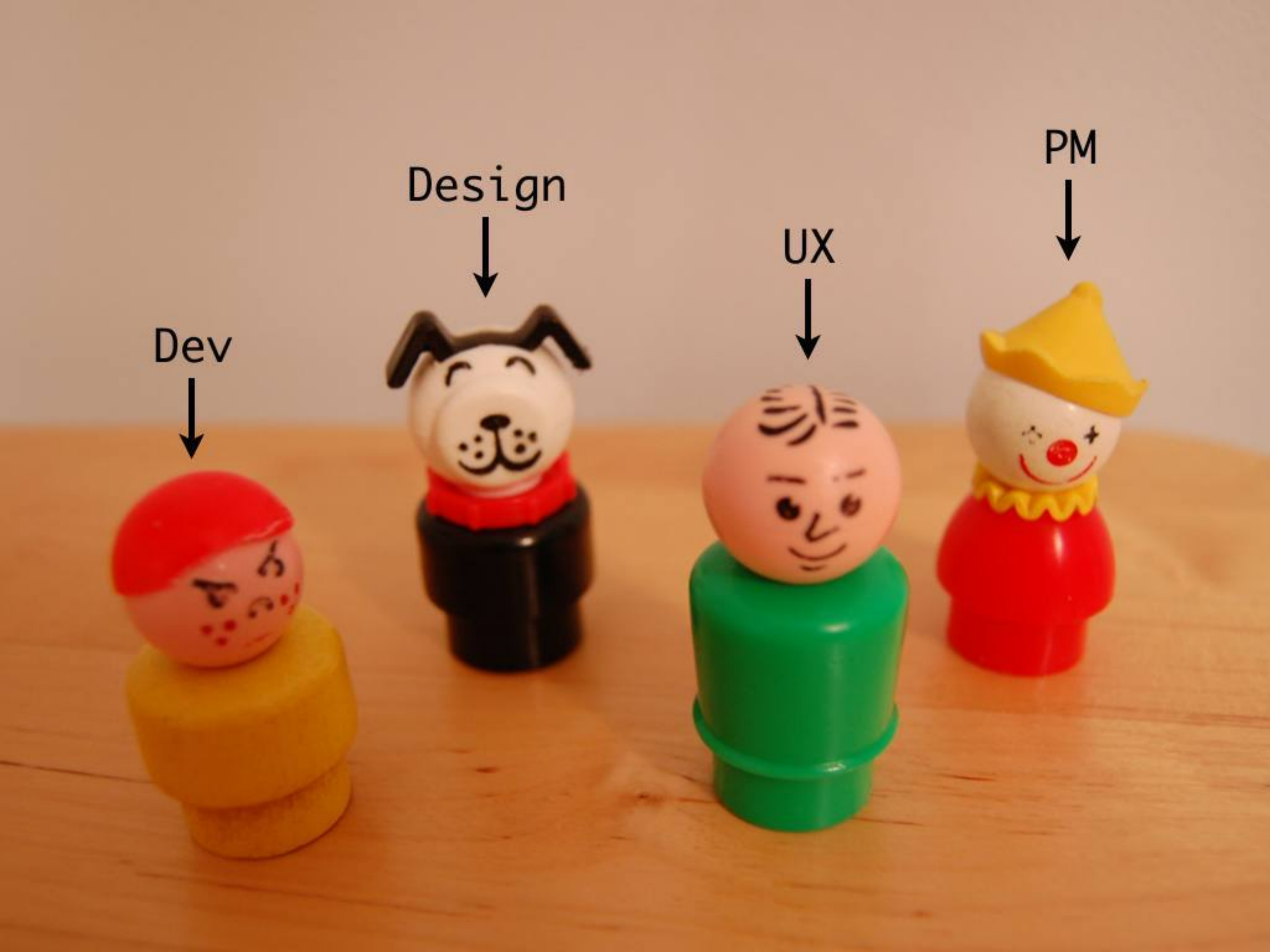
Make Everyone a Tester

Natural Language Acceptance Testing




Dev
Design
UX
PM

QA

QA

QA

why test?

ensure quality

communicate

goals

- developer & PM pairing
- tests written first
- drive a web browser

start with a story

```
Users can register for an account

Requires the following information:

* Email
* Password
* Password Confirmation
* Acceptance of terms / conditions

Acceptance Criteria

* Try with a duplicate email address
* Try with a password less than 6 characters
```



image: papazimouris @ flickr

discuss

Users can register for an account

Requires the following information:

* Email
* Password
* Password Confirmation
* Acceptance of terms / conditions

Acceptance Criteria

* Try with a duplicate email address
* Try with a password less than 6 characters

discuss

Users can register for an account

Requires the following information:

* Email
* Password
* Password Confirmation
* Acceptance of terms / conditions



Acceptance Criteria

* Try with a duplicate email address
* Try with a password less than 6 characters

discuss

success state

Users can register for an account

Requires the following information:

form with data fields

* Email
* Password
* Password Confirmation
* Acceptance of terms / conditions

Acceptance Criteria

* Try with a duplicate email address
* Try with a password less than 6 characters

discuss

success state

Users can register for an account

Requires the following information:

form with data fields

* Email
* Password
* Password Confirmation
* Acceptance of terms / conditions

Acceptance Criteria

error states

* Try with a duplicate email address
* Try with a password less than 6 characters

define behavior

- locations
- page state
- message content

```
State:

* No users exist in the system

Steps:

1. Visit the registration page
2. Enter valid registration information
3. Submit the form

Expectation:

* I should be redirected to the home page
* I should see a success message displayed
```

where am I starting from?

```
State:

* No users exist in the system

Steps:

1. Visit the registration page
2. Enter valid registration information
3. Submit the form

Expectation:

* I should be redirected to the home page
* I should see a success message displayed
```

what are the inputs?

```
State:

* No users exist in the system

Steps:

1. Visit the registration page
2. Enter valid registration information
3. Submit the form

Expectation:

* I should be redirected to the home page
* I should see a success message displayed
```

what is the behavior?

Spasibo: Cras Nec Risus

SPASIBO
CRAS NEC RISUS



Register Log In

Account Registration

Already have an account? Log in to your account here.

Email: user@host.com

Password: ••••••••

Confirm Password: ••••••••

I agree to the Terms & Conditions.

