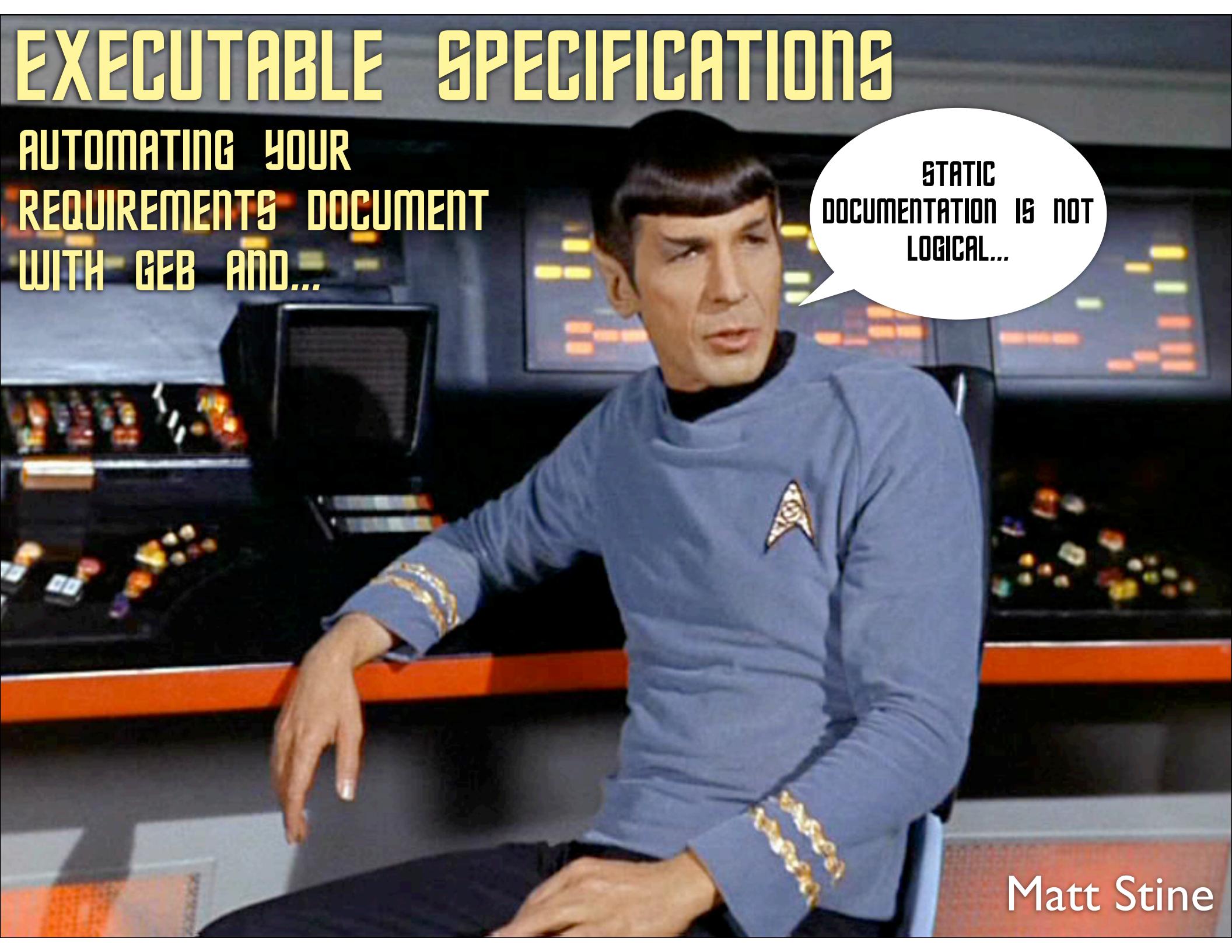


# EXECUTABLE SPECIFICATIONS

AUTOMATING YOUR  
REQUIREMENTS DOCUMENT  
WITH GEB AND...

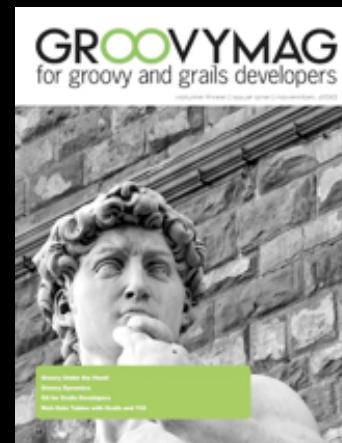
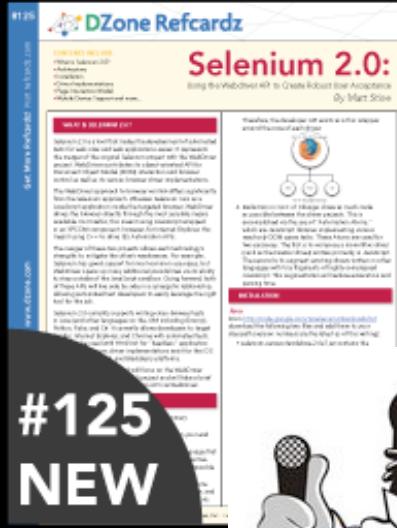
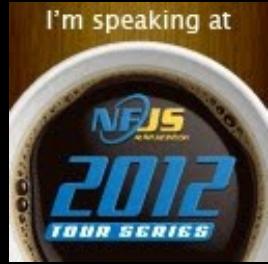
A photograph of Mr. Spock from Star Trek, wearing his iconic blue uniform with a gold Starfleet insignia. He is seated at a control panel with numerous buttons and screens, looking slightly off-camera with a serious expression. A white speech bubble originates from his mouth.

STATIC  
DOCUMENTATION IS NOT  
LOGICAL...

Matt Stine

# bash-3.2\$ whoami

- Enterprise Java/Cloud Consultant
- Speaker (JavaOne, SpringOne/2GX, CodeMash, NFJS, RWX, PAX, UberConf)
- Author (GroovyMag, NFJS the Magazine, Selenium 2.0 Refcard)
- Founder of the Memphis/Mid-South Java User Group
- Former Agile Zone Leader @ DZone



## SEARCH AND REPLACE

A user realizes he mis-capitalized a word everywhere in his document. He tells the word processor to search for all occurrences of it and replace them with the correct capitalization.

As a librarian, I want to be able to search for books by author. I also want to be able to search for books by title. I want to be able to search for books by subject. I want to be able to search for books by year.

As a student I want to purchase a parking pass so that I can drive to school.

Priority: ~~High~~ Should  
Estimate: 4

HOW DO WE KNOW WHEN WE'RE DONE?



FLUFFBOX

Welcome to Fluffbox!

Hi joeuser! Logout

FIND A SPEAKER

FIND A FLUFFBOX

COMING SOON

TOP 20 RENTALS

Get a Free Rental How Fluffbox Works

Search

Easy NFJS Speaker Rentals  
Rent and Return at over 42,000 Locations



**FIND A SPEAKER**

RWX Speakers available now.



**FIND A FLUFFBOX**

42,000 Fluffbox locations near you.

CARD

CONVERSATION

CONFIRMATION

# ACCEPTANCE TESTS

# CASE STUDY



FIND A  
MOVIE

FIND A  
REDBOX

COMING  
SOON

TOP 20  
RENTALS

BUY A  
MOVIE

HELP

Welcome to redbox!

[Log-in](#) | [Create an Account](#) | [Get a Free Rental](#) | [How redbox Works](#)

**It's Better in Blu! Find out how to win one of 5 Blu-ray™ prize packs every day.**



**Easy DVD Rentals.**

Rent and Return at nearly 25,000 Locations



**FIND A  
MOVIE**

New release DVDs available now.



**FIND A  
REDBOX**

Nearly 25,000 redbox locations.

# CASE STUDY



FIND A  
SPEAKER

FIND A  
FLUFFBOX

COMING  
SOON

TOP 20  
RENTALS

Welcome to Fluffbox!

Hi joeuser! [Logout](#)

[Get a Free Rental](#) [How Fluffbox Works](#)

[Search](#)

## Easy NFJS Speaker Rentals

Rent and Return at over 42,000 Locations



## FIND A SPEAKER

RWX Speakers available  
now.



## FIND A FLUFFBOX

42,000 Fluffbox locations  
near you.

Copyright © 2010 Fluffbox Inc.

AS A CUSTOMER, I WANT  
TO BROWSE THE LIST OF  
AVAILABLE SPEAKERS, CHOOSE  
A SPEAKER, AND THEN VIEW  
KIOSKS WHERE HE OR SHE  
IS AVAILABLE SO THAT I CAN  
RENT HIM OR HER.

# ACCEPTANCE TEST

- Given the customer is on the Fluffbox Home Page
- When the customer clicks “FIND A SPEAKER”
- And the customer clicks “Matt Stine”
- And the customer clicks “Find this Speaker”
- And the customer clicks “RENT NOW”
- And the customer clicks “Continue”
- And the customer enters his username, password, and then clicks “Log in”
- Then if we had implemented payment, we’d be ready to pay!

**IT'S THE UI STUPID!**





BUT UI TESTS ARE  
BRITTLE CAPTAIN...



```
@Test
public void testFindSpeakerFlow() throws Exception {
    selenium.open("/fluffbox-rwx/");
    selenium.click("//a[contains(@href, '/fluffbox-rwx/speaker/find')]");
    selenium.waitForPageToLoad("30000");
    selenium.click("link=Alex Miller");
    selenium.waitForPageToLoad("30000");
    selenium.click("link=Find this Speaker");
    selenium.waitForPageToLoad("30000");
    selenium.click("link=RENT NOW");
    selenium.waitForPageToLoad("30000");
    selenium.click("link=Continue");
    selenium.waitForPageToLoad("30000");
    verifyTrue(selenium.isTextPresent("This concludes your fake rental experience!"));
}

{@After
public void tearDown() throws Exception {
    selenium.stop();
}
}
```

```
@Test
public void testFindSpeakerFlow() throws Exception {
    driver.findElement(By.xpath("//a[contains(@href, '/fluffbox-rwx/speaker/find')]"))
        .click();
    driver.findElement(By.linkText("Alex Miller")).click();
    driver.findElement(By.linkText("Find this Speaker")).click();
    driver.findElement(By.linkText("RENT NOW")).click();
    driver.findElement(By.linkText("Continue")).click();

    driver.findElement(By.id("username")).sendKeys("joeuser");
    driver.findElement(By.id("password")).sendKeys("password");
    driver.findElement(By.id("loginButton")).click();

    assertTrue(driver.getPageSource()
        .indexOf("This concludes your fake rental experience!") != -1);
}

{@After
public void tearDown() throws Exception {
    driver.quit();
}
}
```

```
require 'test/unit'
require 'rubygems'
require 'watir'

class TestFindSpeakerFlow < Test::Unit::TestCase

def setup
  Watir::Browser.default = 'firefox'
  @browser = Watir::Browser.new
  @browser.goto "http://localhost:8080/fluffbox-rwx"
end

def test_find_speaker_flow
  @browser.link(:xpath, "//a[contains(@href, '/fluffbox-rwx/speaker/find')]").click
  @browser.link(:text, "Alex Miller").click
  @browser.link(:text, "Find this Speaker").click
  @browser.link(:text, "RENT NOW").click
  @browser.link(:text, "Continue").click

  assert(@browser.text.include? "This concludes your fake rental experience!")
end

def teardown
  @browser.close
end
end
```

