].<anonymous> (http://localhost:4020/jasmine/en/current/tests.html:435:25)

should report even, when number is even.  
Expected false to be true.

Error: Expected false to be true.  
at new <anonymous> (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:94:50)  
at [object Object].toBe (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:1138:29)  
at [object Object].<anonymous> (http://localhost:4020/jasmine/en/current/tests.html:440:53)

should report false, when 1.  
Expected true to be false.

Error: Expected true to be false.  
at new <anonymous> (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:94:50)  
at [object Object].toBe (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:1138:29)  
at [object Object].<anonymous> (http://localhost:4020/jasmine/en/current/tests.html:445:26)

should report false, when 3.  
Expected true to be false.

Error: Expected true to be false.  
at new <anonymous> (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:94:50)  
at [object Object].toBe (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:1138:29)  
at [object Object].<anonymous> (http://localhost:4020/jasmine/en/current/tests.html:450:26)

should report false, when number is odd.  
Expected true to be false.

Error: Expected true to be false.  
at new <anonymous> (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:94:50)  
at [object Object].toBe (http://localhost:4020/static/jasmine-sproutcore/en/current/source/jasmine.js:1138:29)  
at [object Object].<anonymous> (http://localhost:4020/jasmine/en/current/tests.html:455:58)

jasmine 1.0.1 revision 1286311016

6 specs, 6 failures in 0.006s Finished

# isEven

```
Unit.Foo = SC.Record.extend({  
  isEven: function(value) {  
    return value % 2 == 0;  
  }  
});
```

# Output

jasmine 1.0.1 revision 1286311016

6 specs, 0 failures in 0.006s Finished at Mon Dec 05 2011 13:14:05 GMT-0800 (PST)

