

Cucumber and Capybara

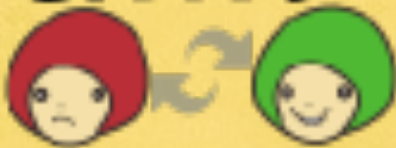
A commons case study

old vs new

	old	new
testing-framework	test-unit v1	cucumber
browser-driver	pure selenium v1	capybara


Cucumber behaviour driven development
with elegance and joy

VS

Test::Unit




Plain text scenarios with features

```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers

  Scenario: Add two numbers
    Given I have entered 50 into the calculator
    And I have entered 70 into the calculator
    When I press add
    Then the result should be 120 on the screen
```

Step definitions

```
Given /I have entered (.*) into the calculator/ do |n|
  calculator = Calculator.new
  calculator.push(n.to_i)
end
```

shamelessly stolen from cukes.info

test-unit: naming

functional area	Test Cases	Test Steps	Expected Result	Automated Test Coverage
				Automated:10.7055 SimpleTest:0
Gardener				
Gardener Frontpage				
	Tour Gardens Pages	1. Click on each link of 'Tour Gardens' (Features, Themebuilder, Pricing, FAQ, Help)	1. Make sure that the links are working and pages are appearing with no css or general issues	
	Community Pages	1. Click on each link of 'Community' (Example sites, Drupal Gardens blog, Forums, SWAG, Drupal 7 Thank You, Customer quotes)	1. Make sure that the links are working and pages are appearing with no css or general issues	

+

```
class Test9999999CruffySessionSystemTest < Test::Unit::TestCase
def test_00_bork
  ...
end
```

cucumber: naming

Feature: Groups

Scenario: As a user I can join an existing group

...

- activity_stream.feature
- blog.feature
- commons_setup.feature
- discussion.feature
- document.feature
- events.feature
- groups.feature**
- homepage.feature
- login_logout.feature
- polls.feature
- profile.feature
- site_analytics.feature
- wiki.feature