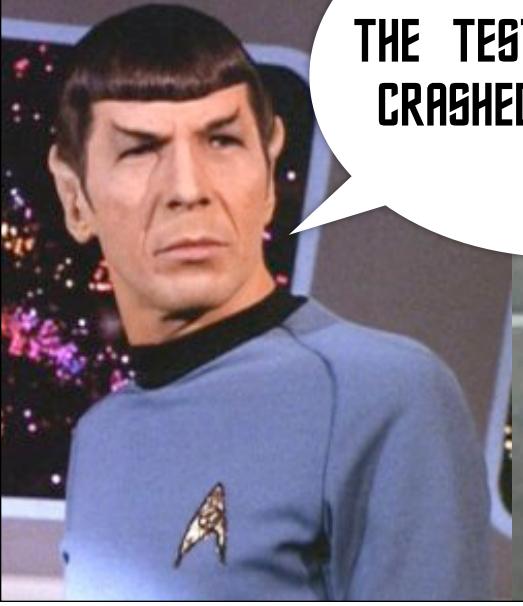
```
@Test
public void testFindSpeakerFlow() throws Exception {
    HtmlPage fluffboxHomePage = webClient.getPage("http://localhost:8080/fluffbox-rwx");
    HtmlPage firstSpeakerListPage = ((HtmlAnchor) fluffboxHomePage
        .getFirstByXPath("//a[contains(@href, '/fluffbox-rwx/speaker/find')]").click());
    HtmlPage alexMillerDetailPage = firstSpeakerListPage.getAnchorByText("Alex Miller").click();
    HtmlPage kioskPage = alexMillerDetailPage.getAnchorByText("Find this Speaker").click();
    HtmlPage rentalPage = kioskPage.getAnchorByText("RENT NOW").click();
    HtmlPage loginPage = rentalPage.getAnchorByText("Continue").click();

    HtmlForm loginForm = loginPage.getFormByName("loginForm");
    loginForm.getInputByName("j_username").setValueAttribute("joeuser");
    loginForm.getInputByName("j_password").setValueAttribute("password");

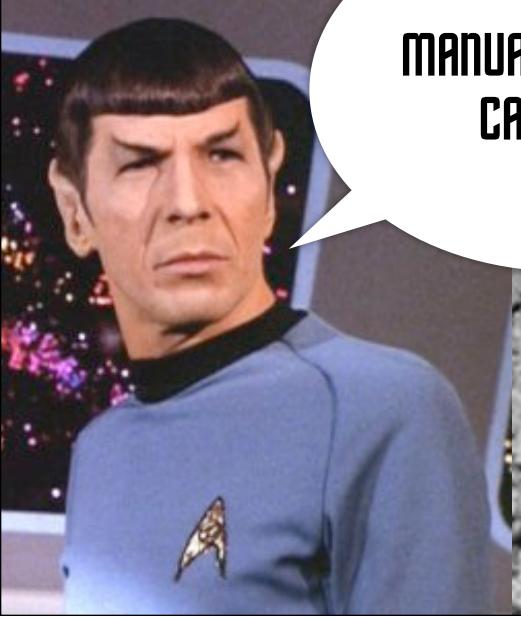
    HtmlPage lastPage = loginPage.getAnchorByText("Log in").click();
    assertTrue(lastPage.asText().contains("This concludes your fake rental experience!"));
}

{@After
public void tearDown() throws Exception {
    webClient.closeAllWindows();
}
}
```





THE TEST SUITE HAS  
CRASHED CAPTAIN...



MANUAL TESTING  
CAPTAIN?



## TESTING

- Things to look for

- performance impact - timing!!
- code coverage - want to hit every line!
- ease-of-use problems.

- Things to do

- lots of unicode strings
  - perhaps use JSS or a unicode test suite?
  - won't work for all output encodings (e.g. Chinese → KOI8-R :))
- lots of UTF-8 bytestings
- invalid UTF-8 bytestings for form input
  - Unicode consortium has some files of these that we can use.
- unicode vs. bytesting output from models.

- Random observations

- template testing can use normal test framework.
- database testing requires multiple settings files

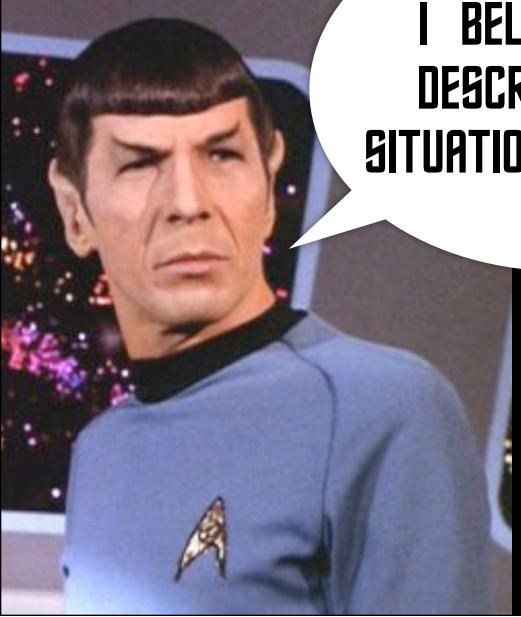
- ? • - capture form submission from various browsers
  - set up testing app that allows form submission & returns token.
  - Tester emails token, locale & browser to me.
  - maybe browser name, OS & locale can be part of the submission.



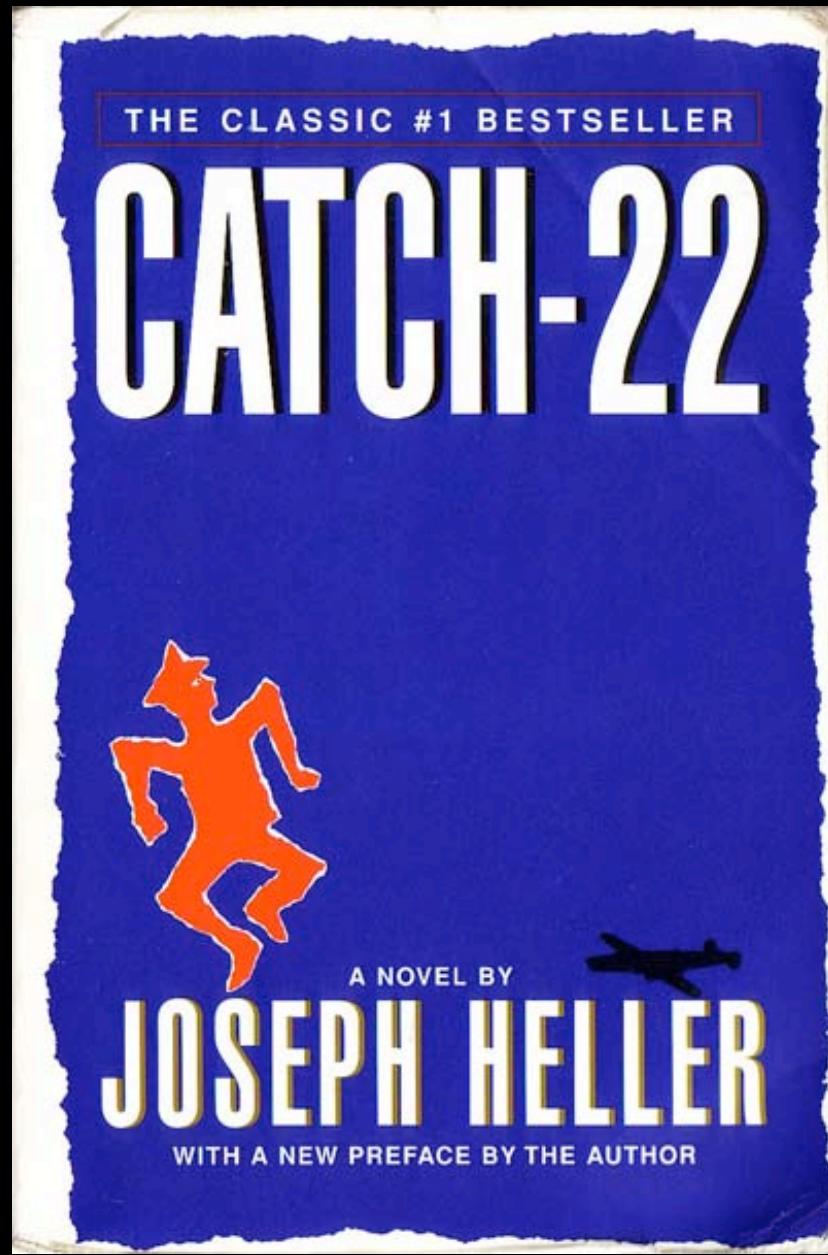
TEST

OUT OF SYNC

TEST



I BELIEVE THIS  
DESCRIBES OUR  
SITUATION CAPTAIN...



Executable UI-focused  
specifications...