# Headless Execution



# PhantomJS



<http://www.phantomjs.org/>



Allows for headless (scriptable) execution of JS

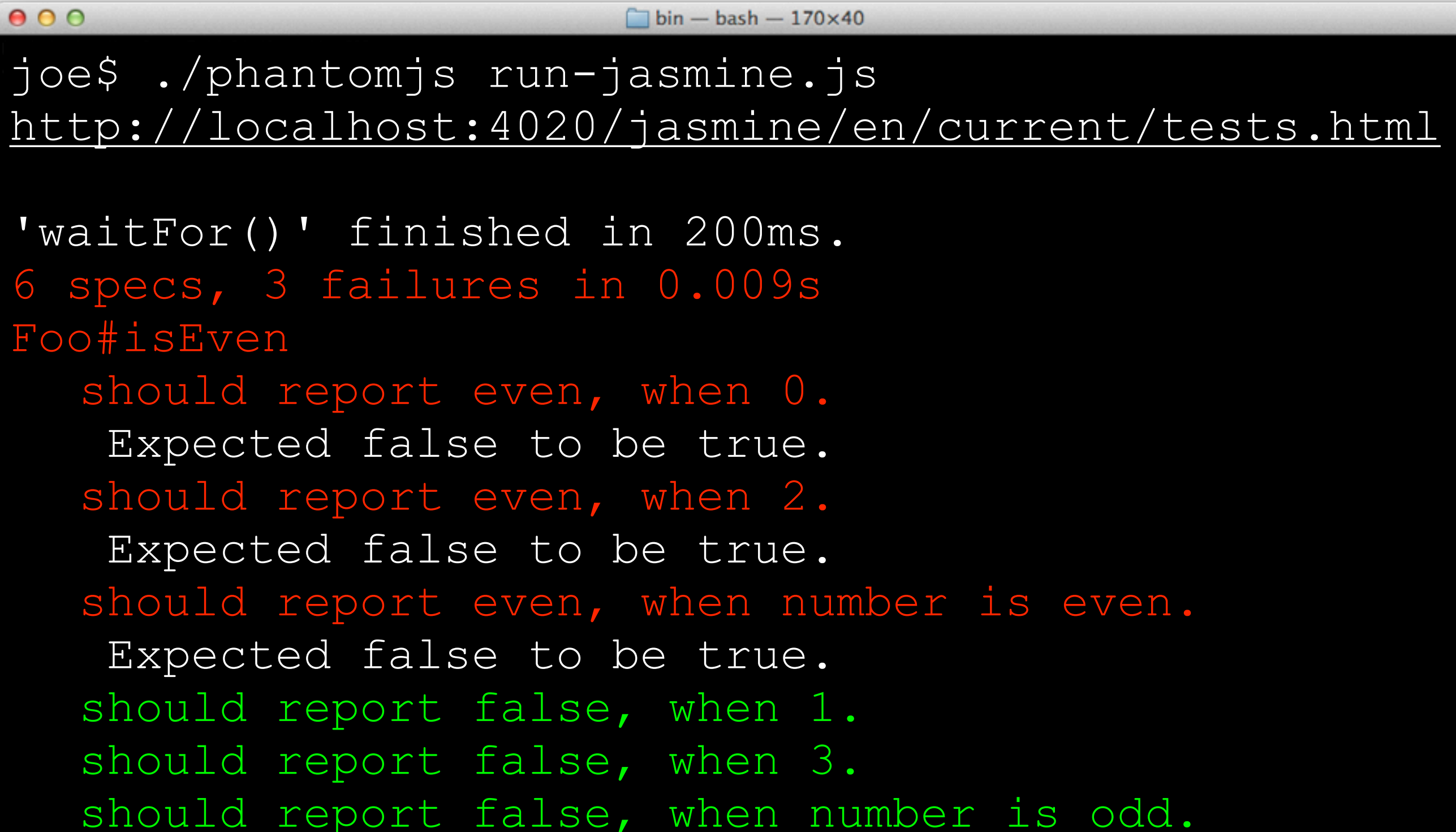


Great for fitting into integration pipeline



Has test runners for QUnit and Jasmine

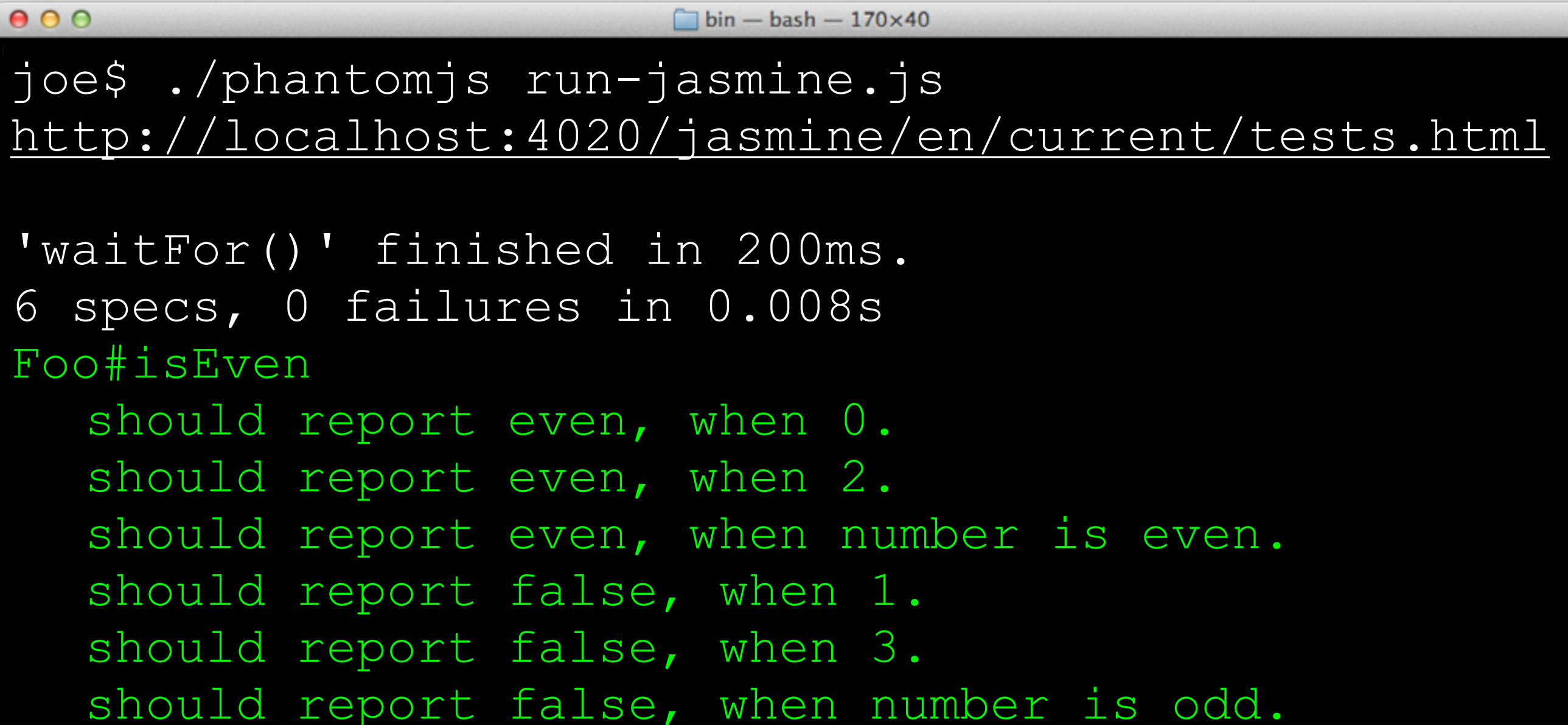
# Now With Colour



```
bin — bash — 170x40
joe$ ./phantomjs run-jasmine.js
http://localhost:4020/jasmine/en/current/tests.html

'waitFor()' finished in 200ms.
6 specs, 3 failures in 0.009s
Foo#isEven
  should report even, when 0.
  Expected false to be true.
  should report even, when 2.
  Expected false to be true.
  should report even, when number is even.
  Expected false to be true.
  should report false, when 1.
  should report false, when 3.
  should report false, when number is odd.
```