Create Account

Terms & Conditions

manual

problems

- time consuming
- prone to error
- no regressions

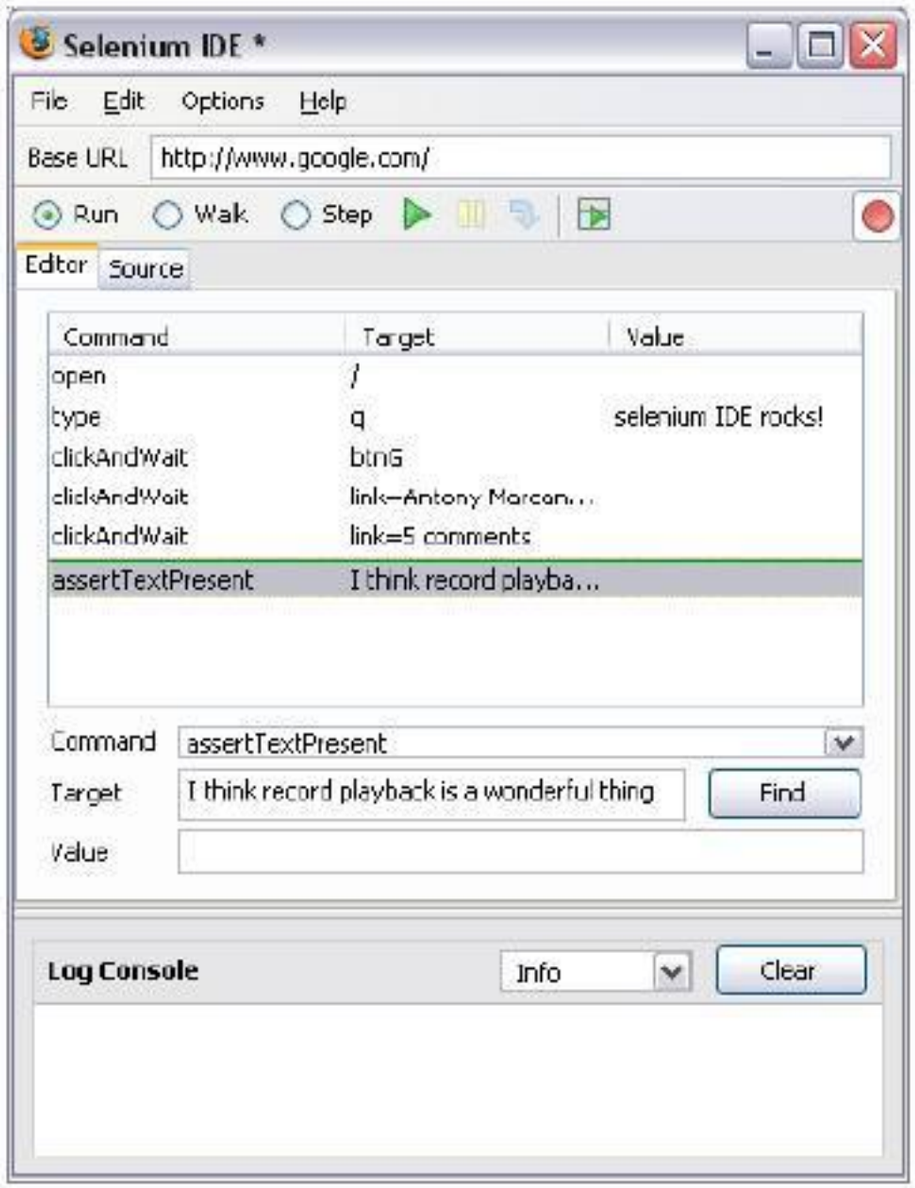
automate!

Note: To use the OpenQA forums, bug tracker, and wiki you need an OpenQA account.

News

June 3, 2008
Selenium IDE 1.0 Beta 2 is available - download now

Selenium IDE is an integrated development environment for Selenium tests. It is implemented as a Firefox extension, and allows you to record, edit, and debug tests. Selenium IDE includes the entire Selenium Core, allowing you to easily and quickly record and play back tests in the actual environment that they will run.
Selenium IDE is not only recording tool: **it is a complete IDE**. You can choose to use its recording capability, or you may edit your scripts by hand. With autocomplete support and the ability to move commands around quickly, Selenium IDE is the ideal environment for creating Selenium tests no matter what style of tests you prefer.



Features:

- Easy record and playback
- Intelligent field selection will use IDs, names, or XPath as needed
- Autocomplete for all common Selenium commands
- Walk through tests
- Debug and set breakpoints
- Save tests as HTML, Ruby scripts, or any other format
- Support for Selenium *user-extensions.js* file
- Option to automatically assert the title of every page

Selenium is a suite of tools to automate web app testing across many platforms.

Selenium...

- runs in many browsers and operating systems
- can be controlled by many programming languages and testing frameworks.

Download Selenium
1.0 beta-2

Ads by Google

Selenium training
Learn the basics of this open source functional testing tool.
www.rtsweb.com

Top 5 QA Testing Tools
2009 Top 5 QA Testing Tools Vendor Rankings. Get Free Report
Business-Software.com/QATestin

Spasibo: Cras Nec Risus

SPASIBO
CRAS NEC RISUS

Edit Account Log Out

Welcome to Spasibo. A winner is you!

Welcome to Spasibo!

This is a site where you can create an account. What can you do after you've creat
account! Is that all? Nope! You can edit that account, so that next time you log in, y

Want to edit your account? Do it!



spasibo-test.html - Selenium IDE

Base URL http://spasibo.com/

Fast Slow

Table Source

Command	Target	Value
type	user_password	password
type	user_password_confi...	password
click	user_terms	
clickAndWait	user_submit	
waitForPageToLoad	5000	
verifyTextPresent	Welcome to Spasibo. ...	
verifyLocation	http://spasibo.com/	

Command verifyLocation

Target http://spasibo.com/ Find

Value

Log Reference UI-Element Rollup Info Clear

[info] Executing: |clickAndWait | user_submit | |
[info] Executing: |waitForPageToLoad | 5000 | |
[info] Executing: |verifyTextPresent | Welcome to Spasibo. A winner is you! | |
[info] Executing: |verifyLocation | http://spasibo.com/ | |

done.

but...