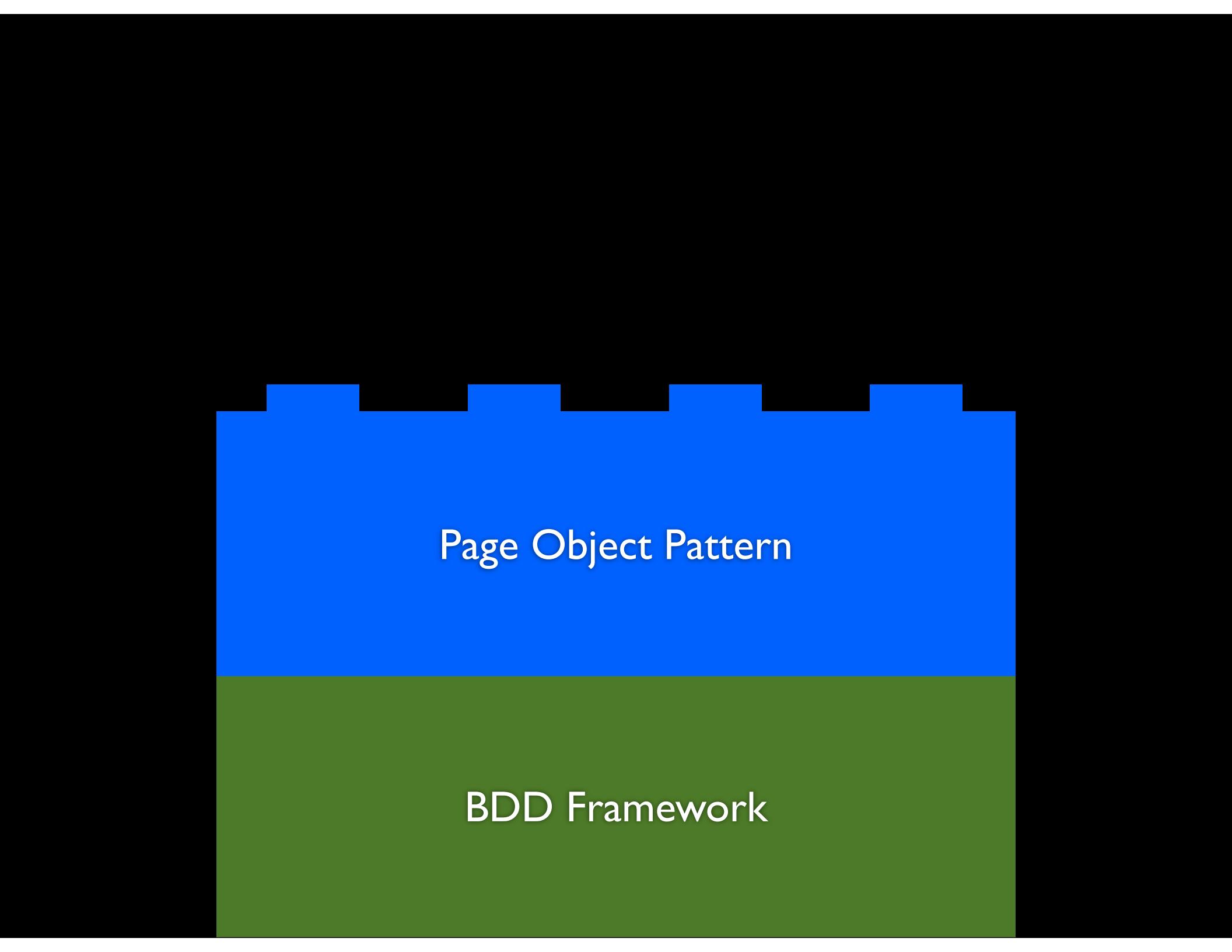
WHAT IF?

..that aren't brittle?

# BUILDING BLOCKS

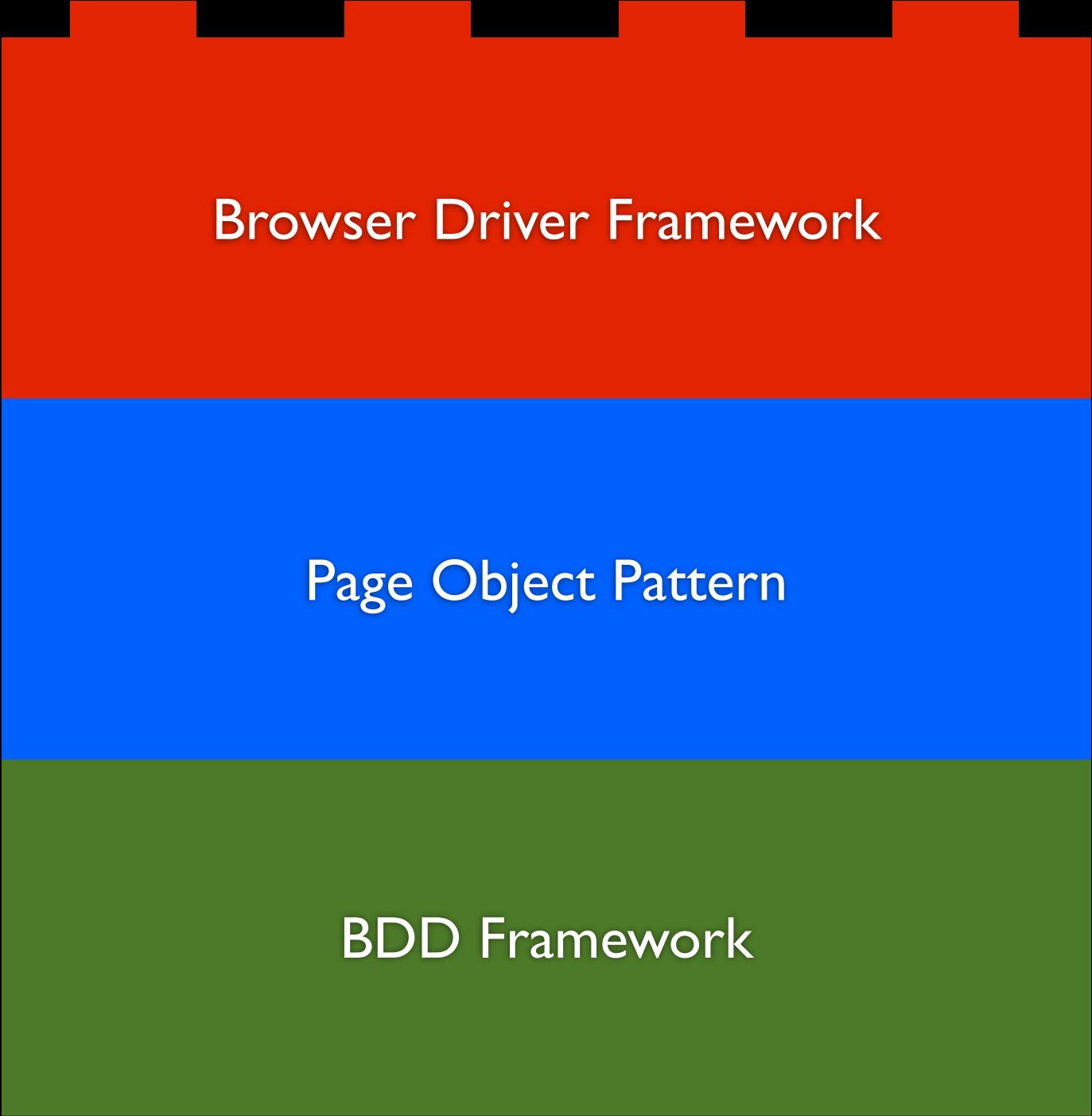


# BDD Framework



Page Object Pattern

BDD Framework



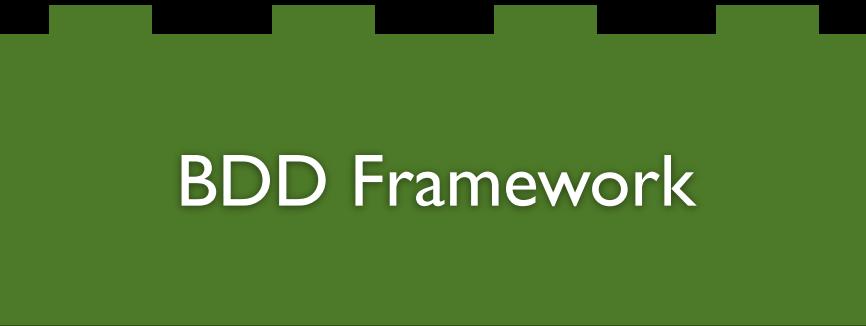
Browser Driver Framework

Page Object Pattern

BDD Framework

# SPOCK

- Testing and specification framework for JVM applications
- Created by Peter Niederwieser
- <http://code.google.com/p/spock>
- Builds on JUnit
- Highly expressive specification language, promoting BDD
- Specifically, GIVEN-WHEN-THEN



BDD Framework

<http://meetspock.appspot.com/>

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Spock Web Console

+

# Spock Web Console

```
1 class HelloSpock extends spock.lang.Specification {
2     def "length of Spock's and his friends' names"() {
3         expect:
4             name.size() == length
5
6             where:
7                 name | length
8                 "Spock" | 5
9                 "Kirk" | 4
10                "Scotty" | 6
11            }
12        }
13
14
15 }
```



Actions > Run Script New Script Publish Script View Recent Scripts

[Subscribe](#)

[Result](#) [Output](#) [Stack Trace](#)

```
HelloSpock
- length of Spock's and his friends' names
```

## Recent Scripts

- [Let's stub a ResultSet!](#) by [Peter Niederwieser](#) 3 weeks ago with tags [#mocking](#)
- [Partial Mocks](#) by [Nathan Neff](#) 1 month ago
- [Testing interaction and result](#) by [David Padbury](#) 2 months ago
- [\\$ in shared fields](#) by [Idaley](#) 4 months ago
- [Checking mock interaction](#) by [Olivier Hubaut](#) 4 months ago with tags [mock](#)
- [srivas](#) by [srivs](#) 4 months ago with tags [sr](#)

## Tips & Tricks

- To run a script, press Alt-Meta-R
- To resize the script area, drag the horizontal rule

## About Spock Web Console

• Based on [Groovy Web Console](#), code hosted on [GitHub](#)

Done

YSlow

Disable Cookies CSS Forms Images Information Miscellaneous Outline Resize Tools View Source Options

Spock Web Console

+

# Spock Web Console

```
1 class HelloSpock extends spock.lang.Specification {
2     def "length of Spock's and his friends' names"() {
3         expect:
4             name.size() == length
5
6             where:
7                 name    | length
8                 "Spock" | 5
9                 "Kirk"  | 3
10                "Scotty" | 6
11            }
12        }
13
14
15 }
```



Actions > Run Script | New Script | Publish Script | View Recent Scripts

[Subscribe](#)

**Result** Output Stack Trace

HelloSpock  
- length of Spock's and his friends' names FAILED

Condition not satisfied:

```
name.size() == length
|           |
Kirk 4      | 3
           false
```

at HelloSpock.length of Spock's and his friends' names(SCRIPT1.groovy:4)

# PAGE OBJECT PATTERN

- Separates orthogonal concerns:
  - Logical application functionality
  - Mechanics of DOM interaction
- “An API to an HTML page...”
- Promotes DRY tests and reuse
- “Peanut Brittle Insurance”
- Also known as “Window Driver Pattern” (Humble/Farley, Continuous Delivery, p.201-204)
- Supported directly by...