# Now With Colour



A terminal window titled "bin — bash — 170x40" displays the execution of a Jasmine test suite. The prompt "joe\$" is followed by the command `./phantomjs run-jasmine.js` and the URL `http://localhost:4020/jasmine/en/current/tests.html`. The output shows the test suite completed in 200ms with 6 specs and 0 failures. The test suite name "Foo#isEven" is highlighted in green. The individual spec descriptions are also highlighted in green: "should report even, when 0.", "should report even, when 2.", "should report even, when number is even.", "should report false, when 1.", "should report false, when 3.", and "should report false, when number is odd."

```
joe$ ./phantomjs run-jasmine.js
http://localhost:4020/jasmine/en/current/tests.html

'waitFor()' finished in 200ms.
6 specs, 0 failures in 0.008s
Foo#isEven
  should report even, when 0.
  should report even, when 2.
  should report even, when number is even.
  should report false, when 1.
  should report false, when 3.
  should report false, when number is odd.
```



# Lebowski

Q/A and Integration Testing

<https://github.com/FrozenCanuck/Lebowski>



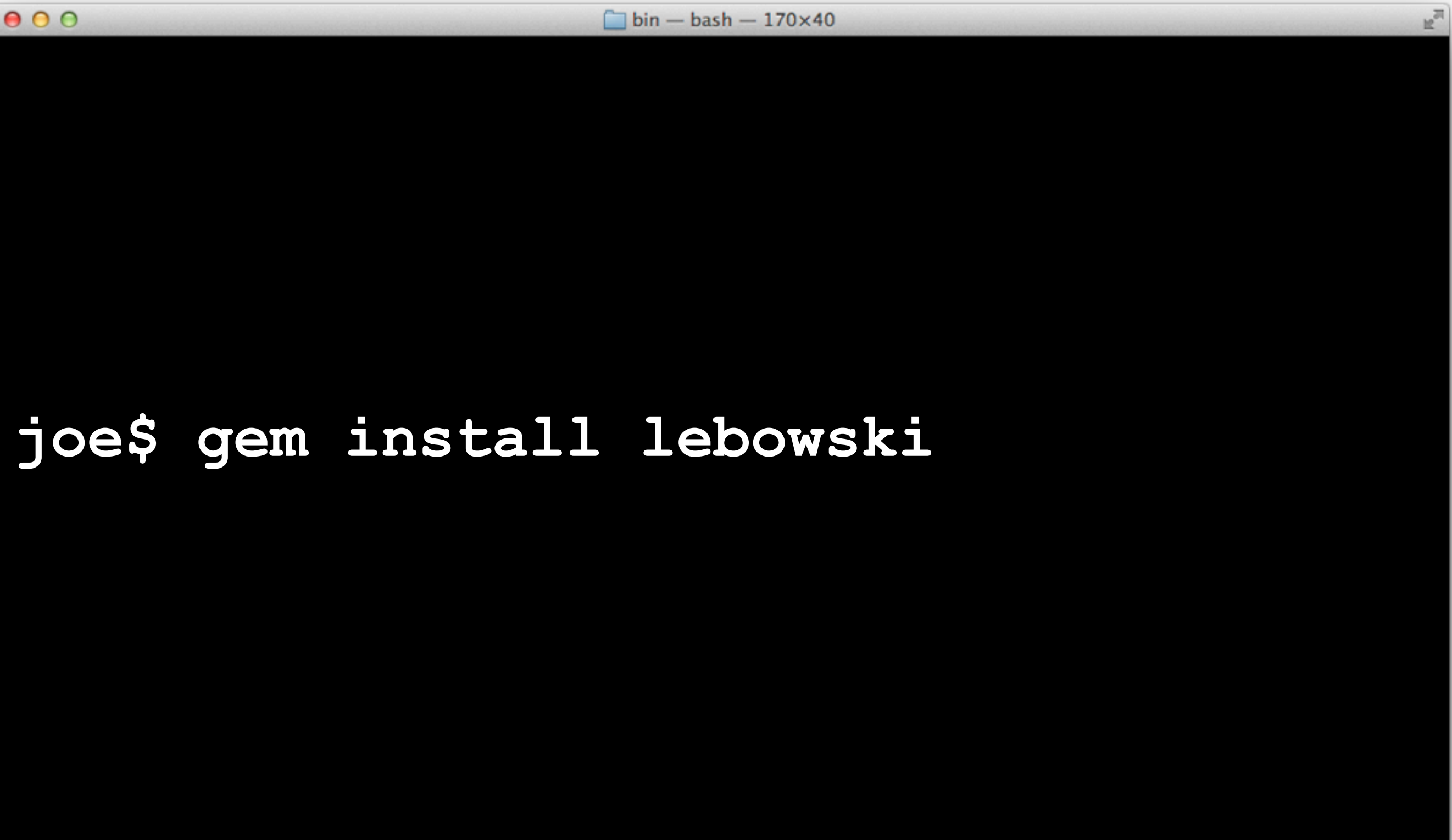
# Lebowski

- Written by FrozenCanuck (MikeC)
- Unit test are great - for units of code
- Need full feature integration testing that starts at the UI
- These are slow - run nightly [pre deploys]