test-unit: tagging

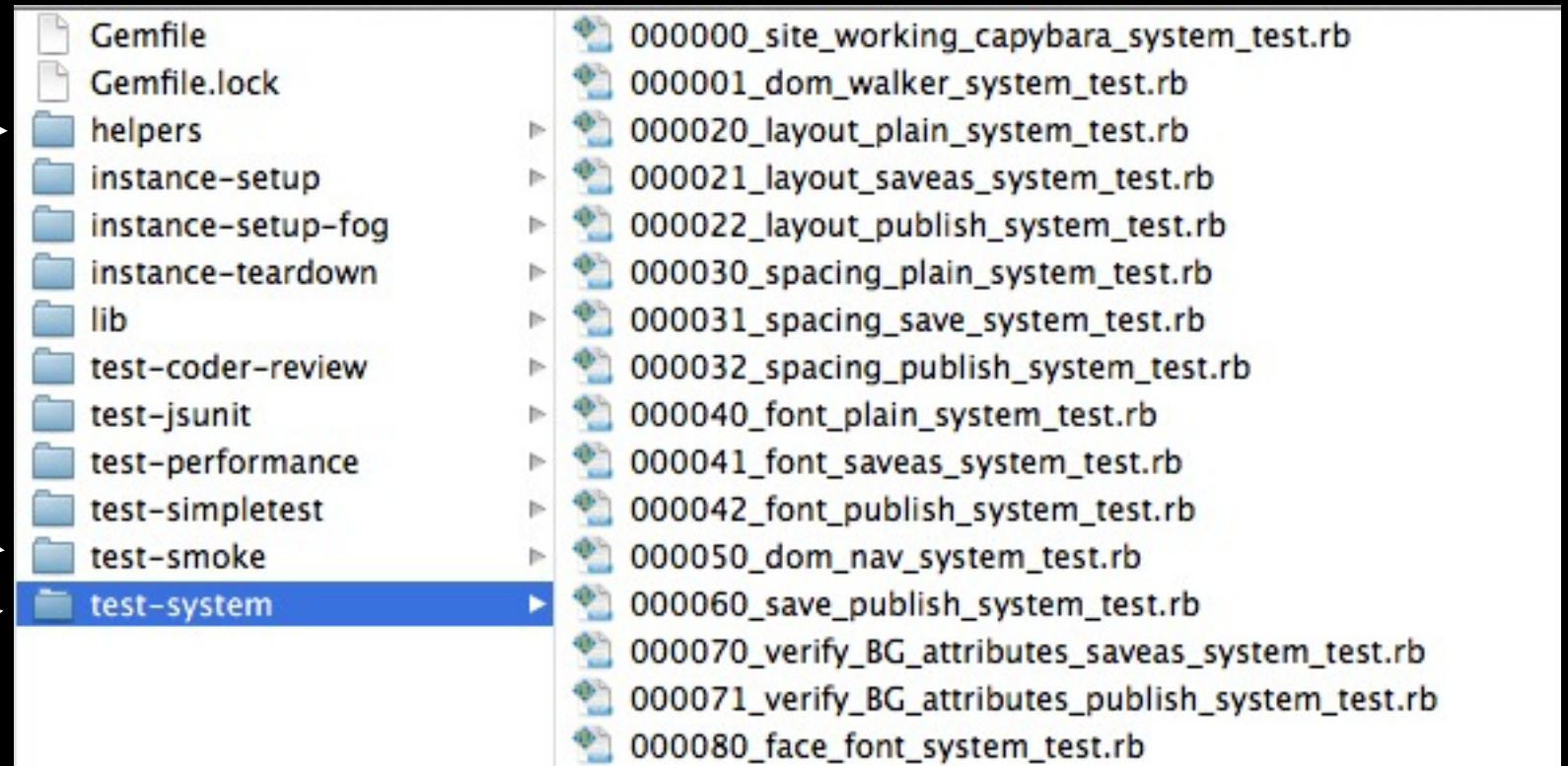
Shared code



Smoke tests



System tests



cucumber: tagging

```
Scenario: As a user I can comment on a blog post and on comme
  Given I am logged in as admin
  And I create a blog entry with the title "Julius Caesar" an
  say?"
  And I am logged in as "derpington" with the password "zomgb
  And I go to '/blog/julius-caesar'
  And I add the comment "Friends, Romans, countrymen, lend me
  When I am logged in as admin
  And I go to '/blog/julius-caesar'
  Then I should see 'Friends, Romans, countrymen'
  And I reply to the first comment with "... and I will sing
  And I go to '/blog/julius-caesar'
  Then I should see 'and I will sing you a song'
```

```
# https://backlog.acquia.com/browse/SP-912
```

@wip

```
Scenario: As a user I can mark a blog post as "Featured"
  Given I create a blog entry with the title "Test featured b
  marked featured?"
  And I mark this content as featured
```

Actual
tags



test-output: test-unit

[parallel](#) » [#527](#) » [Test Results](#)

Test Result

19 failures (+1)



All Failed Tests

Test Name
<<< Test000020LayoutPlain.test 0020 select this page
Error Details
The single-page layout for node <code>"/node/2"</code> is supposed to be <code>"cb"</code> but it is <code>"ca"</code> .

test-output: cucumber

Test Result

15 failures (±0)



All Failed Tests

Test Name
<u><<< Blog.As a user I can mark a blog post as "Featured".As a user I can mark a blog post as "Featured"</u>
Error Details
<code>failed As a user I can mark a blog post as "Featured"</code>

test-output: cucumber

Background

- Given a fresh commons installation features/step_definitions/common_steps.rb:128
- Given I am logged in as admin features/step_definitions/login_logout_steps.rb:14
- Using /devel/switch/commonsadmin to log in

Scenario: A user doesn't want errors on the admin status page features/commons_setup.feature:6

- Given I am on the status report page features/step_definitions/common_steps.rb:10
- Then I shouldn't see any items with an error icon features/step_definitions/commons_setup_steps.rb:1

```
expected: []
got: ["Drupal core update status Not secure! (version 6.24 available)", "There is a security update available for your version of Drupal. To ensure the security of your server, you should update immediately! See the Drupal security advisory for more information."]
Diff:
@@ -1,2 +1,3 @@
-[]
+["Drupal core update status Not secure! (version 6.24 available)",
+ "There is a security update available for your version of Drupal. To ensure the security of your server, you should update immediately! See the
+ (RSpec::Expectations::ExpectationNotMetError)

./features/step_definitions/commons_setup_steps.rb:3:in `^I shouldn't see any items with an error icon$/'
features/commons_setup.feature:8:in `Then I shouldn't see any items with an error icon'

1 Then /^I shouldn't see any items with an error icon$/ do
2   error_items = page.all(:css, 'tr.error').map{|item| item.text}
3   error_items.should == []
4 end
5
```

- And I should see a lot of items with an ok icon features/step_definitions/commons_setup_steps.rb:6