Page Object Pattern



- Groovy Browser Automation
- Created by Luke Daley
- <http://geb.codehaus.org>
- Pronounced with soft “G” like “Jeb”
- Combines:
  - Browser-driving via WebDriver
  - jQuery-inspired content navigation/inspection API
  - Expressiveness of Groovy



Browser Driver Framework

# ACCEPTANCE TEST

- Given the customer is on the Fluffbox Home Page
- When the customer clicks “FIND A SPEAKER”
- And the customer clicks “Matt Stine”
- And the customer clicks “Find this Speaker”
- And the customer clicks “RENT NOW”
- And the customer clicks “Continue”
- And the customer enters his username, password, and then clicks “Log in”
- Then if we had implemented payment, we’d be ready to pay!

```
import spock.lang.Specification

class NakedSpec extends Specification {

    def "test find speaker flow"() {
        given: "Customer is at the Fluffbox Home Page"

        when: "Customer clicks on 'FIND A SPEAKER'"

        and: "Customer clicks on 'Matt Stine'"

        and: "Customer clicks on 'Find this Speaker'"

        and: "Customer clicks on 'RENT NOW'"

        and: "Customer clicks on 'Continue'"

        and: "Customer enters username, password, and clicks 'Log in'"

        then: "If we'd implemented payment, we'd be ready to pay."
    }
}
```

Terminal — bash — 80x24

```
[~/Projects/exec-specs-geb-spock (master)] ➔ gradle clean test -Dtest.single=NakedSpec
```

# PAGE OBJECT

```
package com.deepsouthsoftware.execspec.pages  
  
import geb.Page  
  
class HomePage extends Page {  
    static url = "/fluffbox-rwx"  
  
    static at = { title == "Fluffbox" }  
  
    static content = {  
        findASpeaker(to: FindASpeakerPage) { $("div.headerBox > a.headerLink", 0) }  
        findAFluffbox(to: FindAFluffboxPage) { $("div.headerBox > a.headerLink", 1) }  
    }  
}
```

*How to navigate here...*

*How to verify we got there...*

*Things we can interact with...*

# GEB'S JQUERY-LIKE API

```
$("div.headerBox > a.headerLink", 0)
```

*CSS3 Selector*

*First Match*

The first anchor having the class “headerLink” that is a child of a div having the class “headerBox.”

A SLIGHT DETOUR...

# SIGNATURE OF \$ FUNCTION

```
$(<<css selector>>, <<index or range>>, <<attribute/text matchers>>)
```

```
$("div")
$("div p")
$("div", 0)
$("div", class: "header")
$("div.geb p:first[title='outline']")
$("div", text: "Hello, Rich Web Experience!")
$("div", text: iStartsWith: "hello")
```

# INDEXING

```
<p attr1="a" attr2="b">paragraph1</p>
<p attr1="a" attr2="c">paragraph2</p>
<p>paragraph3</p>
```

```
$(“p”, 0).text() == paragraph1
$(“p”, 1).text() == paragraph2
$(“p”, 2).text() == paragraph3
$(“p”)*.text() == [paragraph1, paragraph2, paragraph3]
$(“p”).text() == paragraph1
```

# ATTRIBUTE / TEXT MATCHING

```
<p attr1="a" attr2="b">paragraph1</p>
<p attr1="a" attr2="c">paragraph2</p>
<p>paragraph3</p>
```

```
$( "p", attr1: "a").size() //2
$( "p", attr2: "c").size() //1
$( "p", text: "paragraph1").size() //1
$( "p", text: "paragraph1", attr1: "a").size() //1
$( "p", text: ~/p.*/).size() //3
$( "p", text: startsWith("p")).size() // 3
$( "p", text: endsWith("2")).size() // 1
```

# FINDING AND FILTERING

```
<div class="a">
  <p class="b">Rich</p>
</div>
<div class="b">Web</div>
```

```
$( "div" ).find( ".b" ).text() == "Rich"
$( "div" ).filter( ".b" ).text() == "Web"
```

# TRaversing

```
<div class="a">
  <div class="b">
    <p class="c">uno</p>
    <p class="d">dos</p>
    <p class="e">tres</p>
  </div>
  <div class="f">quatro</div>
</div>
```

```
$("p.d").previous().text() == "uno"
$("p.d").next().text() == "tres"
$("p.d").parent().text() == "uno\ndos\ntres"
$("p.c").siblings().text() == ["dos", "tres"]
$("div.a").children().text() == ["uno\ndos\ntres", "quatro"]
```

# ACCESSING METADATA

```
<p title="a" class="a para">Rich</p>
<p title="b" class="b para">Web</p>
<p title="c" class="c para">Experience</p>
```

```
$( "p" ).text() == "Rich"
$( "p" ).name() == "p"
$( "p" ).@title == "a"
$( "p" ).classes() == ["a", "para"]
```

```
$( "p" )*.text() == ["Rich", "Web", "Experience"]
$( "p" )*.name() == ["p", "p", "p"]
$( "p" )*.@title == ["a", "b", "c"]
$( "p" )*.classes() == [[["a", "para"], ["b", "para"], ["c", "para"]]]
```

OK, BACK TO THE SPEC...

# A GEB SPEC

```
package com.deepsouthsoftware.execspec.specs

import geb.spock.GebSpec
import com.deepsouthsoftware.execspec.pages.*

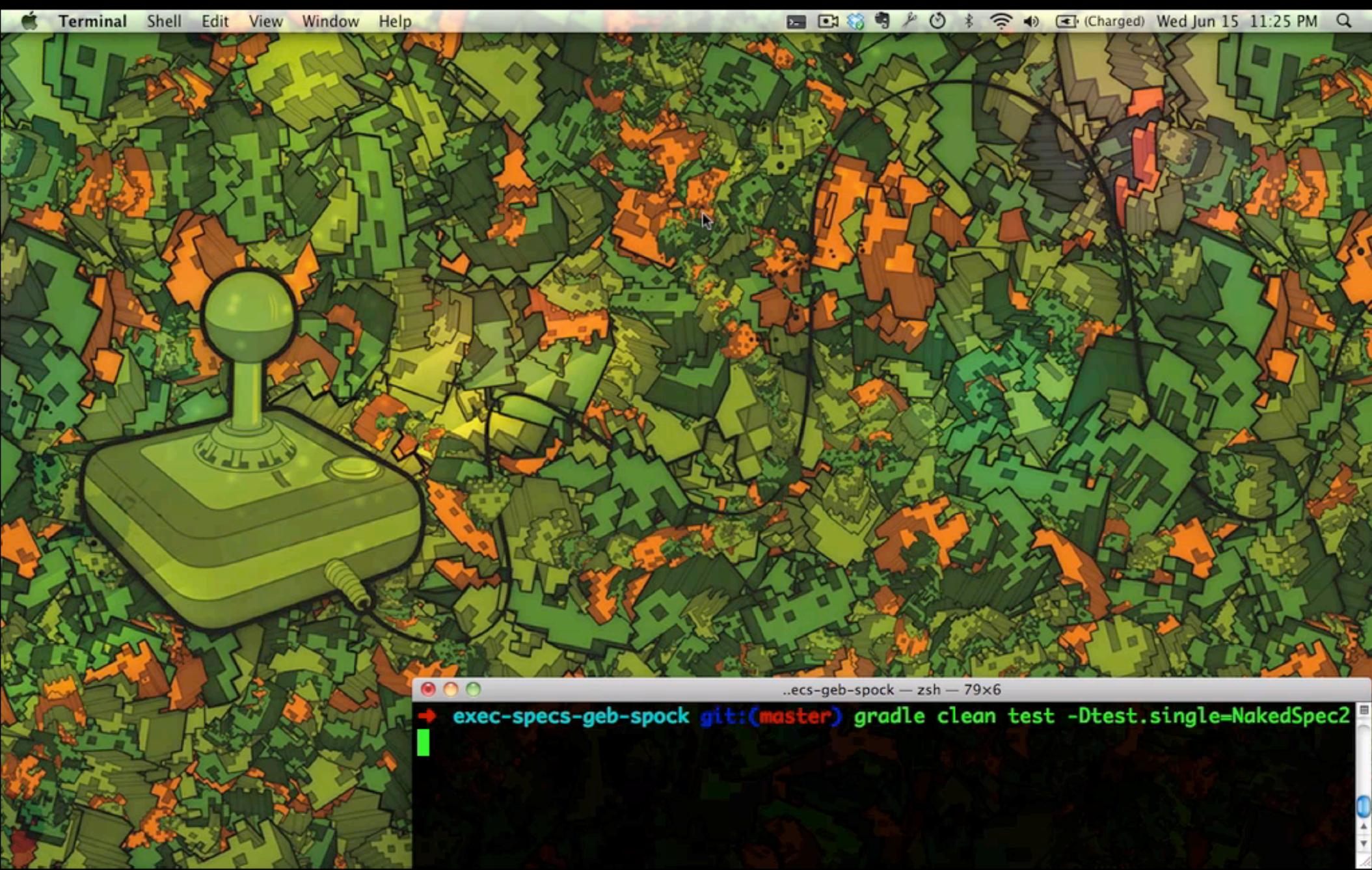
class NakedSpec2 extends GebSpec {

    String getBaseUrl() {
        "http://localhost:8080"
    }

    def "test find speaker flow"() {
        given: "Customer is at the Fluffbox Home Page"
        to HomePage

        when: "Customer clicks on 'FIND A SPEAKER'"
        findASpeaker.click()

        and: "Customer clicks on 'Matt Stine'"
        at FindASpeakerPage
    }
}
```



<http://localhost:8080/fluffbox-rwx/speaker/find>

FLUFFBOX

FIND A SPEAKER   FIND A FLUFFBOX   COMING SOON   TOP 20 RENTALS

Welcome to Fluffbox!   Log-in   Create an Account   Get a Free Rental   How Fluffbox Works  

## All Speakers

1

 <a href="#">Craig Walls</a>	 <a href="#">Venkat Subramaniam</a>	 <a href="#">Matt Stine</a>	 <a href="#">Brian Sletten</a>
 <a href="#">Ken Sipe</a>	 <a href="#">Nathaniel Schutta</a>	 <a href="#">Mark Richards</a>	 <a href="#">Pratik Patel</a>

Copyright © 2010 Fluffbox Inc.

*\$("div.speakerRow > div.speaker > a", 2)*

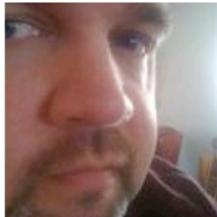
<http://localhost:8080/fluffbox-rwx/speakerClone/findByKiosk/1>

Welcome to Fluffbox!      Log-in    Create an Account    Get a Free Rental    How Fluffbox Works           

## All Speakers

Selected Fluffbox:

Walgreens (Inside)  
3895 W Broward Blvd  
Ft Lauderdale, FL 33312-1017



[Craig Walls](#)



[Venkat Subramaniam](#)



[Matt Stine](#)



[Brian Sletten](#)



[Ken Sipe](#)



[Nathaniel Schutta](#)



[Mark Richards](#)



[Pratik Patel](#)

Copyright © 2010 Fluffbox Inc.