# Lebowski

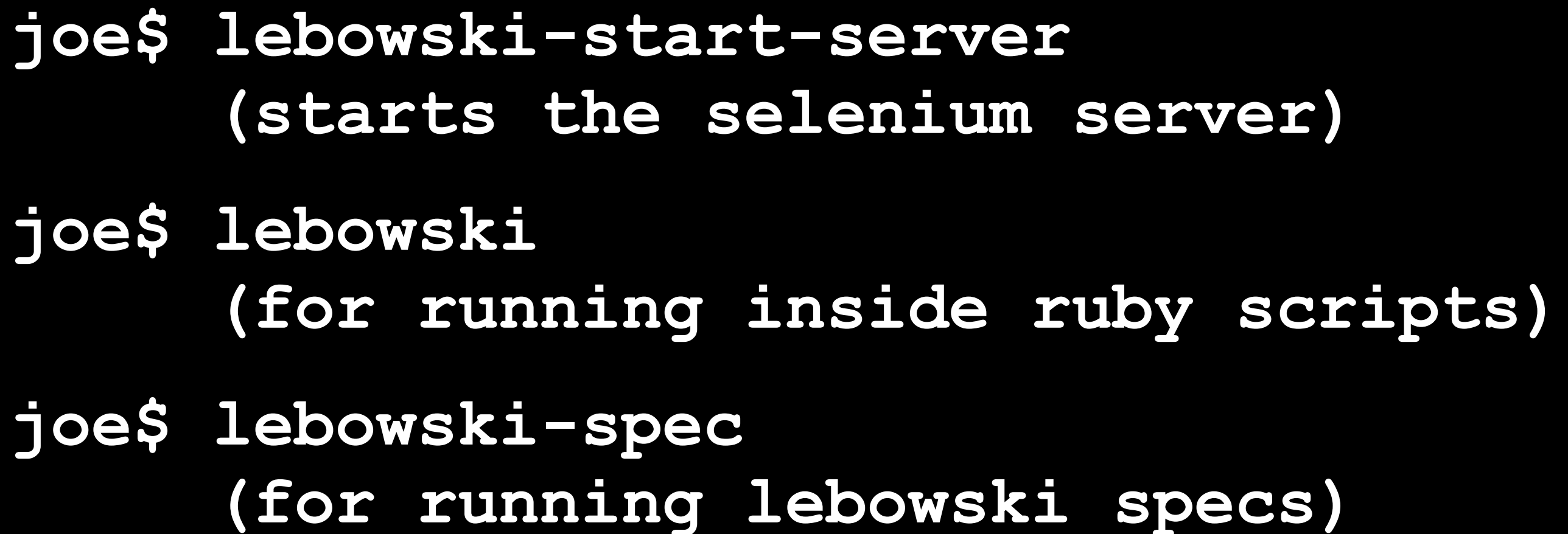
- Based on Selenium
- Uses RSpec
- With paths + proxies requires no JS
  - MyApp.mainPane.labelView.value

# Installation

A screenshot of a macOS-style terminal window. The title bar at the top shows three colored window control buttons (red, yellow, green) on the left, a folder icon followed by the text "bin — bash — 170x40" in the center, and a close button on the right. The main area of the terminal is black with white text. The text "joe\$ gem install lebowski" is displayed on a single line, with a cursor at the end of the word "lebowski".

```
joe$ gem install lebowski
```

