Tech Talk
Selenium - on Rails

by Tino Gomes

http://tinogomes.wordpress.com (In portuguese)

What is Selenium?

http://selenium.openqa.org/

- is a suite of tools specifically for testing web applications.

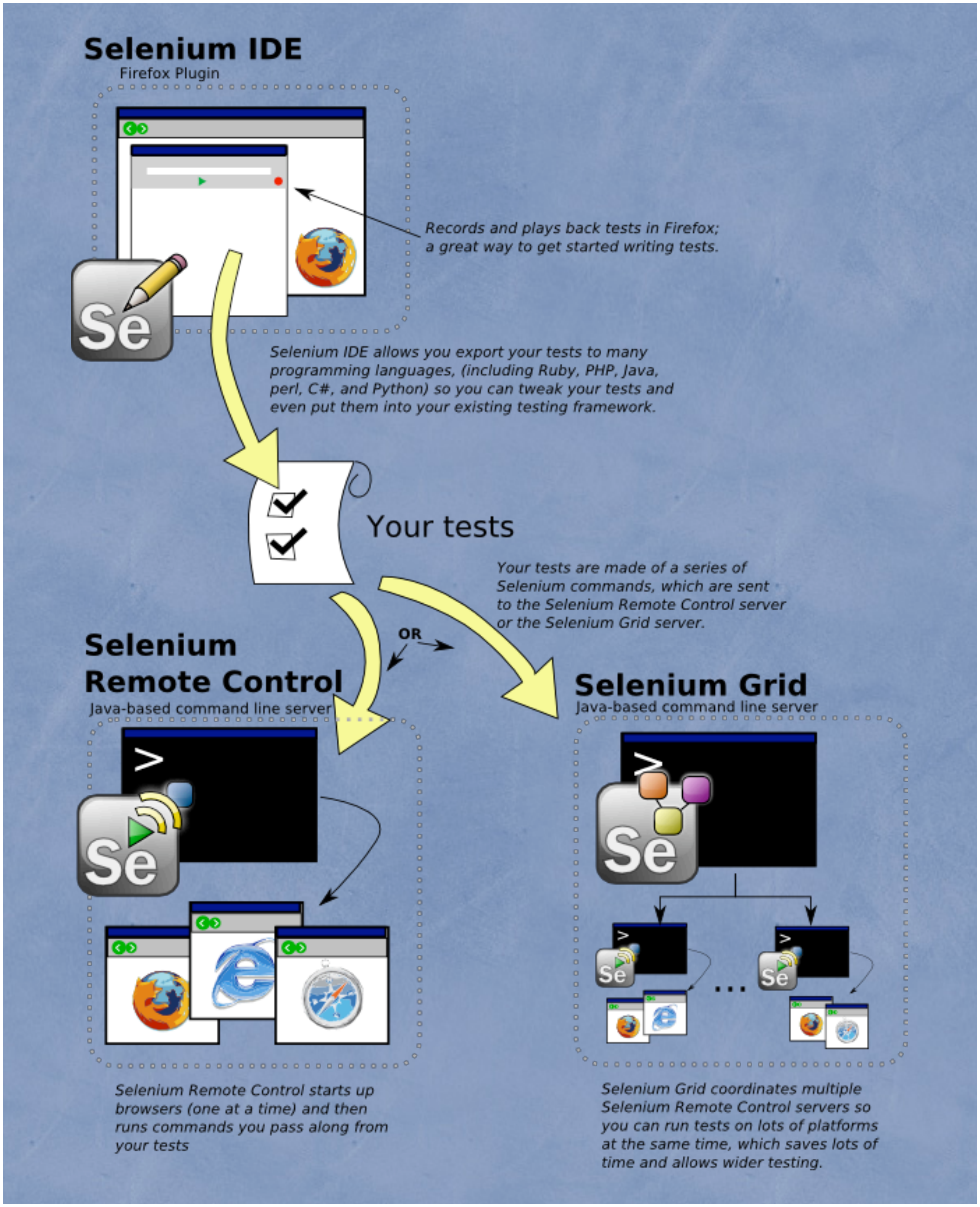
Tools for Selenium

http://selenium.openqa.org/projects/

- Selenium Remote Control - is a client/server system that allows you to control web browsers locally or on other computers, using almost an programming language and testing framework.
- Selenium IDE - is a Firefox add-ons that makes it easy to record and playback tests in Firefox 2+.
- Selenium Grid - takes Selenium Remote Control to another level by running tests on many servers at the same time, cutting down on the time it takes to test multiple browsers or operating systems.
- Selenium Core - is the original Javascript-based testing system. It's now used primarily as a component of Selenium Remote Control, but it can also be used as a pure Javascript/HTML testing system. (Let's see a little...)
- Selenium on Rails - provides an easy way to test Rails application with Selenium Core. Suite files are created on the fly for each directory and the Selenium Core files don't have to pollute /public.

How Selenium works?

http://selenium.openqa.org/about/how.html



Records and plays back tests in Firefox; a great way to get started writing tests.

Selenium IDE allows you export your tests to many programming languages, (including Ruby, PHP, Java, perl, C#, and Python) so you can tweak your tests and even put them into your existing testing framework.

Your tests are made of a series of Selenium commands, which are sent to the Selenium Remote Control server or the Selenium Grid server.