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://spasibo.com/" />
<title>spasibo-test</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">spasibo-test</td></tr>
</thead><tbody>
<tr>
  <td>open</td>
  <td>/</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>link=Register</td>
  <td></td>
</tr>
<tr>
  <td>type</td>
  <td>user_email</td>
  <td>user@host.com</td>
</tr>
<tr>
  <td>type</td>
  <td>user_password</td>
  <td>password</td>
</tr>
<tr>
```

http://spasibo.com – Spasibo: Cras Nec Risus

SPASIBO
CRAS NEC RISUS

Edit Account Log Out

Welcome to Spasibo!

This is a site where you can create an account. What can you do after you've create account! Is that all? Nope! You can edit that account, so that next time you log in, y

Want to edit your account? Do it!

"Comrade Zangief, you have made your country proud and show that our country's spirit can overcome all obstacles."

spasibo-test.html – Selenium IDE

Base URL http://spasibo.com/

Fast Slow

Table Source

Command	Target	Value
open	/	
clickAndWait	link=Register	
type	user_email	user@host.com
type	user_password	password
type	user_password_confi...	password
click	user_terms	
clickAndWait	user_submit	

Command verifyLocation

Target http://spasibo.com/ Find

Value

Log Reference UI-Element Rollup Info Clear

A winner is you! | |
[info] Executing: |verifyLocation | http://spasibo.com/ | |
[info] Executing: |open | / | |
[info] Executing: |clickAndWait | link=Register | |
[error] Element link=Register not found

http://spasibo.com – Spasibo: Cras Nec Risus

SPASIBO
CRAS NEC RISUS

Register Log In



NYET! A FAIL IS YOU

Email has already been taken

Account Registration

Already have an account? Log in to your account here.

Email: user@host.com

Password:

Confirm Password:

I agree to the Terms & Conditions.

Create Account

Terms & Conditions

spasibo-test.html – Selenium IDE

Base URL http://spasibo.com/

Fast Slow

Table Source

Command	Target	Value
type	user_password	password
type	user_password_confi...	password
click	user_terms	
clickAndWait	user_submit	
waitForPageToLoad	5000	
verifyTextPresent	Welcome to Spasibo. ...	
verifyLocation	http://spasibo.com/	

Command verifyLocation

Target http://spasibo.com/ Find

Value

Log Reference UI-Element Rollup Info Clear

A winner is you! | |

[error] false

[info] Executing: |verifyLocation | http://spasibo.com/ | |

[error] Actual value 'http://spasibo.com/register' did not match 'http://spasibo.com/'

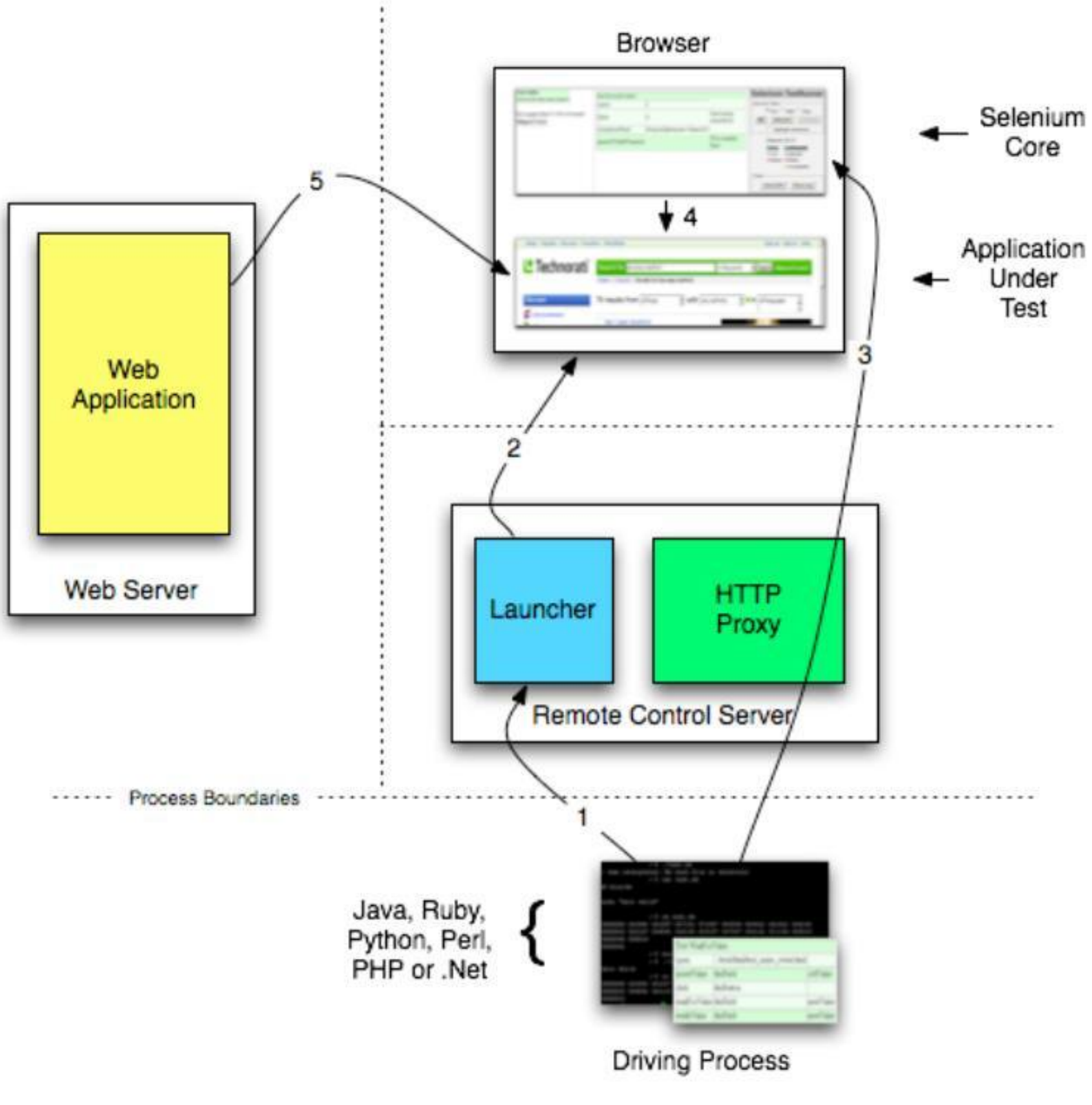
problems

- not maintainable
- not reusable
- not understandable



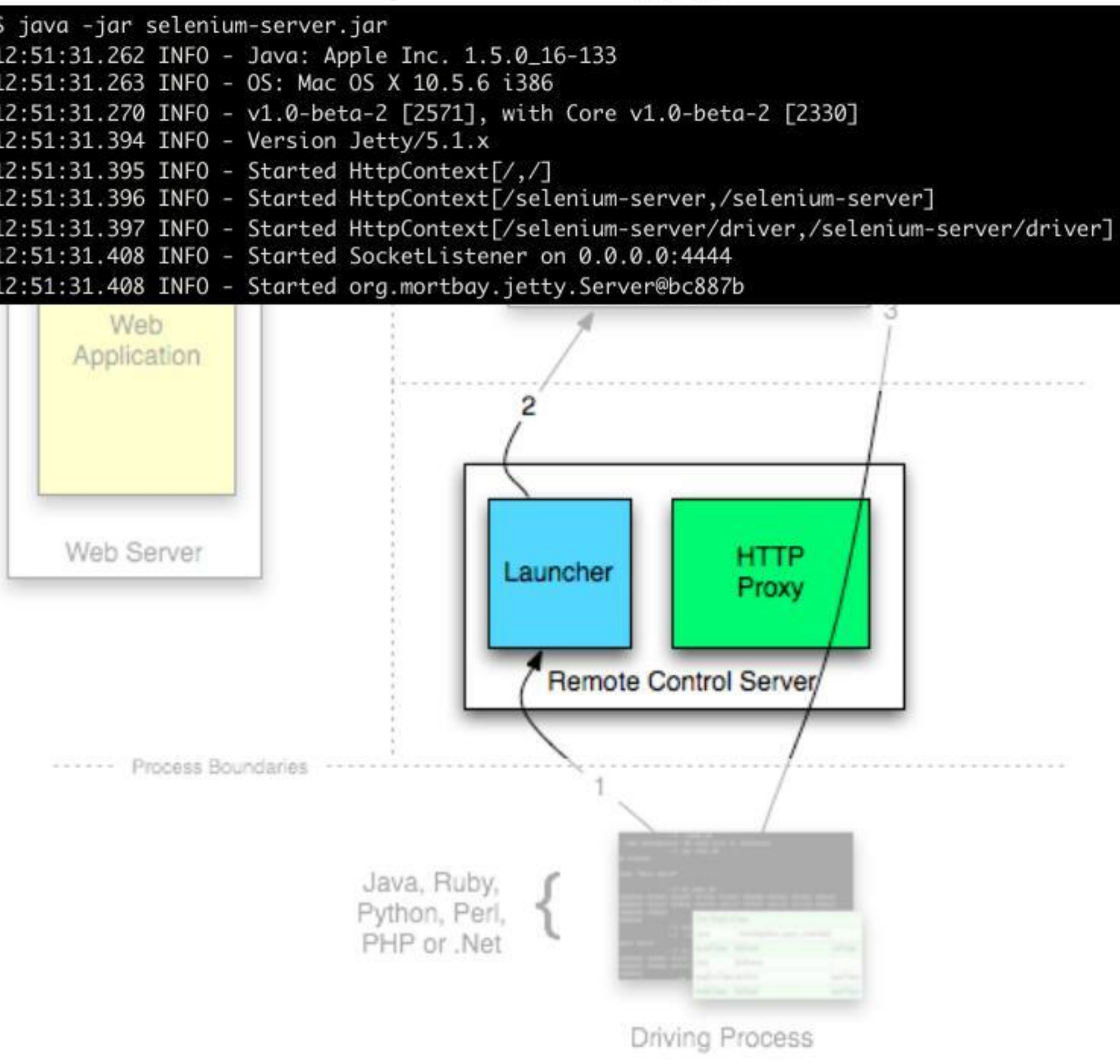
Selenium Remote Control

write automated web application tests in a familiar language and have them run in the browser


Browser
Selenium Core
Application Under Test
5
4
3
2
1
Web Application
Web Server
Launcher
HTTP Proxy
Remote Control Server
Process Boundaries
Java, Ruby, Python, Perl, PHP or .Net
Driving Process


Web Application
Web Server
5
Spasibo: Cras Nec Risus
SPASIBO
CRAS NEC RISUS
Register
Log In
Account Registration
Already have an account? Log in to your account here.
Email:
user@host.com
Password:
Confirm Password:
I agree to the Terms & Conditions.
Create Account
Terms & Conditions
Done
YSlow
0.115s
Process Boundaries
1
Java, Ruby, Python, Perl, PHP or .Net
Driving Process