# Useful Commands



```
bin — bash — 170x40

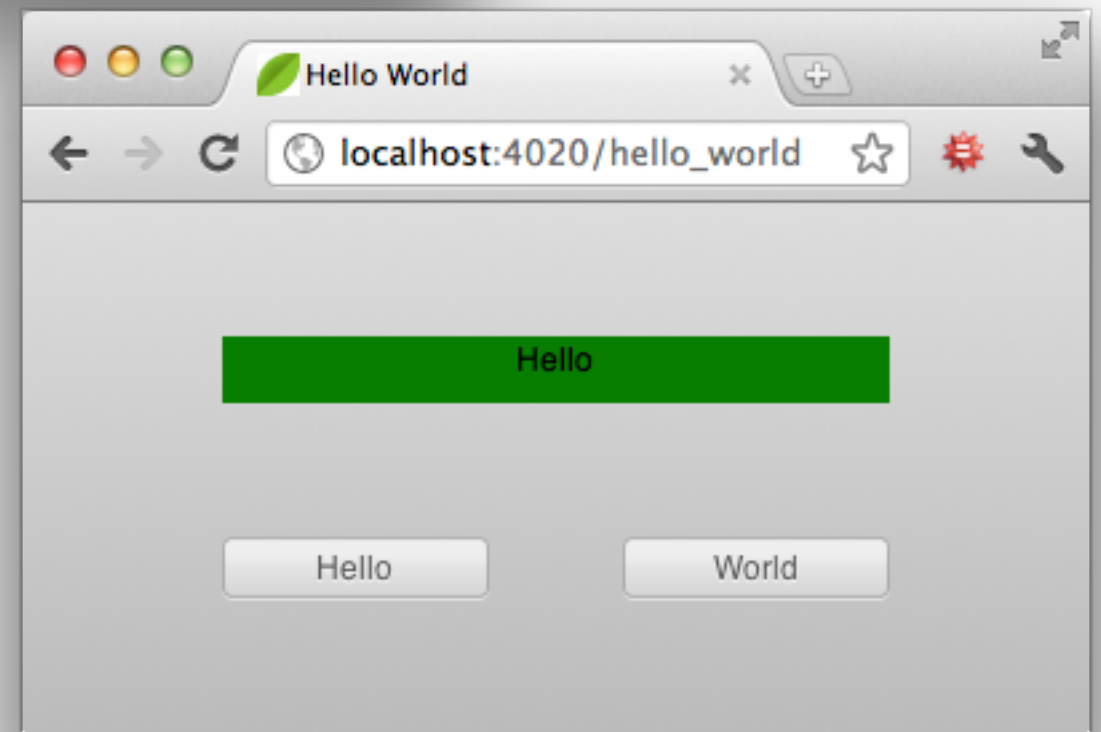
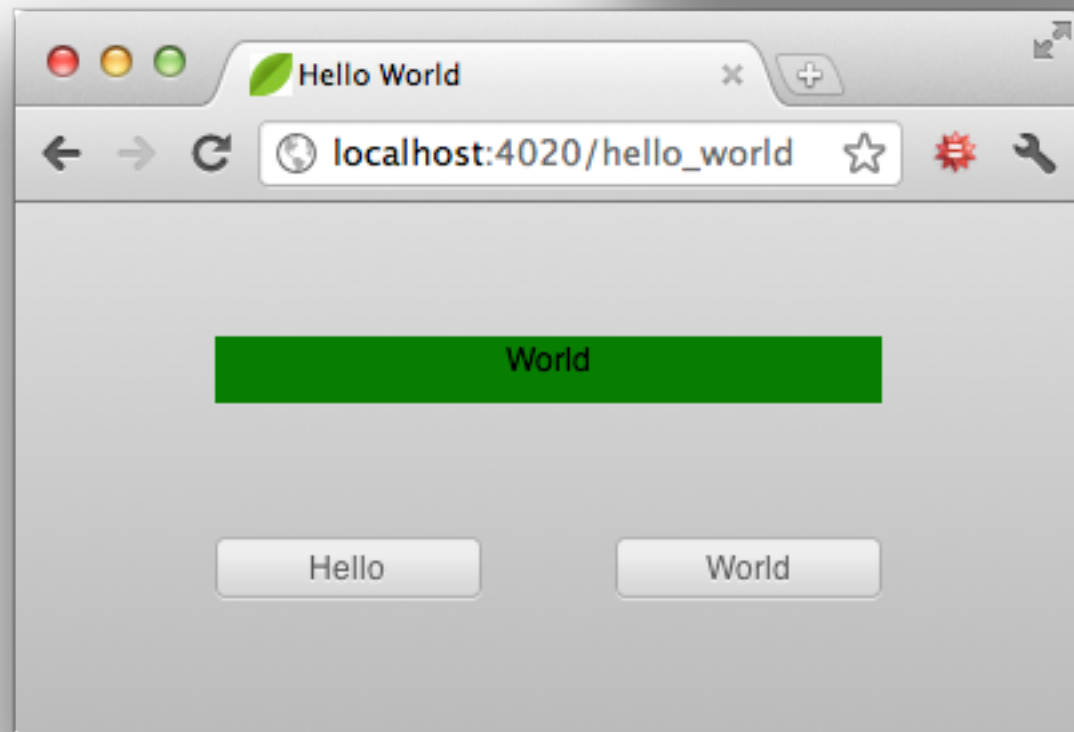
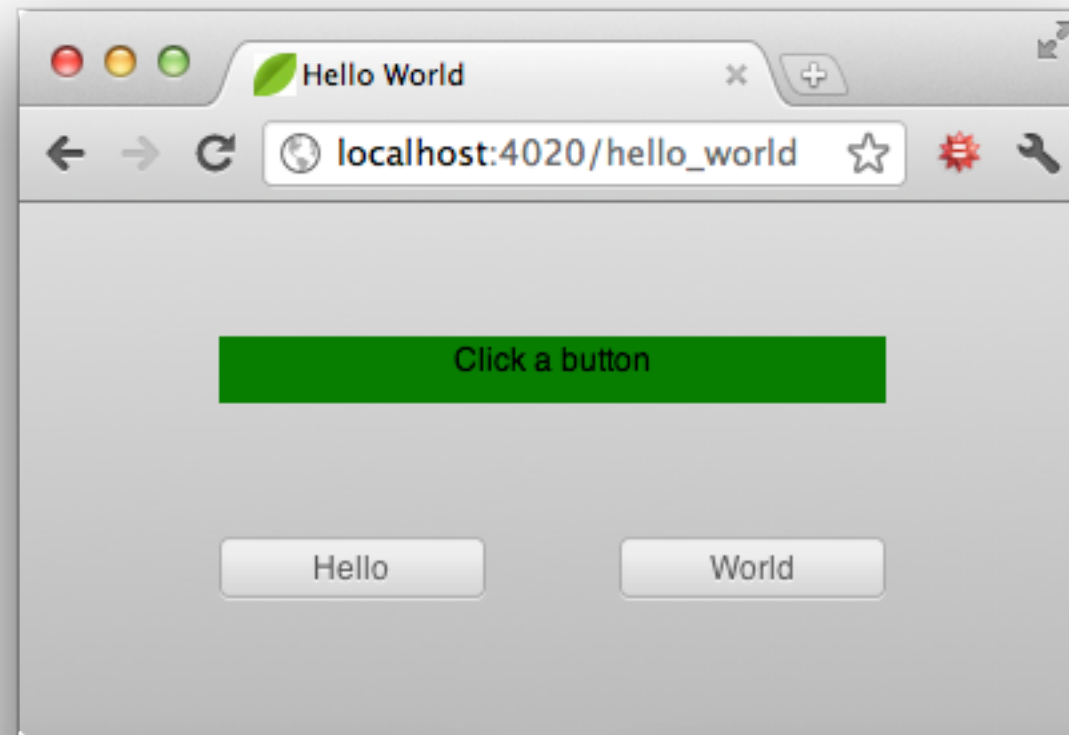
joe$ lebowski-start-server
      (starts the selenium server)

joe$ lebowski
      (for running inside ruby scripts)

joe$ lebowski-spec
      (for running lebowski specs)
```