Selenium Remote Control starts up browsers (one at a time) and then runs commands you pass along from your tests

Selenium Grid coordinates multiple Selenium Remote Control servers so you can run tests on lots of platforms at the same time, which saves lots of time and allows wider testing.

Railscasts

with Ryan Bates

Railscasts - by Ryan Bates

Episode 116

Formats

http://selenium-on-rails.openqa.org/

- Selenese, .sel (If the RedCloth gem is available the Selenese test cases can use it for better markup)
  - http://selenium-core.openqa.org/reference.html
- RSelenese, .rsel
  - http://release.openqa.org/selenium-remote-control/0.9.2/doc/ruby/
- HTML/RHTML
- Partials

Let's play now!

- rails blog -d mysql
- cd blog
- mate .
- rake db:create:all
- ruby script/generate scaffold post title:string body:text published:boolean
- rake db:migrate
- rake db:test:clone
- ruby script/plugin install git://github.com/ryanb/selenium-on-rails.git
- ruby script/server -e test
- http://localhost:3000/selenium

Selenese Mode
ruby script/generate selenium posts/new.sel

```
|open|/selenium/setup?fixtures=all|

|open|/posts/new|

|assertElementPresent|//h1[text()='New post']||

|typeKeys|post_title|New post for example|

|type|post_body|The book is on the table!|

|check|post_published||

|clickAndWait|post_submit||

|assertElementPresent|//p[@style='color: green;' and text() = 'Post was successfully created.']||
```

RSelenese Mode - A Partial
ruby script/generate selenium posts/_index_sctructure.rsel

```
open '/posts'
assert_element_present '//h1[text()="Listing posts"]'
assert_element_present 'link=New post'
```