Screenshot of the error



The screenshot shows the Commons QA admin interface. At the top, there's a navigation bar with the Commons logo and the text 'Commons QA'. On the right, it says 'Welcome back commonsadmin'. Below the navigation bar, there's a search bar with the text 'Entire site' and a 'SEARCH' button. In the main content area, there's a notification banner that reads: 'Drupal core update status Not secure! (version 6.24 available). There is a security update available for your version of Drupal. To ensure the security of your server, you should update immediately! See the Drupal security advisory for more information.' Below the notification, there's a table with columns for 'Status', 'Title', and 'Description'. The table contains one row with a red error icon in the 'Status' column and the text 'Drupal core update status Not secure! (version 6.24 available)' in the 'Title' column. Below the table, there's a breadcrumb trail: 'Home > Administer > Reports'.

test-unit: setup

- setup/teardown
- in v2: self.startup / self.shutdown
(no easy access to instance variables)
- no predefined integration with selenium

cucumber: setup

- Scenario backgrounds
- Scenario hooks
 - Before/After/Around
(all of them can use tags)
 - Fully integrated with capybara
(@selenium, @webkit, @...)

<https://github.com/cucumber/cucumber/wiki/Hooks>

test-unit: code

```
def test_01_create_basic_webform
  Log.logger.info("Starting test_01_create_basic_webform")
  @browser.open('/')
  login(@user.name, @user.password)
  self.make_sure_webform_is_enabled()
  webformmgr = WebformManager.new(@browser)
  @browser.open("#{webformmgr.create_webform_url}")
```

cucumber code: features

Feature: Blog

Background:

Given a fresh commons installation

Given a user named "derpington" with the password "zomgbbq"

Given I am logged in as "derpington" with the password "zomgbbq"

Given I have joined the default group

Scenario Outline: As a user I can create a blog post

Given I create a blog entry with the title "<TitleText>" and the body "<BodyText>"

Then I should see a blog entry with "<BodyText>" in it

And I should see a headline with "<TitleText>" in it

Examples:

TitleText	BodyText	
test title uno	This is a test body	
Nön ÄSCîî titlé	umop əpɪsdn ʊ,ɪ 'əʊ ʔə xool	

cucumber code: features

Feature: Blog

Background:

Given a fresh commons installation

Given a user named "derpington" with the password "zomgbbq"

Given I am logged in as "derpington" with the password "zomgbbq"

Given I have joined the default group



Scenario Outline: As a user I can create a blog post

Given I create a blog entry with the title "<TitleText>" and the body "<BodyText>"