```
$ java -jar selenium-server.jar
12:51:31.262 INFO - Java: Apple Inc. 1.5.0_16-133
12:51:31.263 INFO - OS: Mac OS X 10.5.6 i386
12:51:31.270 INFO - v1.0-beta-2 [2571], with Core v1.0-beta-2 [2330]
12:51:31.394 INFO - Version Jetty/5.1.x
12:51:31.395 INFO - Started HttpContext[/,/]
12:51:31.396 INFO - Started HttpContext[/selenium-server,/selenium-server]
12:51:31.397 INFO - Started HttpContext[/selenium-server/driver,/selenium-server/driver]
12:51:31.408 INFO - Started SocketListener on 0.0.0.0:4444
12:51:31.408 INFO - Started org.mortbay.jetty.Server@bc887b
```

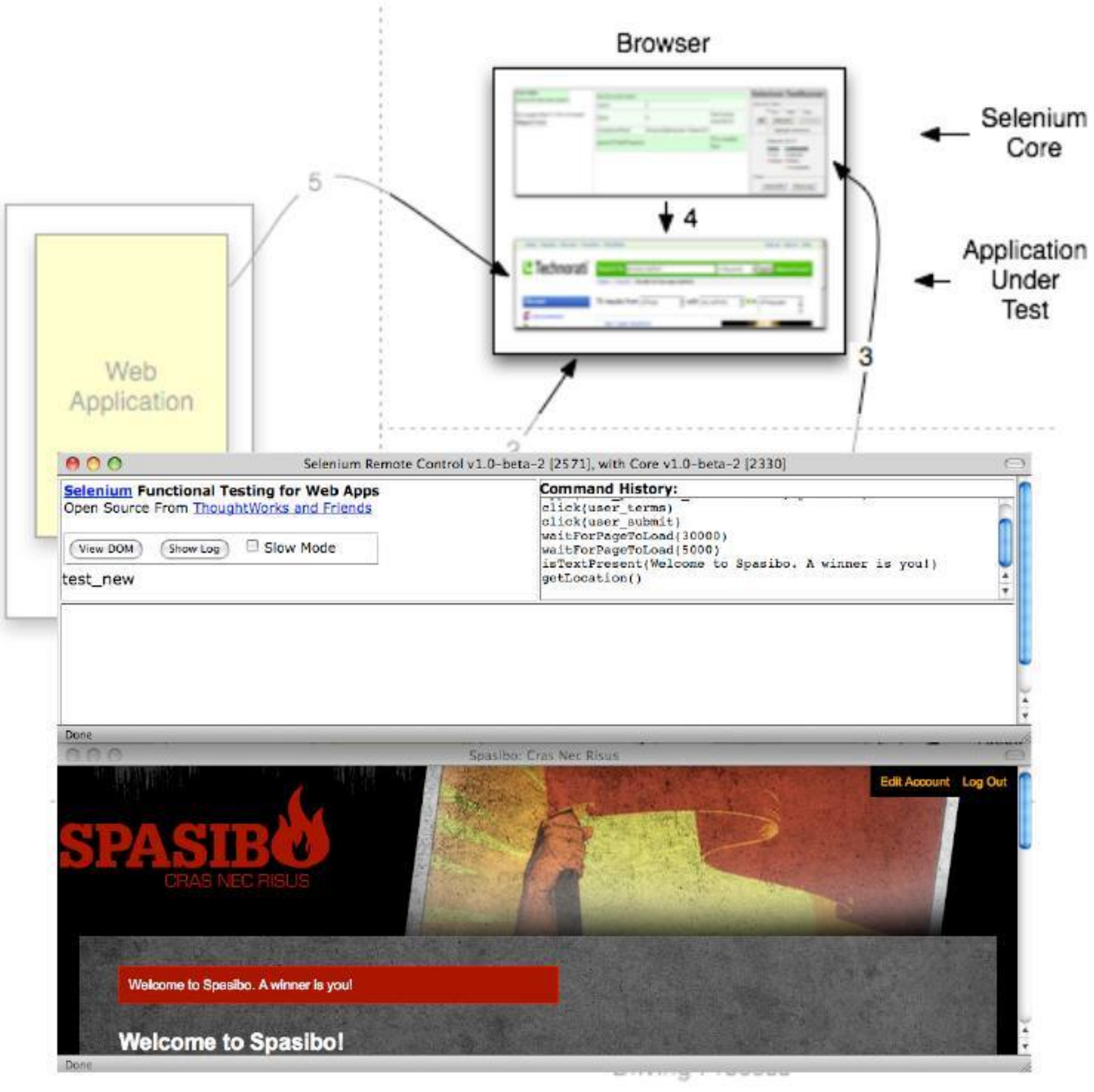