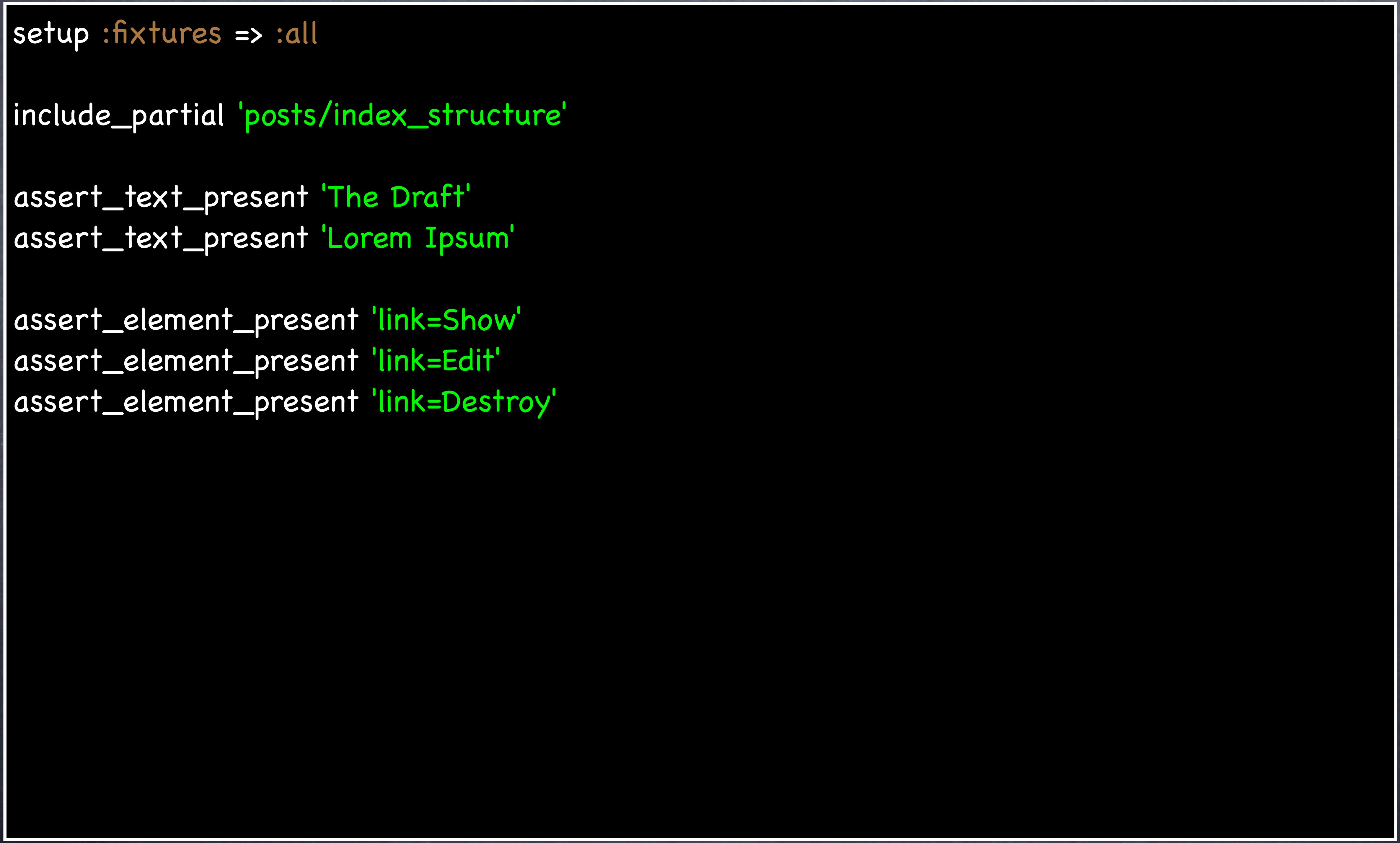
RSelenese Mode

ruby script/generate selenium posts/index_empty.rsel

```
#delete from posts
setup :clear_tables => :posts

#use partial
include_partial 'posts/index_structure'

#will fail, change for verify_text_present to see the difference between this and assert...
assert_text_present 'No posts'

assert_element_not_present 'link=Show'
assert_element_not_present 'link=Edit'
assert_element_not_present 'link=Destroy'
```