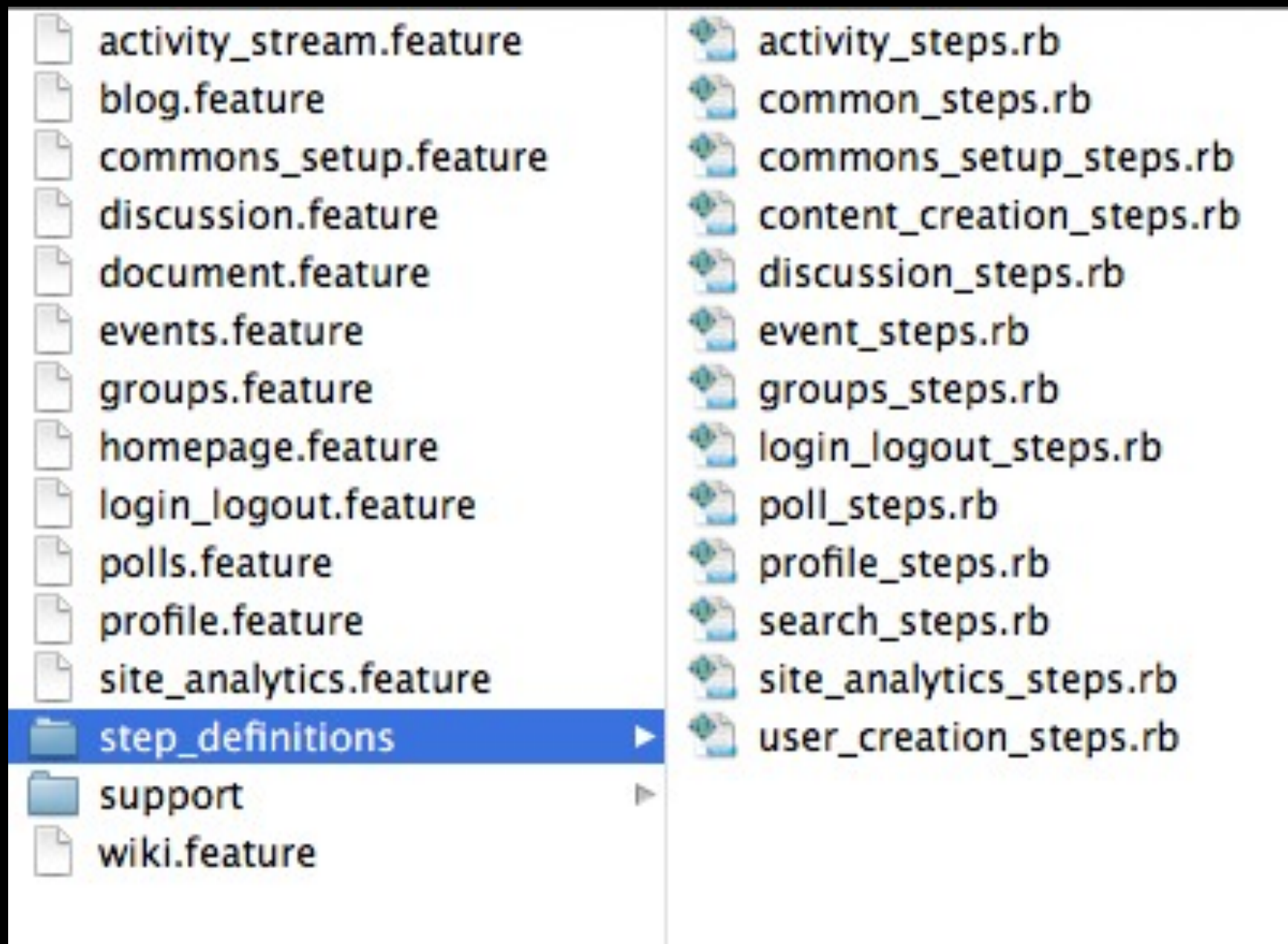
Then I should see a blog entry with "<BodyText>" in it

And I should see a headline with "<TitleText>" in it

Examples:

TitleText	BodyText	
test title uno	This is a test body	
Nön ÄSCîî titlé	umop əpɪsdn ʊ,ɪ 'əʊ ʔə ɹool	

cucumber code: step definitions



cucumber code: step definitions

```
Given /^I have joined the default group/ do
  visit('/og/subscribe/1?destination=groups')
  click_button('Join')
end
```


cucumber code: step definitions

Simple ones

```
And /^I edit the current content$/ do
  within(:css, 'div.content-tabs-inner'){ click_link('Edit') }
end
```

cucumber code: step definitions

Variables

```
Then /^I should see the image ['"](.*)['"]$/ do |image_url|  
  page.should have_css("img[src='#{image_url}']")  
end
```

```
And /I should see a link with the text ['"](.*)['"]/ do |text|  
  page.should have_xpath("//a[contains(text(), text)]")  
end
```