```ruby
class NewTest < Test::Unit::TestCase

  ...

  def test_new
    @selenium.open "/"
    @selenium.click "link=Register"
    @selenium.wait_for_page_to_load "30000"
    @selenium.type "user_email", "user@host.com"
    @selenium.type "user_password", "password"
    @selenium.type "user_password_confirmation", "password"
    @selenium.click "user_terms"
    @selenium.click "user_submit"
    @selenium.wait_for_page_to_load "30000"
    @selenium.wait_for_page_to_load "5000"
    begin
        assert @selenium.is_text_present("Welcome to Spasibo. A winner is you!")
    rescue Test::Unit::AssertionFailedError
        @verification_errors << $!
    end
    begin
        assert_equal "http://spasibo.com/", @selenium.get_location
    rescue Test::Unit::AssertionFailedError
        @verification_errors << $!
    end
  end
end
```




Browser
Selenium
Core
Application
Under
Test
5
4
3
2
Web
Application
Selenium Remote Control v1.0-beta-2 [2571], with Core v1.0-beta-2 [2330]
Selenium Functional Testing for Web Apps
Open Source From ThoughtWorks and Friends
View DOM
Show Log
Slow Mode
test_new
Command History:
click(user_terms)
click(user_submit)
waitForPageToLoad(30000)
waitForPageToLoad(5000)
isTextPresent(Welcome to Spasibo. A winner is you!)
getLocation()
Done
Spasibo: Cras Nec Risus
Edit Account
Log Out
SPASIBO
CRAS NEC RISUS
Welcome to Spasibo. A winner is you!
Welcome to Spasibo!
Done

```
$ gem install Selenium
```

export the test

```ruby
require "selenium"
require "test/unit"
require "open-uri"

class NewTest < Test::Unit::TestCase
  def setup
    base_url = "http://spasibo.com/"
    open("#{base_url}setup/clean_slate")

    @verification_errors = []
    if $selenium
      @selenium = $selenium
    else
      @selenium = Selenium::SeleniumDriver.new("localhost", 4444, "*chrome", base_url, 10000);
      @selenium.start
    end
    @selenium.set_context("test_new")
  end

  def teardown
    @selenium.stop unless $selenium
    assert_equal [], @verification_errors
  end

  def test_new
    @selenium.open "/"
    @selenium.click "link=Register"
    @selenium.wait_for_page_to_load "30000"
    @selenium.type "user_email", "user@host.com"
    @selenium.type "user_password", "password"
    @selenium.type "user_password_confirmation", "password"
    @selenium.click "user_terms"
    @selenium.click "user_submit"
    @selenium.wait_for_page_to_load "30000"
    @selenium.wait_for_page_to_load "5000"
    begin
        assert @selenium.is_text_present("Welcome to Spasibo. A winner is you!")
    rescue Test::Unit::AssertionFailedError
        @verification_errors << $!
    end
    begin
        assert_equal "http://spasibo.com/", @selenium.get_location
    rescue Test::Unit::AssertionFailedError
        @verification_errors << $!
    end
  end
end
```

```
def setup
  ...
  @selenium = Selenium::SeleniumDriver.new("localhost", 4444, "*chrome", base_url, 10000);
  @selenium.start
  ...
end
```

port

start URL

host

browser

timeout

setup

```
def setup
  base_url = "http://spasibo.com/"
  open("#{base_url}setup/clean_slate")

  ...
end
```

no users exist in the system

don’t do this in production

```
def setup
  base_url = "http://spasibo.com/"
  open("#{base_url}setup/clean_slate")

  ...
end
```

no users exist in the system

```
def test_new
  @selenium.open "/"
  @selenium.click "link=Register"
  @selenium.wait_for_page_to_load "30000"
  @selenium.type "user_email", "user@host.com"
  @selenium.type "user_password", "password"
  @selenium.type "user_password_confirmation", "password"
  @selenium.click "user_terms"
  @selenium.click "user_submit"
  @selenium.wait_for_page_to_load "30000"
  @selenium.wait_for_page_to_load "5000"

  ...

end
```

1. visit the registration page

```
def test_new
  @selenium.open "/"
  @selenium.click "link=Register"
  @selenium.wait_for_page_to_load "30000"
  @selenium.type "user_email", "user@host.com"
  @selenium.type "user_password", "password"
  @selenium.type "user_password_confirmation", "password"
  @selenium.click "user_terms"
  @selenium.click "user_submit"
  @selenium.wait_for_page_to_load "30000"
  @selenium.wait_for_page_to_load "5000"

  ...

end
```

2. enter valid information

```
def test_new
  @selenium.open "/"
  @selenium.click "link=Register"
  @selenium.wait_for_page_to_load "30000"
  @selenium.type "user_email", "user@host.com"
  @selenium.type "user_password", "password"
  @selenium.type "user_password_confirmation", "password"
  @selenium.click "user_terms"
  @selenium.click "user_submit"
  @selenium.wait_for_page_to_load "30000"
  @selenium.wait_for_page_to_load "5000"

  ...

end
```

3. submit the form

```
def test_new

  ...

  begin
      assert @selenium.is_text_present("Welcome to Spasibo. A winner is you!")
  rescue Test::Unit::AssertionFailedError
      @verification_errors << $!
  end
  begin
      assert_equal "http://spasibo.com/", @selenium.get_location
  rescue Test::Unit::AssertionFailedError
      @verification_errors << $!
  end
end
```

success message displayed

```
def test_new

  ...

  begin
      assert @selenium.is_text_present("Welcome to Spasibo. A winner is you!")
  rescue Test::Unit::AssertionFailedError
      @verification_errors << $!
  end
  begin
      assert_equal "http://spasibo.com/", @selenium.get_location
  rescue Test::Unit::AssertionFailedError
      @verification_errors << $!
  end
end
```

user is redirected

```
$ ruby -rubygems spasibo_test.rb
Loaded suite spasibo_test
Started
.
Finished in 10.559078 seconds.