RSelenese Mode

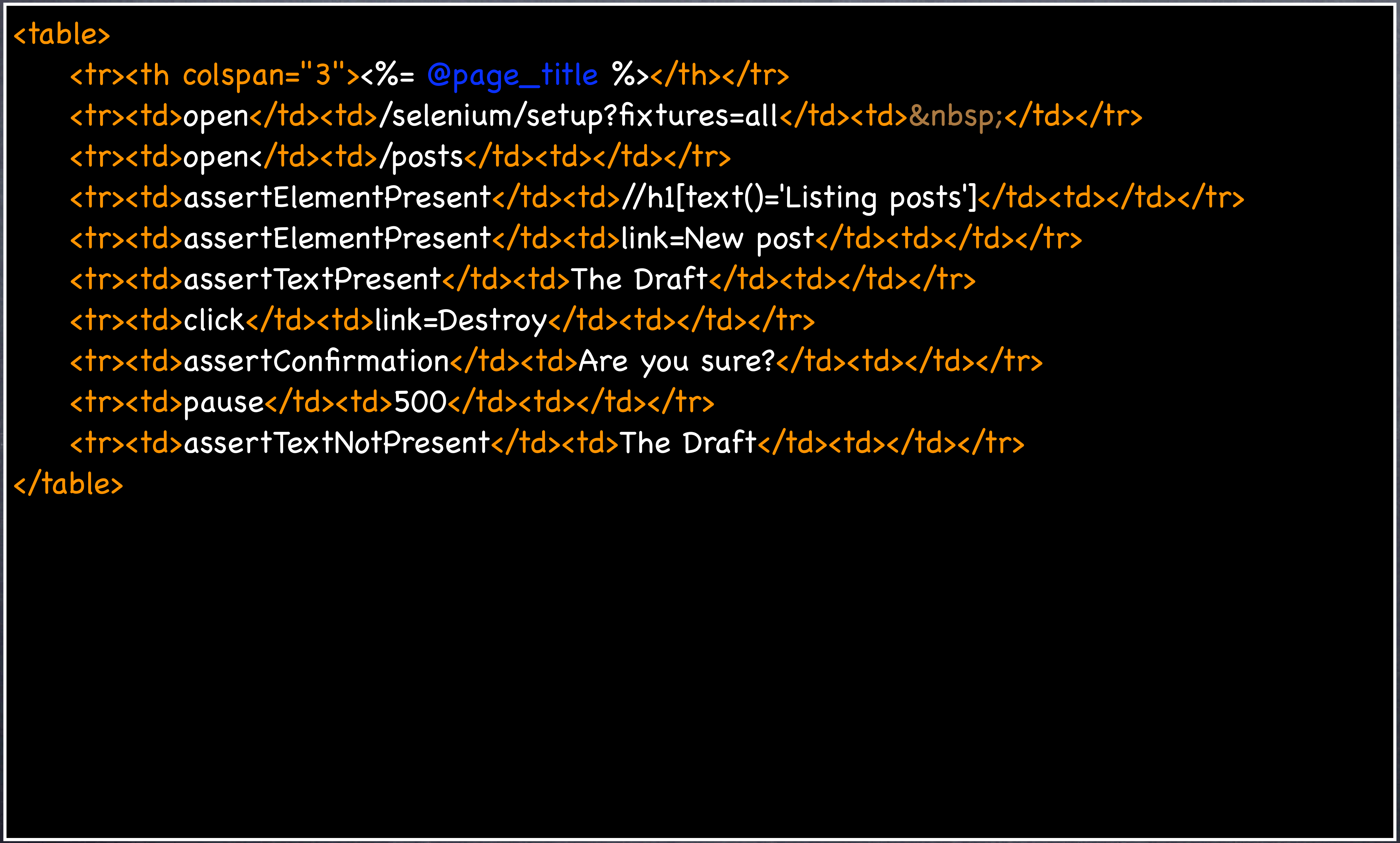
ruby script/generate selenium posts/index_full.rsel



```
setup :fixtures => :all

include_partial 'posts/index_structure'

assert_text_present 'The Draft'
assert_text_present 'Lorem Ipsum'

assert_element_present 'link=Show'
assert_element_present 'link=Edit'
assert_element_present 'link=Destroy'
```

RHTML Mode

ruby script/generate selenium posts/destroy.rhtml



```
<table>
    <tr><th colspan="3"><%= @page_title %></th></tr>
    <tr><td>open</td><td>/selenium/setup?fixtures=all</td><td>&nbsp;</td></tr>
    <tr><td>open</td><td>/posts</td><td></td></tr>
    <tr><td>assertElementPresent</td><td>//h1[text()='Listing posts']</td><td></td></tr>
    <tr><td>assertElementPresent</td><td>link=New post</td><td></td></tr>
    <tr><td>assertTextPresent</td><td>The Draft</td><td></td></tr>
    <tr><td>click</td><td>link=Destroy</td><td></td></tr>
    <tr><td>assertConfirmation</td><td>Are you sure?</td><td></td></tr>
    <tr><td>pause</td><td>500</td><td></td></tr>
    <tr><td>assertTextNotPresent</td><td>The Draft</td><td></td></tr>
</table>
```

Review

- http://github.com/ryanb/selenium-on-rails/tree - Plugin Selenium on Rails for Rails 2.1
- http://selenium.openqa.org/ - All about Selenium
- http://railscasts.com/episodes/116-selenium - Ryan Bates Screencast
- http://selenium-core.openqa.org/reference.html - Reference for Selenium Core
- http://release.openqa.org/selenium-remote-control/0.9.2/doc/ruby/ - RDoc for Ruby Selenium Remote Control

Thank you! :D