# A Simple Example

# Hello World



# Setup

```
App = MainApplication.new \  
  :app_root_path => "/hello_world",  
  :app_name => "HelloWorldApp",  
  :browser => :firefox  
  
App.start do |app|  
  app['mainPage.mainPane.isPaneAttached'] == true  
end  
  
App.move_to 1, 1  
App.resize_to 1024, 768  
  
App.define_path 'group', 'mainPage.mainPane.groupView'
```

# A Spec

```
describe "Hello World Test" do

  before(:all) do
    @label = App['group.label', 'SC.LabelView']
    @hello_button = App['group.helloButton', ButtonView]
    @world_button = App['#world-button', ButtonView]
  end

  it "will check that label has an initial value 'click a button'" do
    @label.should have_value /click a button/i
  end

  it "will check that hello button has title 'hello'" do
    @hello_button.should have_title /hello/i
  end

  it "will check that world button has title 'world'" do
    @world_button.should have_title /world/i
  end

  it "will click hello button and then confirm label has value 'hello'" do
    @hello_button.click
    @label.should have_value /hello/i
  end

  it "will click world button and then confirm label has value 'world'" do
    @world_button.click
```



# Hello World Controller

```
HelloWorldApp.helloWorldController = SC.Object.create({  
  caption: null,  
  doHello: function() {  
    //this.set('caption', '_hello'.loc());  
  },  
  doWorld: function() {  
    //this.set('caption', '_world'.loc());  
  }  
}) ;
```