1 tests, 3 assertions, 0
failures, 0 errors
```

problems

- dependent on a developer
- still not understandable

State:

* No users exist in the system

Steps:

1. Visit the registration page
2. Enter valid registration information
3. Submit the form

Expectation:

* I should be redirected to the home page
* I should see a success message displayed

can I make that executable?


Cucumber

Behavior-Driven Development

tool that speaks to domain experts first and code second

```
$ gem install cucumber
```

tweak the language

State:

* No users exist in the system

Steps:

1. Visit the registration page
2. Enter valid registration information
3. Submit the form

Expectation:

* I should be redirected to the home page
* I should see a success message displayed

State:

* Given no users exist in the system

Steps:

1. Visit the registration page
2. Enter valid registration information
3. Submit the form

Expectation:

* I should be redirected to the home page
* I should see a success message displayed

State:

* Given no users exist in the system

Steps:

1. When I visit the registration page
2. And I enter valid registration information
3. And I submit the form

Expectation:

* I should be redirected to the home page
* I should see a success message displayed

State:

* Given no users exist in the system

Steps:

1. When I visit the registration page
2. And I enter valid registration information
3. And I submit the form

Expectation:

* Then I should be redirected to the home page
* And I should see a success message displayed

State:

* Given no users exist in the system

Steps:

1. When I visit the registration page
2. And I enter valid registration information
3. And I submit the form

Expectation:

* Then I should be redirected to the home page
* And I should see a success message displayed

create a feature

```gherkin
# features/registration.feature

Feature: User Registration
  In order to identify users in the system
  Users must be able to log in

  Scenario: Successful Registration
    Given no users exist in the system
    When I visit the registration page
    And I enter valid registration information
    And I submit the form
    Then I should be redirected to the home page
    And I should see a success message displayed
```

```
$ cucumber registration.feature
```

```
You can implement step definitions for undefined steps with
these snippets:

Given /^no users exist in the system$/ do
  pending
end

When /^I visit the registration page$/ do
  pending
end

When /^I enter valid registration information$/ do
  pending
end

When /^I submit the form$/ do
  pending
end

Then /^I should be redirected to the home page$/ do
  pending
end

Then /^I should see a success message displayed$/ do
  pending
end
```

```
# steps/registration_steps.rb

Given /^no users exist in the system$/ do
  pending
end

When /^I visit the registration page$/ do
  pending
end

When /^I enter valid registration information$/ do
  pending
end

When /^I submit the form$/ do
  pending
end

Then /^I should be redirected to the home page$/ do
  pending
end

Then /^I should see a success message displayed$/ do
  pending
end
```

definitions

```ruby
# steps/registration_steps.rb

Given /^no users exist in the system$/ do
  open "#{BASE_URL}/setup/clean_slate"
end

When /^I visit the registration page$/ do
  pending
end

When /^I enter valid registration information$/ do
  pending
end

When /^I submit the form$/ do
  pending
end

Then /^I should be redirected to the home page$/ do
  pending
end

Then /^I should see a success message displayed$/ do
  pending
end
```

state

```ruby
# steps/registration_steps.rb

Given /^no users exist in the system$/ do
  open "#{BASE_URL}/setup/clean_slate"
end

When /^I visit the registration page$/ do
  @browser.open "/register"
end

When /^I enter valid registration information$/ do
  @browser.type  "user_email", "user@host.com"
  @browser.type  "user_password", "password"
  @browser.type  "user_password_confirmation", "password"
  @browser.click "user_terms"
end

When /^I submit the form$/ do
  @browser.click "user_submit"
  @browser.wait_for_page_to_load
end

Then /^I should be redirected to the home page$/ do
  pending
end

Then /^I should see a success message displayed$/ do
  pending
end
```

steps

```ruby
# steps/registration_steps.rb

Given /^no users exist in the system$/ do
  open "#{BASE_URL}/setup/clean_slate"
end

When /^I visit the registration page$/ do
  @browser.open "/register"
end

When /^I enter valid registration information$/ do
  @browser.type  "user_email", "user@host.com"
  @browser.type  "user_password", "password"
  @browser.type  "user_password_confirmation", "password"
  @browser.click "user_terms"
end

When /^I submit the form$/ do
  @browser.click "user_submit"
  @browser.wait_for_page_to_load
end

Then /^I should be redirected to the home page$/ do
  assert_equal "#{BASE_URL}/", @browser.get_location
end

Then /^I should see a success message displayed$/ do
  assert @browser.is_text_present("Welcome to Spasibo. A winner is you!")
end
```

expectation

```
$ cucumber registration.feature
Feature: User Registration
  In order to identify users in the system
  Users must be able to log in

  Scenario: Successful registration                  # registration.feature:5
    Given no users exist in the system               # steps/registration_steps.rb:1
    When I visit the registration page               # steps/registration_steps.rb:5
    And I enter valid registration information       # steps/registration_steps.rb:9
    And I submit the form                            # steps/registration_steps.rb:16
    Then I should be redirected to the home page     # steps/registration_steps.rb:21
    And I should see a success message displayed     # steps/registration_steps.rb:25

1 scenario (1 passed)
6 steps (6 passed)
```

problems

- dependent on a developer
- no control over input data

```
# registration.feature

Feature: User Registration
  In order to identify users in the system
  Users must be able to log in

  Scenario: Successful registration
    Given no users exist in the system
  When I visit "/register"
  And I enter "user@host.com" for "user_email"
  And I enter "password" for "user_password"
  And I enter "password" for "user_password_confirmation"
  And I check the "user_terms" box
  And I click the "user_submit" button
  Then I should be redirected to "/"
  And I should see the message "Welcome to Spasibo. A winner is you!"
```

```
You can implement step definitions for undefined steps with these
snippets:

Given /^no users exist in the system$/ do
  pending
end

When /^I visit "([^\"]*)"$/ do |arg1|
  pending
end

When /^I enter "([^\"]*)" for "([^\"]*)"$/ do |arg1, arg2|
  pending
end

When /^I check the "([^\"]*)" box$/ do |arg1|
  pending
end

When /^I click the "([^\"]*)" button$/ do |arg1|
  pending
end

Then /^I should be redirected to "([^\"]*)"$/ do |arg1|
  pending
end

Then /^I should see the message "([^\"]*)"$/ do |arg1|
  pending
end
```

```
# steps/registration_steps.rb

Given /^no users exist in the system$/ do
  pending
end

When /^I visit "([^\"]*)"$/ do |arg1|
  pending
end

When /^I enter "([^\"]*)" for "([^\"]*)"$/ do |arg1, arg2|
  pending
end

When /^I check the "([^\"]*)" box$/ do |arg1|
  pending
end

When /^I click the "([^\"]*)" button$/ do |arg1|
  pending
end

Then /^I should be redirected to "([^\"]*)"$/ do |arg1|
  pending
end

Then /^I should see the message "([^\"]*)"$/ do |arg1|
  pending
end
```

definitions

```ruby
# steps/registration_steps.rb

Given /^no users exist in the system$/ do
  open "#{BASE_URL}/setup/clean_slate"
end