*\$("div.speakerRow > div.speaker > a", 2)*

# MODULES

```
package com.deepsouthsoftware.execspec.modules

import geb.Module

class MattStineModule extends Module {

    static content = {
        link { $("div.speakerRow > div.speaker > a", 2) }
    }

}
```

# USING MODULES IN PAGES

```
package com.deepsouthsoftware.execspec.pages

import geb.Page
import com.deepsouthsoftware.execspec.modules.SpeakerDetailLinkModule

class MattStineDetailPage extends Page {
    static url = "speaker/show/3"

    static at = { $("h1").text() == "Matt Stine" }

    static content = {
        findThisSpeaker(to: FindKioskBySpeakerPage) { module SpeakerDetailLinkModule }
    }
}
```

# USING MODULES IN SPECS

```
class NakedSpec3 extends GebSpec {

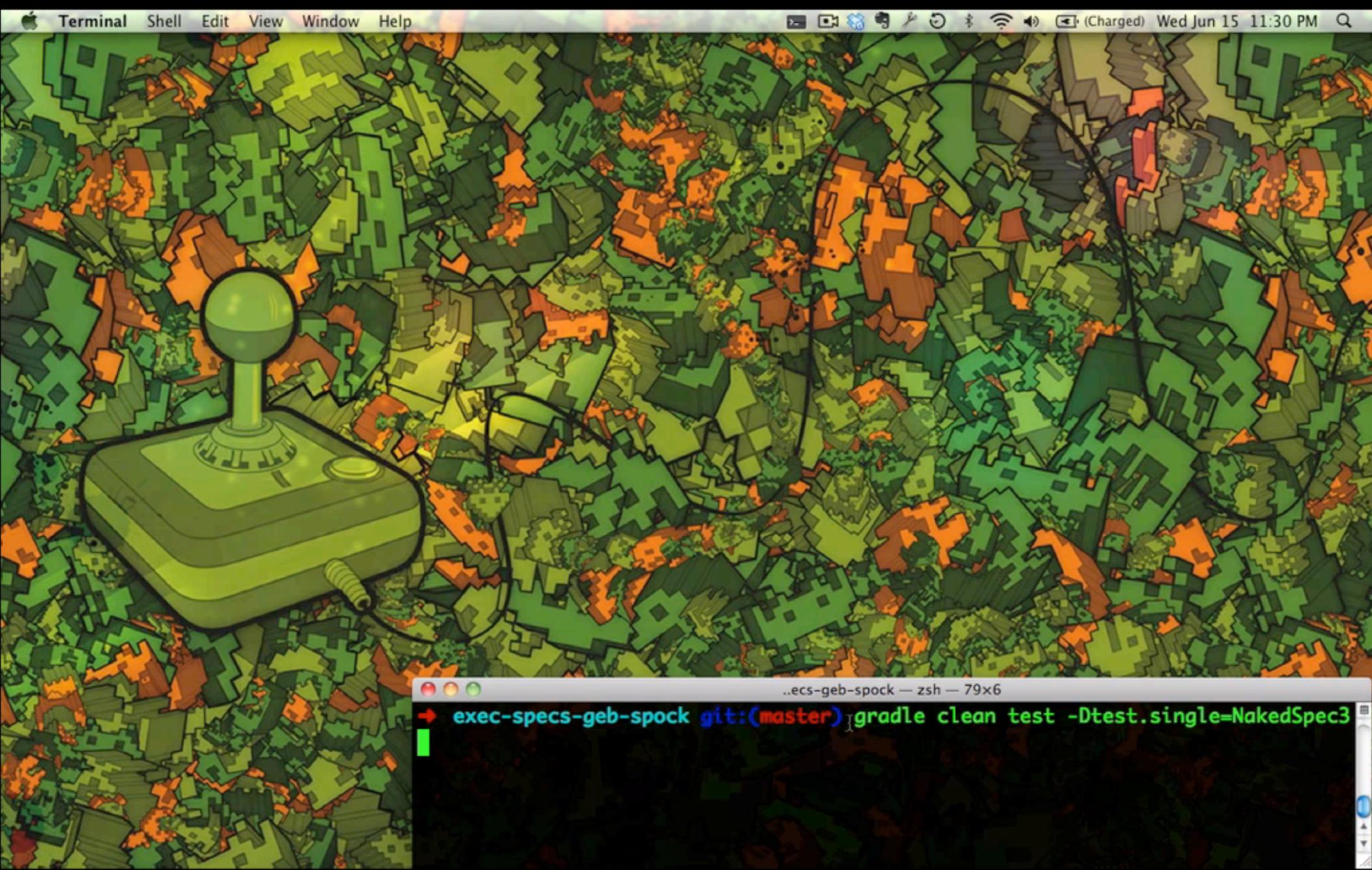
    String getBaseUrl() {
        "http://localhost:8080"
    }

    def "test find speaker flow"() {
        given: "Customer is at the Fluffbox Home Page"
        to HomePage

        when: "Customer clicks on 'FIND A SPEAKER'"
        findASpeaker.click()

        and: "Customer clicks on 'Matt Stine'"
        at FindASpeakerPage
        mattStine.link.click()

        and: "Customer clicks on 'Find this Speaker'"
        at MattStineDetailPage
    }
}
```



..ecs-geb-spock — zsh — 79x6

```
exec-specs-geb-spock git:(master) gradle clean test -Dtest.single=NakedSpec3
```

# MORE MODULES AND PAGES

```
and: "Customer clicks on 'Find this Speaker'"  
at MattStineDetailPage  
findThisSpeaker.link.click()
```

```
and: "Customer clicks on 'RENT NOW'"  
at FindKioskBySpeakerPage  
rentNow.link.click()
```

```
and: "Customer clicks on 'Continue'"  
at ReserveSpeakerPage  
continueToSignInOrCreateAccount.click()
```



FIND A  
SPEAKER

FIND A  
FLUFFBOX

COMING  
SOON

TOP 20  
RENTALS

Welcome to Fluffbox!

Hi joeuser! [Logout](#)

[Get a Free Rental](#) [How Fluffbox Works](#)

[Search](#)

## Find a Fluffbox

Selected speaker:

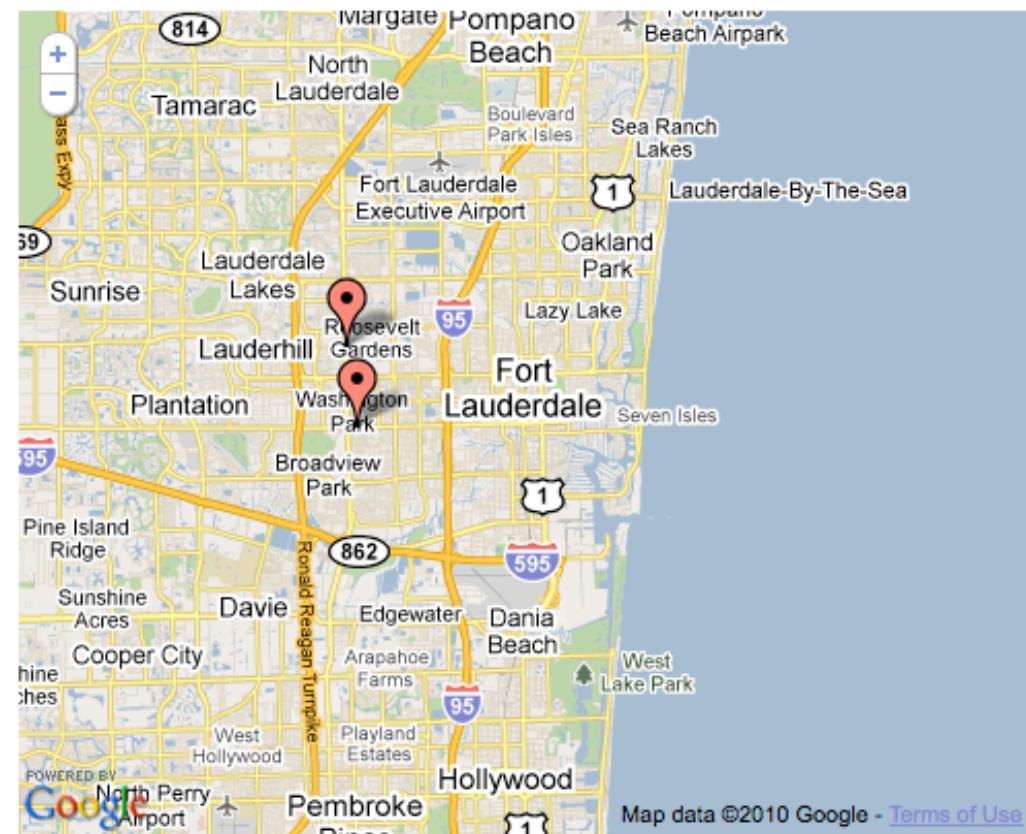
Alex Miller  
Sr. Engineer with Terracotta Inc.

**Walgreens (Inside)**  
3895 W Broward Blvd  
Ft Lauderdale, FL 33312-1017

[RENT NOW](#)

**Winn Dixie (Outside)**  
1531 N State Road 7  
Lauderhill, FL 33313-5807

[RENT NOW](#)



\$('a.kioskLink').getAt(1)

```
package com.deepsouthsoftware.execspec.modules

import geb.Module

class FirstKioskModule extends Module {

    static content = {
        link { $('a.kioskLink', 1) }
    }

}
```

```
package com.deepsouthsoftware.execspec.pages

import geb.Page
import com.deepsouthsoftware.execspec.modules.FirstKioskModule

class FindKioskBySpeakerPage extends Page {
    static url = "speakerClone/findBySpeaker/3"

    static at = {
        $("h2").text() == "Find a Fluffbox"
        &&
        $("h3").text() == "Selected speaker:"
    }

    static content = {
        rentNow(to: ReserveSpeakerPage) { module FirstKioskModule }
    }
}
```

[FIND A  
SPEAKER](#)[FIND A  
FLUFFBOX](#)[COMING  
SOON](#)[TOP 20  
RENTALS](#)

Welcome to Fluffbox!