# Continuous Integration



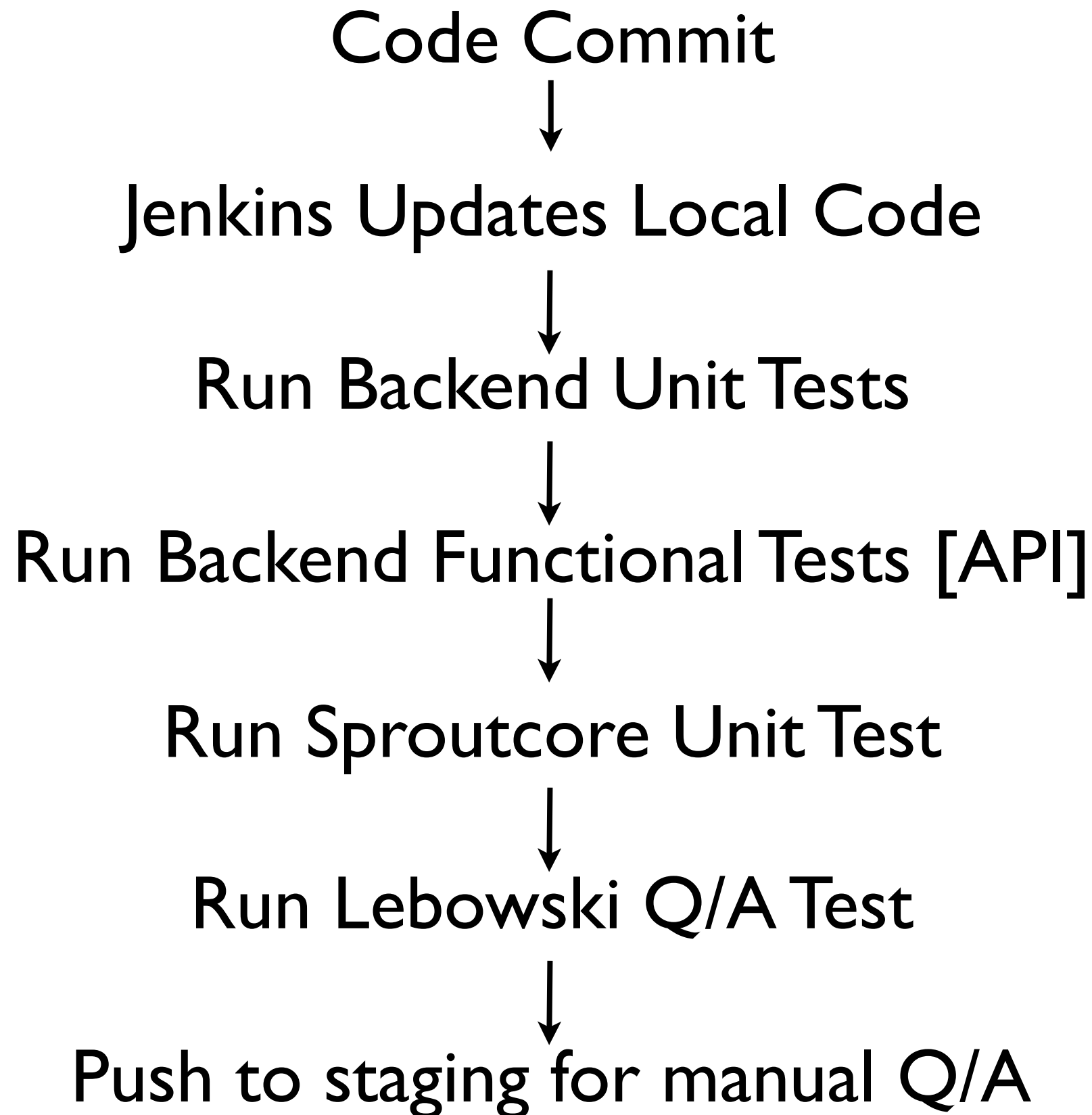
# Jenkins

# Jenkins

<http://jenkins-ci.org/>

- OpenSource CI server
- Highly configurable
- Unit tests scriptable through PhantomJS
- Integration Tests scriptable through Lebowski
- Code Quality++ !!!

# Basic Loop at Matygo





matygo

# We are Hiring

- SproutCore / Scala / Ruby On Rails
- Fun / Flexible working environment
- Delivering affordable education
  - Make a dent in the universe :P

# Questions?

[joe@matygo.com](mailto:joe@matygo.com)