When /^I visit "([^\"]*)"$/ do |arg1|
  pending
end

When /^I enter "([^\"]*)" for "([^\"]*)"$/ do |arg1, arg2|
  pending
end

When /^I check the "([^\"]*)" box$/ do |arg1|
  pending
end

When /^I click the "([^\"]*)" button$/ do |arg1|
  pending
end

Then /^I should be redirected to "([^\"]*)"$/ do |arg1|
  pending
end

Then /^I should see the message "([^\"]*)"$/ do |arg1|
  pending
end
```

state

```ruby
# steps/registration_steps.rb

Given /^no users exist in the system$/ do
  open "#{BASE_URL}/setup/clean_slate"
end

When /^I visit "([^\"]*)"$/ do |path|
  @browser.open path
end

When /^I enter "([^\"]*)" for "([^\"]*)"$/ do |value, text_field_id|
  @browser.type text_field_id, value
end

When /^I check the "([^\"]*)" box$/ do |checkbox_id|
  @browser.check checkbox_id
end

When /^I click the "([^\"]*)" button$/ do |submit_button_id|
  @browser.click submit_button_id
  @browser.wait_for_page_to_load
end

Then /^I should be redirected to "([^\"]*)"$/ do |arg1|
  pending
end

Then /^I should see the message "([^\"]*)"$/ do |arg1|
  pending
end
```

steps

```ruby
# steps/registration_steps.rb

Given /^no users exist in the system$/ do
  open "#{BASE_URL}/setup/clean_slate"
end

When /^I visit "([^\"]*)"$/ do |path|
  @browser.open path
end

When /^I enter "([^\"]*)" for "([^\"]*)"$/ do |value, text_field_id|
  @browser.type text_field_id, value
end

When /^I check the "([^\"]*)" box$/ do |checkbox_id|
  @browser.check checkbox_id
end

When /^I click the "([^\"]*)" button$/ do |submit_button_id|
  @browser.click submit_button_id
  @browser.wait_for_page_to_load
end

Then /^I should be redirected to "([^\"]*)"$/ do |path|
  assert_equal "#{BASE_URL}#{path}", @browser.get_location
end

Then /^I should see the message "([^\"]*)"$/ do |message|
  assert @browser.is_text_present(message)
end
```

expectation

```
$ cucumber registration.feature
Feature: User Registration
  In order to identify users in the system
  Users must be able to log in

  Scenario: Successful registration                                  # registration.feature:7
    Given no users exist in the system                               # steps/registration_steps.rb:1
    When I visit "/register"                                         # steps/registration_steps.rb:5
    And I enter "user@host.com" for "user_email"                     # steps/registration_steps.rb:9
    And I enter "password" for "user_password"                       # steps/registration_steps.rb:9
    And I enter "password" for "user_password_confirmation"          # steps/registration_steps.rb:9
    And I check the "user_terms" box                                 # steps/registration_steps.rb:13
    And I click the "user_submit" button                             # steps/registration_steps.rb:17
    Then I should be redirected to "/"                               # steps/registration_steps.rb:22
    And I should see the message "Welcome to Spasibo. A winner is you!" # steps/registration_steps.rb:26

1 scenario (1 passed)
9 steps (9 passed)
```

extend

```
Scenario: No data supplied
  Given no users exist in the system
  When I visit "/register"
  And I click the "user_submit" button
  Then I should be redirected to "/register"
  And I should see the message "NYET! A FAIL IS YOU"
  And I should see the message "Email can't be blank"
  And I should see the message "Terms must be accepted"

Scenario: Password not confirmed
Given no users exist in the system
When I visit "/register"
And I enter "user@host.com" for "user_email"
And I enter "password" for "user_password"
And I enter "drowssap" for "user_password_confirmation"
And I check the "user_terms" box
And I click the "user_submit" button
Then I should be redirected to "/register"
And I should see the message "NYET! A FAIL IS YOU"
And I should see the message "Password doesn't match confirmation"
```

```
Scenario: Successful registration with alternate data
  Given no users exist in the system
  When I visit "/register"
  And I enter "patrick@viget.com" for "user_email"
  And I enter "password123" for "user_password"
  And I enter "password123" for "user_password_confirmation"
  And I check the "user_terms" box
  And I click the "user_submit" button
  Then I should be redirected to "/"
  And I should see the message "Welcome to Spasibo. A winner is you!"
```



remove duplication

image: haakseth @ flickr

```
Scenario: Successful registration
  Given no users exist in the system
  When I visit "/register"
  And I enter "user@host.com" for "user_email"
  And I enter "password" for "user_password"
  And I enter "password" for "user_password_confirmation"
  And I check the "user_terms" box
  And I click the "user_submit" button
  Then I should be redirected to "/"
  And I should see the message "Welcome to Spasibo. A winner is you!"
```

```
Scenario Outline: Successful registration
  Given no users exist in the system
  When I visit "/register"
  And I enter "user@host.com" for "user_email"
  And I enter "password" for "user_password"
  And I enter "password" for "user_password_confirmation"
  And I check the "user_terms" box
  And I click the "user_submit" button
  Then I should be redirected to "/"
  And I should see the message "Welcome to Spasibo. A winner is you!"
```

```
Scenario Outline: Successful registration
  Given no users exist in the system
  When I visit "/register"
  And I enter "<email>" for "user_email"
  And I enter "<password>" for "user_password"
  And I enter "<password_confirmation>" for "user_password_confirmation"
  And I check the "user_terms" box
  And I click the "user_submit" button
  Then I should be redirected to "/"
  And I should see the message "Welcome to Spasibo. A winner is you!"
```

```
Scenario Outline: Successful registration
  Given no users exist in the system
  When I visit "/register"
  And I enter "<email>" for "user_email"
  And I enter "<password>" for "user_password"
  And I enter "<password_confirmation>" for "user_password_confirmation"
  And I check the "user_terms" box
And I click the "user_submit" button
Then I should be redirected to "/"
And I should see the message "Welcome to Spasibo. A winner is you!"