[Log-in](#)   [Create an Account](#)   [Get a Free Rental](#)   [How Fluffbox Works](#) [Search](#)

## Pickup location:

Walgreens (Inside)  
3895 W Broward Blvd  
Ft Lauderdale, FL 33312-1017

## Selected speaker:



**Matt Stine**  
Technical Architect, AutoZone

**Click Continue to sign-in or  
create an account.**

## Your total:

Sub total	\$1.00
Tax	\$0.07
Total	\$1.07

# GEB FORM INTERACTION

**Member sign in**

Username:

Password:

Remember me | [Forgot password?](#)

[Register as new User](#) [Log in](#)

# GEB FORM INTERACTION

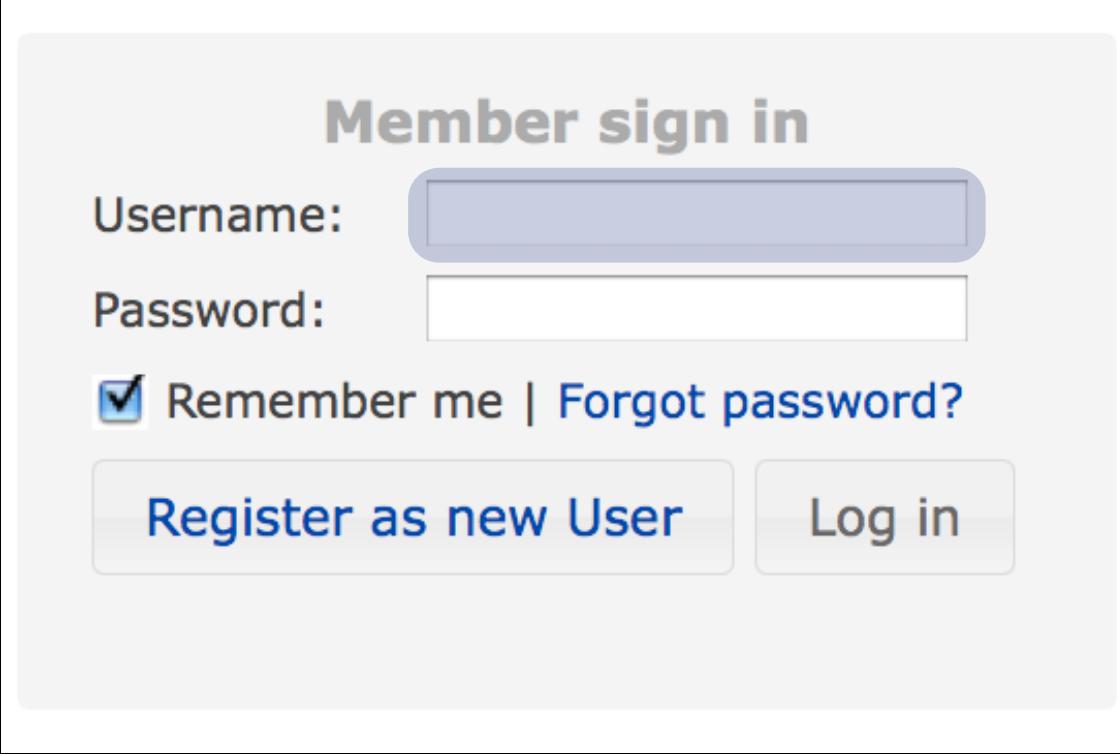
Member sign in

Username:

Password:

Remember me | [Forgot password?](#)

[Register as new User](#) [Log in](#)



```
$(“form”, name: “loginForm”).j_username
```

# GEB FORM INTERACTION

Member sign in

Username:

Password:

Remember me | [Forgot password?](#)

[Register as new User](#) [Log in](#)

```
$(“form”, name: “loginForm”).j_password
```

# GEB FORM INTERACTION

Member sign in

Username:

Password:

Remember me | [Forgot password?](#)

[Register as new User](#) [Log in](#)

```
$(“a#loginButton”)
```

# MORE FORM INTERACTION

```
$("form").conference == "RWX"
$("form").conference = "PAX"
assert $("form").conference == "PAX"

$("form").mySelect = ["Rich", "Web", "Experience"]
$("form").myCheckbox = true
$("form").myRadioButton = "RWX"
$("form").myTextArea = "Project \n Automation \n Experience"
```

```
package com.deepsouthsoftware.execspec.pages

import geb.Page

class LoginPage extends Page {
    static url = "login/auth"

    static at = { title == "Login" }

    static content = {
        loginForm { $("form", name: "loginForm") }
        loginButton(to: PaymentPage) { $("a#loginButton") }
    }
}
```

```
and: "Customer clicks on 'Continue'"  
at ReserveSpeakerPage  
continueToSignInOrCreateAccount.click()  
  
and: "Customer enters username, password, and clicks 'Log in'"  
at LoginPage  
loginForm.j_username = "joeuser"  
loginForm.j_password = "password"  
loginButton.click()  
  
then: "If we'd implemented payment, we'd be ready to pay."  
at PaymentPage
```

```
package com.deepsouthsoftware.execspec.specs

import geb.spock.GebSpec
import com.deepsouthsoftware.execspec.pages.*

class FindSpeakerFlowSpec extends GebSpec {

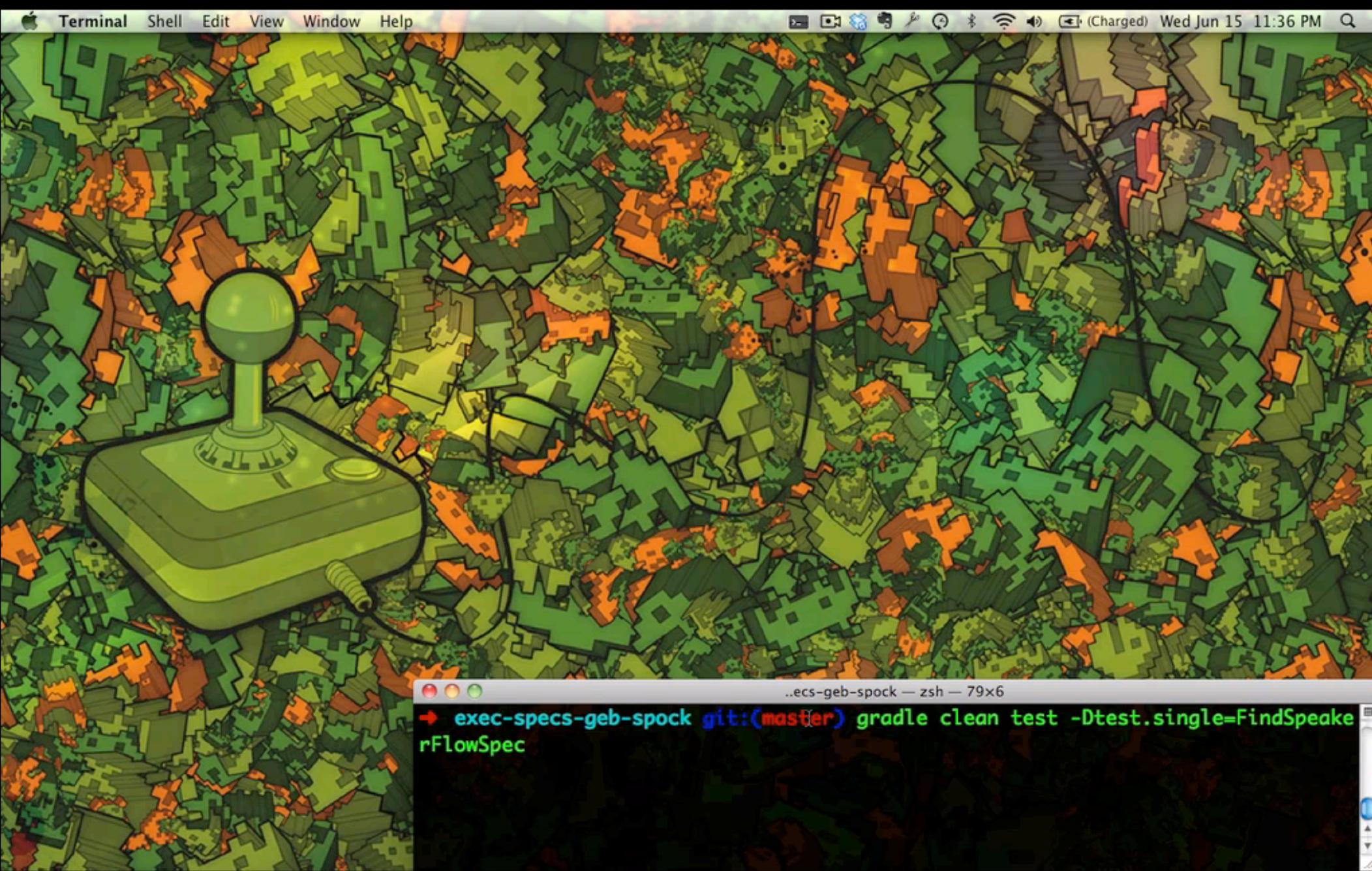
    String getBaseUrl() {
        "http://localhost:8080"
    }

    def "test find speaker flow"() {
        given: "Customer is at the Fluffbox Home Page"
        to HomePage

        when: "Customer clicks on 'FIND A SPEAKER'"
        findASpeaker.click()

        and: "Customer clicks on 'Matt Stine'"
        at FindASpeakerPage
        mattStine.link.click()
    }
}
```

```
and: "Customer clicks on 'Find this Speaker'"  
at MattStineDetailPage  
findThisSpeaker.link.click()  
  
and: "Customer clicks on 'RENT NOW'"  
at FindKioskBySpeakerPage  
rentNow.link.click()  
  
and: "Customer clicks on 'Continue'"  
at ReserveSpeakerPage  
continueToSignInOrCreateAccount.click()  
  
and: "Customer enters username, password, and clicks 'Log in'"  
at LoginPage  
loginForm.j_username = "joeuser"  
loginForm.j_password = "password"  
loginButton.click()  
  
then: "If we'd implemented payment, we'd be ready to pay."  
at PaymentPage  
}  
}
```



AS A CUSTOMER, I WANT  
TO BROWSE THE LIST OF  
KIOSKS IN MY AREA, CHOOSE  
A KIOSK, VIEW SPEAKERS  
AVAILABLE AT THAT KIOSK  
AND THEN CHOOSE ONE SO  
THAT I CAN RENT HIM OR  
HER.