cucumber: step definitions

Combining steps

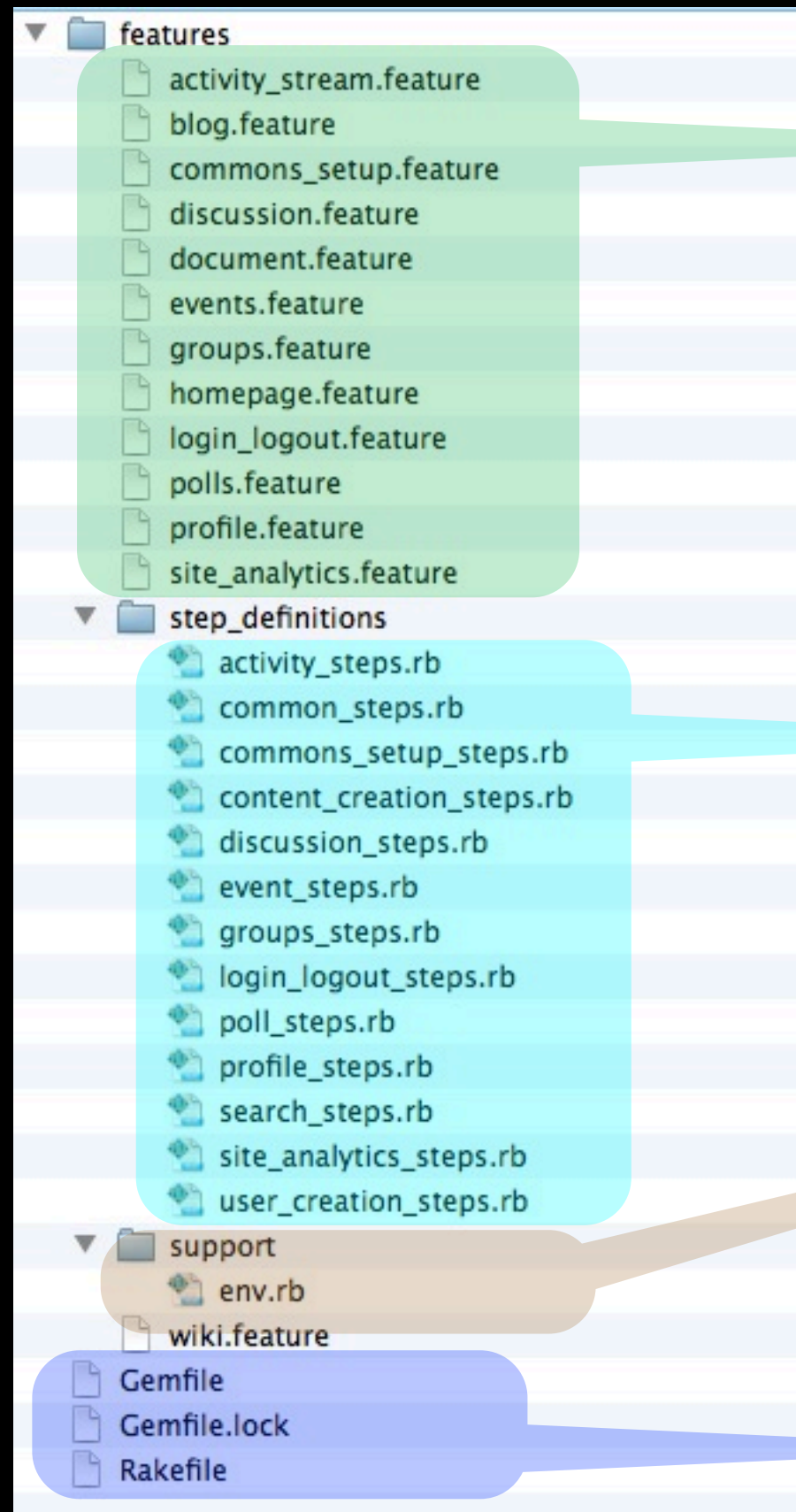
```
And /^I add the comment ["'](.*)["']$/ do |text|  
  step "I click on 'Comment'"  
  step 'I disable the rich-text editor'  
  step "I fill in 'Comment' with '#{text}'"  
  step "I press 'Save'"  
end
```

cucumber: step definitions

Advanced steps

```
#The ?: tells us that we don't want to capture that part in a variable
When /^(?:I am|I'm|I) (?:on|viewing|looking at|look at|go to|visit|visiting) ["']?([^"']*?)["']?$/ do |path|
  translation_hash = {
    "the status report page" => '/admin/reports/status',
    "the blog posts page" => '/content/blogs',
    "the blog post page" => '/content/blogs',
    [...]
    'the bookmarks page' => '/bookmarks',
  }
  if translation_hash.key?(path)
    visit(translation_hash[path])
  else
    if path.start_with?("/")
      visit(path)
    else
      raise "I don't know how to go to this path: #{path.inspect}."
    end
  end
end
end
```


File Layout



Features:
The test descriptions

Step definitions:
Mapping text to code

env.rb:
Setting up the environment

Gemfile: Which gems?
Gemfile.lock: Which versions?
Rakefile: Misc tasks

File Layout: env.rb?

env.rb does these things:

- it loads Bundler
- it loads Capybara
 - ... and sets the default driver
- It loads the capybara-screenshot gem
- it launches XVFB
- it populates the \$site_capabilities hash
 - determines whether or not we have the devel module available

General usage

- Run one feature:
\$ cucumber features/blog.feature
- Run the specific scenario at line 42:
\$ cucumber features/blog.feature:42

Other nifty things

- `cucumber --dry-run:`
Allows to check for missing step definitions
- `cucumber --format usage:`
Allows to figure out which steps get called the most or which steps don't get called. Also tells you which steps take the longest
- `cucumber --format rerun:`
Saves failed tests to `rerun.txt` and allows to rerun just those failed tests
- `cucumber --tags @wip:`
Will only run tests tagged `@wip`.
Also possible: `--tags ~@wip` to NOT run them

Code smells

(we have some of those)

- Try to abstract the actual implementation of the steps out of the scenarios
 - Good:
“Given I am logged in as an administrator”
 - Bad:
Given I go to “/login”
And I enter “admin” in the “username” field
And I enter “foobar” in the “password” field
[...]