Examples:
  | email                | password    | password_confirmation |
  | user@host.com        | password    | password              |
  | patrick@viget.com | password123 | password123           |
```

```
$ cucumber registration.feature
Feature: User registration
  In order to identify users in the system
  Users must be able to log in

  Scenario Outline: Successful Registration                                  # registration.feature:7
    Given no users exist in the system                                       # steps/registration_steps.rb:1
    When I visit "/register"                                                 # steps/registration_steps.rb:5
    And I enter "<email>" for "user_email"                                   # steps/registration_steps.rb:9
    And I enter "<password>" for "user_password"                             # steps/registration_steps.rb:9
    And I enter "<password_confirmation>" for "user_password_confirmation"   # steps/registration_steps.rb:9
    And I check the "user_terms" box                                         # steps/registration_steps.rb:13
    And I click the "user_submit" button                                     # steps/registration_steps.rb:17
    Then I should be redirected to "/"                                       # steps/registration_steps.rb:22
    And I should see the message "Welcome to Spasibo. A winner is you!"      # steps/registration_steps.rb:26

    Examples:
      | email              | password    | password_confirmation |
      | user@host.com      | password    | password              |
      | patrick@viget.com  | password123 | password123           |

2 scenarios (2 passed)
18 steps (18 passed)
```


STOP
HAMMER TIME!
ALL WAY
image: mediageek @ flickr

future

```
Scenario: Duplicate email address
  Given a user exists in the system with the email "user@host.com"
  When I visit "/register"
  And I enter valid registration information
And I enter "user@host.com" for "user_email"
And I click the "user_submit" button
Then I should be redirected to "/register"
And I should see the message "Email has already been taken"
```

refactor

```
Scenario: Duplicate email address
  Given a user exists in the system with the email "user@host.com"
  When I visit "/register"
  And I enter valid registration information
And I enter "user@host.com" for "user_email"
And I click the "user_submit" button
Then I should be redirected to "/register"
And I should see the message "Email has already been taken"
```

refactor

```
Given /^a user exists in the system with the email "([^\"]*)"$/ do |email|
  Given 'no users exist in the system'
  When 'I visit "/register"'
  When 'I enter valid registration information'
  When 'I enter "' + email + '" for "user_email"'
  When 'I click the "user_submit" button'
end
```

```
When /^I enter valid registration information$/ do
  When 'I enter "user@host.com" for "user_email"'
  When 'I enter "password" for "user_password"'
  When 'I enter "password" for "user_password_confirmation"'
  When 'I check the "user_terms" box'
end
```

refactor

```
Scenario: Duplicate email address                                    # registration.feature:51
  Given a user exists in the system with the email "user@host.com"   # steps/registration_steps.rb:30
  When I visit "/register"                                           # steps/registration_steps.rb:5
  And I enter valid registration information                         # steps/registration_steps.rb:38
  And I enter "user@host.com" for "user_email"                       # steps/registration_steps.rb:9
  And I click the "user_submit" button                               # steps/registration_steps.rb:17
  Then I should be redirected to "/register"                         # steps/registration_steps.rb:22
  And I should see the message "Email has already been taken"        # steps/registration_steps.rb:26
```

```
$ cucumber registration.feature --name "Success"
Feature: User Registration
  In order to identify users in the system
  Users must be able to log in

  Scenario: Successful registration                                  # registration.feature:7
    Given no users exist in the system                               # steps/registration_steps.rb:1
    When I visit "/register"                                         # steps/registration_steps.rb:5
    And I enter "user@host.com" for "user_email"                     # steps/registration_steps.rb:9
    And I enter "password" for "user_password"                       # steps/registration_steps.rb:9
    And I enter "password" for "user_password_confirmation"          # steps/registration_steps.rb:9
    And I check the "user_terms" box                                 # steps/registration_steps.rb:13
    And I click the "user_submit" button                             # steps/registration_steps.rb:17
    Then I should be redirected to "/"                               # steps/registration_steps.rb:22
    And I should see the message "Welcome to Spasibo. A winner is you!" # steps/registration_steps.rb:26

  Scenario: Successful registration with alternate data              # registration.feature:18
    Given no users exist in the system                               # steps/registration_steps.rb:1
    When I visit "/register"                                         # steps/registration_steps.rb:5
    And I enter "patrick@viget.com" for "user_email"                 # steps/registration_steps.rb:9
    And I enter "password123" for "user_password"                    # steps/registration_steps.rb:9
    And I enter "password123" for "user_password_confirmation"       # steps/registration_steps.rb:9
    And I check the "user_terms" box                                 # steps/registration_steps.rb:13
    And I click the "user_submit" button                             # steps/registration_steps.rb:17
    Then I should be redirected to "/"                               # steps/registration_steps.rb:22
    And I should see the message "Welcome to Spasibo. A winner is you!" # steps/registration_steps.rb:26

2 scenarios (2 passed)
18 steps (18 passed)
```

run by name

```
@slow
Scenario: Duplicate email address
  Given a user exists in the system with the email "user@host.com"
  When I visit "/register"
And I enter valid registration information
And I enter "user@host.com" for "user_email"
And I click the "user_submit" button
Then I should be redirected to "/register"
And I should see the message "Email has already been taken"
```

tagging

```
$ cucumber registration.feature --tags @slow
Feature: User Registration
  In order to identify users in the system
  Users must be able to log in

  @slow
  Scenario: Duplicate email address                                    # registration.feature:51
    Given a user exists in the system with the email "user@host.com" # steps/registration_steps.rb:30
    When I visit "/register"                                         # steps/registration_steps.rb:5
    And I enter valid registration information                       # steps/registration_steps.rb:38
    And I enter "user@host.com" for "user_email"                     # steps/registration_steps.rb:9
    And I click the "user_submit" button                             # steps/registration_steps.rb:17
    Then I should be redirected to "/register"                       # steps/registration_steps.rb:22
    And I should see the message "Email has already been taken"      # steps/registration_steps.rb:26

1 scenario (1 passed)
7 steps (7 passed)
```

run by tag

```
$ cucumber registration.feature --tags ~@slow
```

Windows XP Professional

Selenium Remote Control v... | Spasibo: Cras Nec Risus

http://spasibo.com/register

Edit Account Log Out

SPASIBO
CRAS NEC RISUS

```
ta-2>java -jar selenium-server.jar
01:39:19.835 INFO - Java: Sun Microsystems Inc. 11.3-b02
01:39:19.835 INFO - OS: Windows XP 5.1 x86
01:39:19.835 INFO - v1.0-beta-2 [2571], with Core v1.0-beta-2 [2330]
01:39:19.929 INFO - Version Jetty/5.1.x
01:39:19.945 INFO - Started HttpContext[/selenium-server/driver,/selenium-server/driver]
01:39:19.945 INFO - Started HttpContext[/selenium-server,/selenium-server]
01:39:19.945 INFO - Started HttpContext[/,/]
01:39:19.960 INFO - Started SocketListener on 0.0.0.0:4444
01:39:19.960 INFO - Started org.mortbay.jetty.Server@109a4c
```

NYET! A FAIL IS YOU

Email has already been taken

Account Registration

Already have an account? **Log in to your account here.**

Email: user@host.com

Password:

Confirm Password:

I agree to the **Terms & Conditions**.

Create Account

cross platform

Start | C:\WINDOWS\system32... | Spasibo: Cras Nec Ris... | 1:35 AM

To return to your computer, press Control-⌘

continuous integration



cruisecontrol

continuous integration

integration?



cruisecontrol

Resources

- Selenium: http://seleniumhq.org/
- Cucumber: http://cukes.info/
- Slides / Code: http://github.com/reagent/nlt

Contact

- patrick.reagan@viget.com
- http://twitter.com/reagent



image: nomeacuerdo @ flickr