# ACCEPTANCE TEST

- Given the customer is on the Fluffbox Home Page
- When the customer clicks “FIND A FLUFFBOX”
- And the customer searches for kiosks in Ft. Lauderdale”
- And the customer selects the first kiosk in the list
- And the customer clicks “Matt Stine”
- And the customer clicks “RENT NOW!”
- And the customer clicks “Continue”
- And the customer enters his username, password, and then clicks “Log in”
- Then if we had implemented payment, we’d be ready to pay!

```
package com.deepsouthsoftware.execspec.specs

import geb.spock.GebSpec
import com.deepsouthsoftware.execspec.pages.*

class FindKioskFlowSpec extends GebSpec {

    String getBaseUrl() {
        "http://localhost:8080"
    }

    def "test find kiosk flow"() {
        given: "Customer is at the Fluffbox Home Page"
        to HomePage

        when: "Customer clicks on 'FIND A FLUFFBOX'"
        findAFluffbox.click()

        and: "Customer searches for kiosks in Ft. Lauderdale"
        at FindAFluffboxPage
        searchForKiosks('Ft. Lauderdale')
    }
}
```

```
package com.deepsouthsoftware.execspec.pages

import geb.Page
import com.deepsouthsoftware.execspec.modules.FirstKioskModule

class FindAFluffboxPage extends Page {
    static url = "kiosk/find"

    static at = { title == "Find a Fluffbox" }

    def searchForKiosks(def where) {
        searchKiosksForm.searchCriteria = where
        searchButton.click()
        waitFor(5, 0.5) { findSpeakersHere.link.present }
    }

    static content = {
        searchKiosksForm { $('form', name: 'searchKiosksForm') }
        searchButton { $('input', name: 'searchButton') }
        findSpeakersHere(to: FindSpeakerByKioskPage) {
            module FirstKioskModule
        }
    }
}
```

```
and: "Customer selects the first kiosk"
findSpeakersHere.link.click()

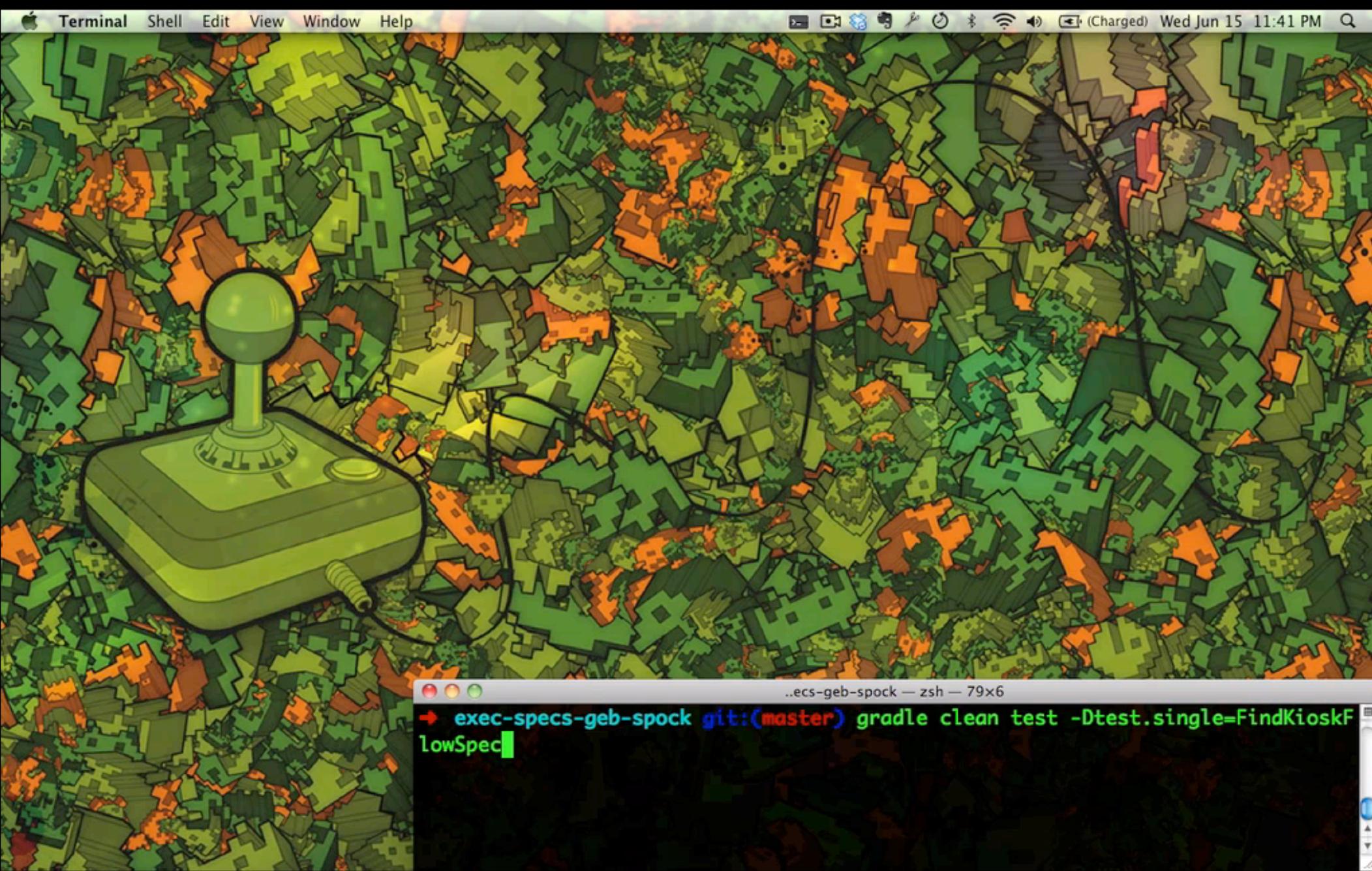
and: "Customer clicks on 'Matt Stine'"
at FindSpeakerByKioskPage
mattStine.link.click()

and: "Customer clicks 'RENT NOW!''"
at MattStineClonePage
rentNow.link.click()

and: "Customer clicks on 'Continue'"
at ReserveSpeakerPage
continueToSignInOrCreateAccount.click()

and: "Customer enters username, password, and clicks 'Log in'"
at LoginPage
loginForm.j_username = "joeuser"
loginForm.j_password = "password"
loginButton.click()

then: "If we'd implemented payment, we'd be ready to pay."
at PaymentPage
```



```
..ecs-geb-spock — zsh — 79x6
→ exec-specs-geb-spock git:(master) gradle clean test -Dtest.single=FindKioskForLowSpec
```

**AS A CUSTOMER, WHEN I  
CLICK ON A SPEAKER'S  
NAME, I SHOULD GET A  
SPEAKER DETAILS PAGE SO  
THAT I CAN SEE HIS OR  
HER TITLE.**

# DATA-DRIVEN ACCEPTANCE TEST

- Given the customer is on the Fluffbox Home Page
- When the customer clicks “FIND A SPEAKER”
- And the customer clicks on a speaker’s name
- Then the customer should see the speaker’s title

```
def "test find speaker flow"() {
    given: "Customer is at the Fluffbox Home Page"
    to HomePage

    when: "Customer clicks on 'FIND A SPEAKER'"
    findASpeaker.click()

    and: "Customer clicks on a speaker's name"
    at FindASpeakerPage
    speakerLink(speaker).click()

    then: "Customer should see speaker's title"
    speakerTitle == title

    where:
    speaker           | title
    "Craig Walls"    | "Author of Spring in Action"
    "Venkat Subramaniam" | "Founder of Agile Developer, Inc."
    "Brian Sletten"   | "Forward Leaning Software Engineer"
    "Ken Sipe"        | "Gradleware CT0"
}
```

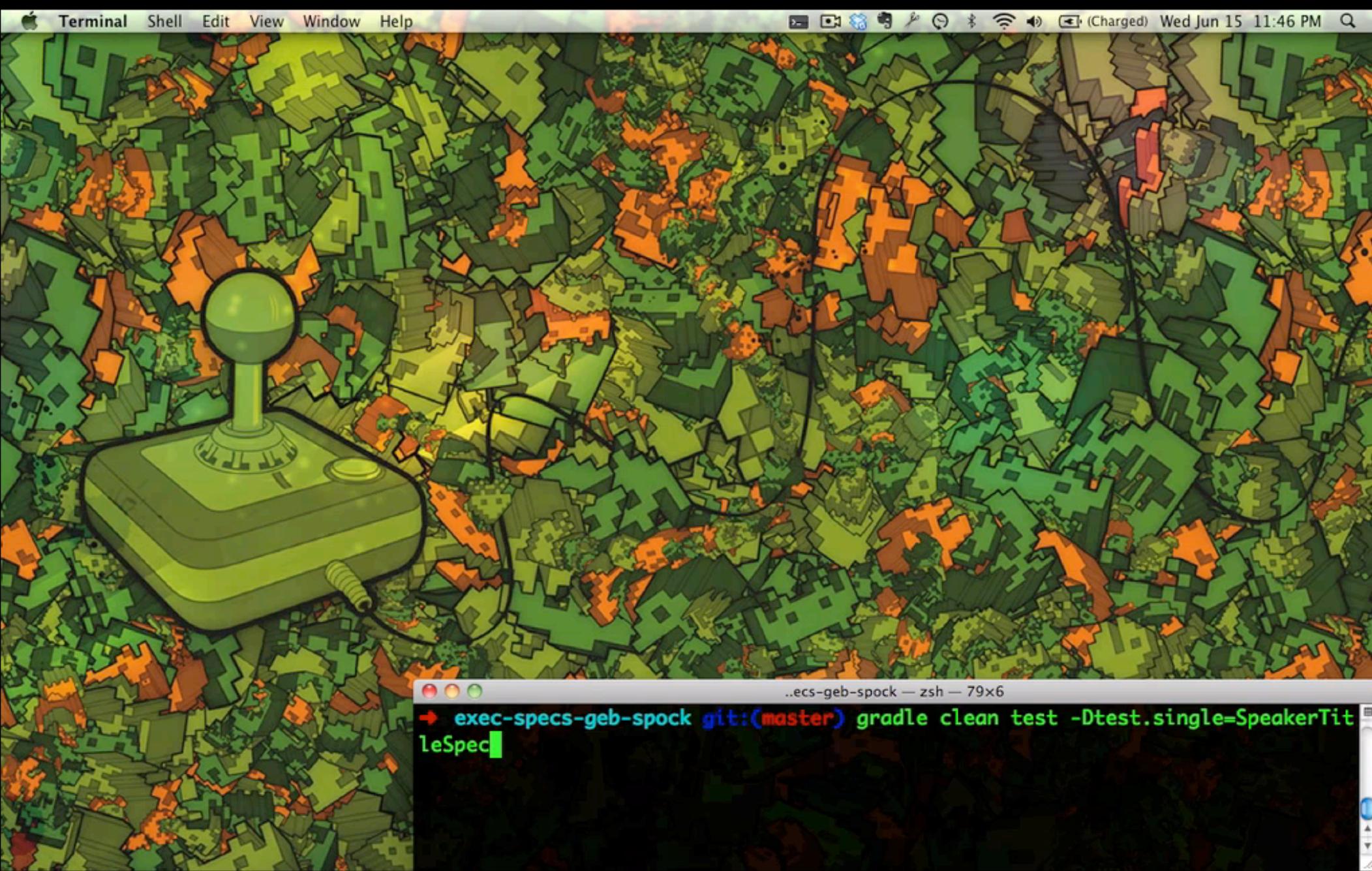
```
package com.deepsouthsoftware.execspec.pages

import geb.Page
import com.deepsouthsoftware.execspec.modules.MattStineModule

class FindASpeakerPage extends Page {
    static url = "speaker/find"

    static at = { title == "Find a Speaker" }

    static content = {
        mattStine(to: MattStineDetailPage) { module MattStineModule }
        speakerLink(to: SpeakerDetailPage) { name -> $('a', text: name) }
    }
}
```



**EXCEL- DRIVEN SPEC**

Screenshot of Microsoft Excel showing a table with four rows of data. The table has two columns: 'Name' and 'Description'. The 'Name' column contains names of authors, and the 'Description' column contains their professional roles.