Capybara
vs
Selenium



Capybara and Selenium

Selenium

- An API
- Bindings for actual browsers (IE, Chrome, FF, ...)

Capybara:

- An API
- A big set of tests

Capybara drivers:

- Remote controls for browsers and browser simulators that can be “plugged into”

Capybara, usually 3rd party projects

Selenium: setup

- Selenium I needs:
 - An installed browser
 - A running Selenium RC (java app)
 - An X server
- With Saucelabs it needs:
 - A running Sauceconnect process

Problems with Selenium

- Slow: Launching Firefox with a new profile
- Slow: Adding Webdriver extension
- Slow: Communicates over JSON/REST
- Bad: No Console.log output
- Bad: JS Errors are invisible
- Bad: Selenium I has limitations
- Bad: No proper implicit waits

Capybara: setup

Capybara needs:

- A driver

Capybara drivers need:

- selenium webdriver: X Server
- headless webkit: X Server, QT
- poltergeist: X Server, the phantomjs binary
- akephalos: java
- mechanize: no dependencies

Capybara: drivers

	Javascript + DOM	Speed	Stability
Webdriver	10	7	6 (recently)
Headless Webkit	9	9	9
Poltergeist	9 (PhantomJS)	8	5
Akephalos	6 (HTML Unit)	6	8
Mechanize	0	10	10

Capybara API: clicking

```
click_link( 'id-of-link' )  
click_link( 'Link Text' )  
click_button( 'Save' )  
click_on( 'Link Text' )  
click_on( 'Button Value' )
```

Capybara API: forms

```
fill_in('First Name', :with => 'John')  
fill_in('Password', :with => 'Seekrit')  
fill_in('Description', :with => 'Really Long Text...')  
choose('A Radio Button')  
check('A Checkbox')  
uncheck('A Checkbox')  
attach_file('Image', '/path/to/image.jpg')  
select('Option', :from => 'Select Box')
```

Capybara API: querying

```
page.has_selector?('table tr')  
page.has_selector?(:xpath, '//table/tr')  
page.has_no_selector?(:content)
```

```
page.has_xpath('//table/tr')  
page.has_css('table tr.foo')  
page.has_content('foo')
```

Capybara API: rspec matchers

```
page.should have_selector('table tr')  
page.should have_selector(:xpath, '//table/tr')  
page.should have_no_selector(:content)
```

```
page.should have_xpath('//table/tr')  
page.should have_css('table tr.foo')  
page.should have_content('foo')
```


Capybara API: finding

```
find_field('First Name').value  
find_link('Hello').visible?  
find_button('Send').click
```

```
find(:xpath, "//table/tr").click  
find("#overlay").find("h1").click  
all('a').each { |a| a[:href] }
```

Capybara API: scoping

```
find( '#navigation' ).click_link( 'Home' )  
find( '#navigation' ).should have_button( 'Sign out' )
```

```
within( "li#employee" ) do  
  fill_in 'Name', :with => 'Jimmy'  
end
```

```
within( :xpath, "//li[@id='employee']" ) do  
  fill_in 'Name', :with => 'Jimmy'  
end
```


Capybara API: AJAX?

```
Capybara.default_wait_time = 5
```

```
click_link('foo')  
#Ajax stuff happens that adds "bar"  
click_link('bar')  
#Ajax stuff happens that adds "baz"  
page.should have_content('baz')
```

Capybara: Heads up for Ajax!

Bad

```
!page.has_xpath?( 'a' )
```

Good

```
page.has_no_xpath?( 'a' )
```

In Selenium: AJAX :(

```
wait = Selenium::WebDriver::Wait.new(:timeout => 5)
btn = wait.until {
  @browser.find_element(:xpath => @contmgr.sort_asc)
}
```

Note: PHP

Behat

Cucumber in PHP

<http://behat.org/>

Mink

Capybara in PHP

<http://mink.behat.org/>

Capybara Cucumber

 2,215  355

 2,943  389

Behat

 342  43

Mink

 229  72

Any questions?