	A	B
1	Craig Walls	Author of Spring in Action
2	Venkat Subramaniam	Founder of Agile Developer, Inc.
3	Brian Sletten	Forward Leaning Software Engineer
4	Nathaniel Schutta	Author, speaker, software engineer focused on user interface design.
5		
6		

```
class ExcelDrivenSpec extends GebSpec {  
    @Shared dataSet = []  
  
    def setupSpec() {  
        def excelInputStream = getClass()  
            .getResourceAsStream("/DataDrivenSpec.xls")  
        def wb = new HSSFWorkbook(excelInputStream)  
        def sheet = wb.getSheetAt(0)  
  
        sheet.rowIterator().each { row ->  
            def dataSetRow = []  
  
            dataSetRow << row.getCell(0).stringCellValue  
            dataSetRow << row.getCell(1).stringCellValue  
  
            dataSet << dataSetRow  
        }  
    }  
  
    String getBaseUrl() {  
        "http://localhost:8080"  
    }  
}
```

```
def "test find speaker flow"() {
    given: "Customer is at the Fluffbox Home Page"
    to HomePage

    when: "Customer clicks on 'FIND A SPEAKER'"
    findASpeaker.click()

    and: "Customer clicks on a speaker's name"
    at FindASpeakerPage
    speakerLink(speaker).click()

    then: "Customer should see speaker's title"
    speakerTitle == title

    where:
        [speaker, title] << dataSet
    }
}
```

**DB- DRIVEN SPEC**

```
class DatabaseDrivenSpec extends GebSpec {
    @Shared sql = Sql.newInstance("jdbc:h2:mem:", "org.h2.Driver")

    def setupSpec() {
        sql.execute("create table speakers (id int primary key, speaker varchar, title varchar)")
        sql.execute("insert into speakers values (1, 'Craig Walls', 'Author of Spring in Action')")
        sql.execute("insert into speakers values (2, 'Venkat Subramaniam', 'Founder of Agile Developer, Inc.')")
        sql.execute("insert into speakers values (3, 'Brian Sletten', 'Forward Leaning Software Engineer')")  
        sql.execute("insert into speakers values (4, 'Ken Sipe', 'Gradleware CTO')")
    }

    String getBaseUrl() {
        "http://localhost:8080"
    }
}
```

```
def "test find speaker flow"() {
    given: "Customer is at the Fluffbox Home Page"
        to HomePage

    when: "Customer clicks on 'FIND A SPEAKER'"
        findASpeaker.click()

    and: "Customer clicks on a speaker's name"
        at FindASpeakerPage
        speakerLink(speaker).click()

    then: "Customer should see speaker's title"
        speakerTitle == title

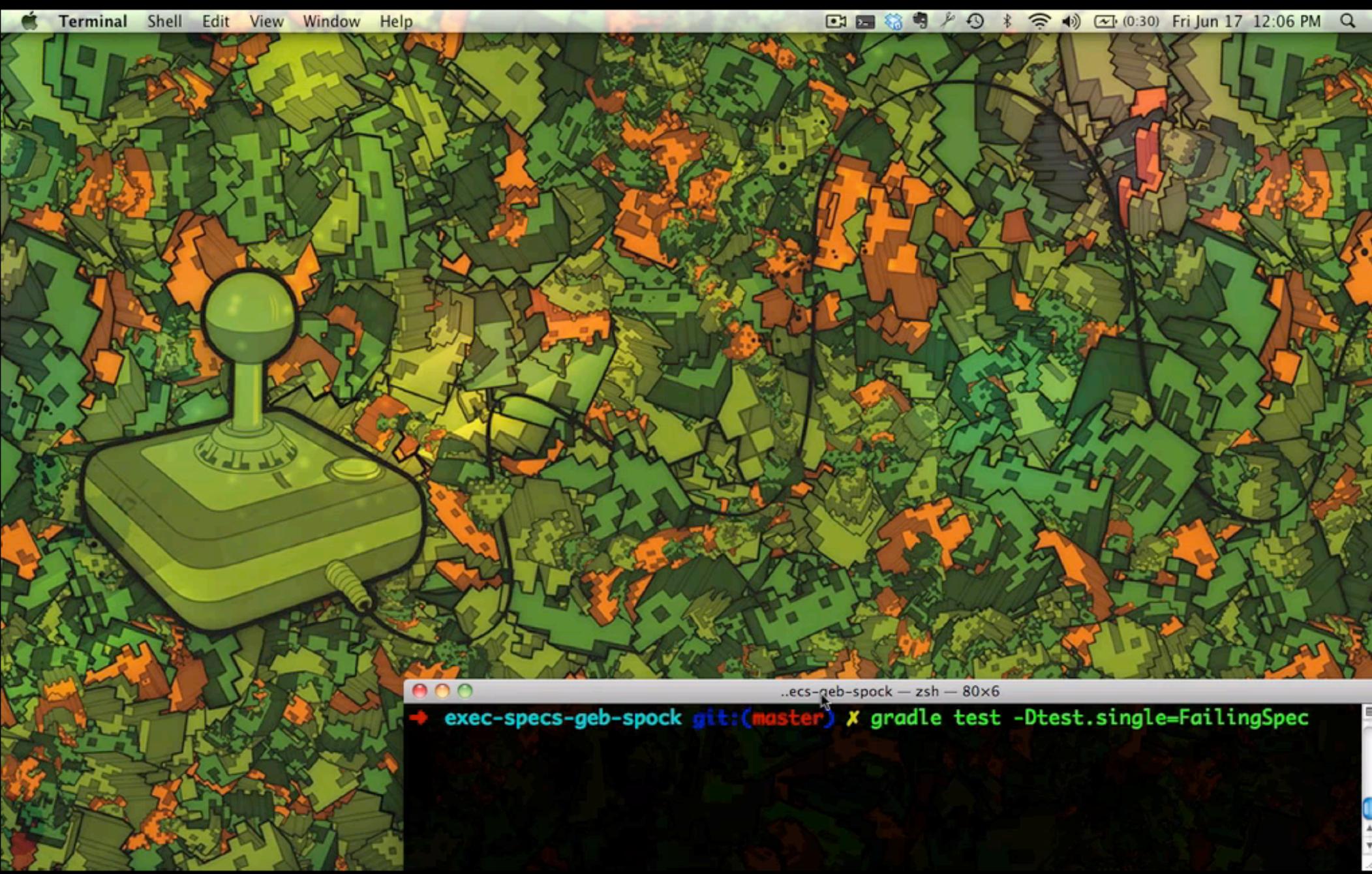
    where:
        [speaker, title] << sql.rows("select speaker, title from speakers")
}

}
```

**WHEN SPECS FAIL...**

A screenshot of a Microsoft Word document titled "DataDrive". The ribbon menu is visible at the top, showing tabs for Home, Layout, Tables, Charts, SmartArt, Formulas, Data, and Review. The Home tab is selected. The font toolbar below the ribbon shows "Calibri (Body)" and "12". The table below has 7 columns labeled A through G. Row 1 contains "Craig Walls" in column A and "Java EE Spec Lead" in column B. Row 2 contains "Venkat Subramaniam" in column A and "Founder of Waterfall, Inc." in column B. Row 3 contains "Brian Sletten" in column A and "Forward Leaning Software Engineer" in column B. Row 4 contains "Nathaniel Schutta" in column A and "Leading JSF Consultant" in column B. Row 5 is empty.

	A	B	C	D	E	F	G
1	Craig Walls	Java EE Spec Lead					
2	Venkat Subramaniam	Founder of Waterfall, Inc.					
3	Brian Sletten	Forward Leaning Software Engineer					
4	Nathaniel Schutta	Leading JSF Consultant					
5							



..ecs-neb-spock — zsh — 80x6

```
exec-specs-geb-spock git:(master) ✘ gradle test -Dtest.single=FailingSpec
```

## Tests

Name	Status	Type	Time(s)
test find speaker flow	Failure	<p><b>Condition not satisfied: speakerTitle == title         Java EE Spec Lead   false   21 differences (19% similarity)   (Author) (of) Sp(ring) (in Action)   (Java--) (EE) Sp(ec--) (Lead-----) Author of Spring in Action</b></p> <p>Condition not satisfied:</p> <pre>speakerTitle == title         Java EE Spec Lead   false   21 differences (19% similarity)   (Author) (of) Sp(ring) (in Action)   (Java--) (EE) Sp(ec--) (Lead-----)   Author of Spring in Action</pre> <p>at com.deepsouthsoftware.execspec.specs.FailingSpec.test find speaker flow(FailingSpec.groovy:43)</p> <p><b>Condition not satisfied: speakerTitle == title         Founder of Waterfall, Inc.   false   13 differences (59% similarity)   Founder of (Agil)e( Deve)l(oper), Inc.   Founder of (Wat-)e(rfal-)l(----), Inc. Founder of Agile Developer, Inc.</b></p> <p>Condition not satisfied:</p> <pre>speakerTitle == title         Founder of Waterfall, Inc.   false   13 differences (59% similarity)   Founder of (Agil)e( Deve)l(oper), Inc.   Founder of (Wat-)e(rfal-)l(----), Inc.   Founder of Agile Developer, Inc.</pre> <p>at com.deepsouthsoftware.execspec.specs.FailingSpec.test find speaker flow(FailingSpec.groovy:43)</p>	12.367

# SUMMARY

- Acceptance criteria described in terms of UI
- Automated UI tests considered brittle
- Manual UI tests impossible to keep in-synch
- But we **MUST TEST!**
- Spock+Page Objects+Geb = **VIA  
BLE  
SOLUTION**

# CODE SAMPLES

- Case Study App (Fluffbox): <https://github.com/mstine/fluffbox-rwx>
- Geb/Spock Samples: <https://github.com/mstine/exec-specs-geb-spock>



MISSION ACCOMPLISHED  
CAPTAIN.

Matt Stine

[matt.stine@gmail.com](mailto:matt.stine@gmail.com)

Twitter: mstine

<http://www.mattstine.com>

PLEASE FILL OUT YOUR EVALUATIONS!