

Using Selenium for configuration management in Drupal

Young-Jin Kim & Allen Shaw

youngjin@emphanos.com, allen@emphanos.com

Emphanos LLC, Chicago, IL



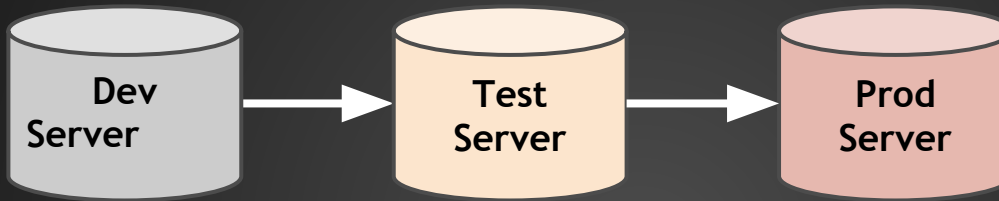
emphanos

open innovation



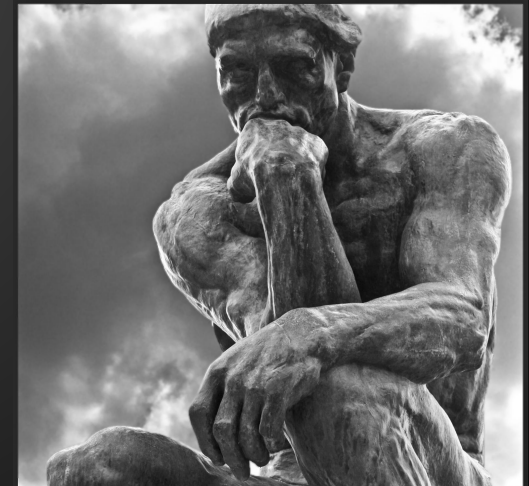
1 2 3 4 5 6 7 8 9 10

Avoid Drupal configuration hell



Hard because Drupal is heavily UI driven. What makes it flexible makes it hard to preserve configurations, only a few options:

- Features Module... (only works partially)
- Meticulous documentation? (not happenin')
- Selenium Record/Playback (!)





1 2 3 4 5 6 7 8 9 10

What is Selenium?

Testing framework to record and playback website user interactions originally developed by ThoughtWorks for acceptance testing, released as open source. De facto standard in functional and acceptance testing of web applications. <http://seleniumhq.org>



Why learn yet another tool?

1. Ultimately will save you a lot of time and headaches
2. Distribute configuration work across team members
3. Inspect and improve scripts for accuracy and re-use (!)
4. Take script from staging to production without the dread
5. Teach someone how to do a configuration via Selenium



1 2 3 4 5 6 7 8 9 10

Record with Selenium IDE [Demo]

Selenium IDE (Integrated Development Environment) allows you to record website interactions via a Firefox plugin. Once installed you'll have the **ability to move-cut-copy-paste lines** within the Selenium script and control the IDE using the following shortcut keys:

- X → execute command on the line
- B → set a break point to stop execution
- S → set/clear new internal start point

NOTE: **Disable** in IDE wide settings Options → Options → General → "Record absolute URL", super important for dev-test-prod scripting



1 2 3 4 5 6 7 8 9 10

Common gotchas of Selenium IDE

DOM structure funkiness, Network lag, AJAX calls, database delays can timeout Selenium or trip it up. It's always a good idea to adjust the raw recorded Selenese with these Selenium workarounds:

Prefer	When	Avoid	Situation
open	instead of	click, clickAndWait	if URL is known, this is faster
clickAndWait	instead of	click	form submit button or page refresh
waitForTextPresent	insert after	click	AJAX calls
runScript + JQuery	instead of	click, select	if you want ability to re-run script again
assertValue	instead of	<u><i>{blind faith}</i></u>	if you want to ensure toggle state

NOTE: Use right-click on an HTML element to view available Selenium commands, use command completion and Reference tab



1 2 3 4 5 6 7 8 9 10

Mastering Selenium Test Suites

Test Suites are collections of individual Test Cases strung together. Each Test Suite can contain multiple occurrences of a single Test Case (e.g. login_step.html), but you can't nest a Test Suite inside another Test Suite, just as you can't nest a Test Case inside another Test Case.

KEY POINT: If a specific set of Selenium steps appears as a snippet multiple times, create a separate new test case out of that snippet and break it up the Test Suite into reusable components

KEY POINT: Use a consistent naming convention for the Test Cases and Test Suites, e.g. use prefixes "SUITE_" or "TEST_" with ordinal numbers to indicate relative ordering, e.g. "TEST_01_Login", "TEST_02_Set_Taxnomy", "TEST_03_Add_User", etc.



1 2 3 4 5 6 7 8 9 10

Store variables early, use often

Use variables to store usernames, passwords and configuration specific data that's used throughout your site.

Initialize variables at the beginning of a Test Suite so you only have to change them in one place!

Store variable early	Reuse variable later
<pre><tr> <td>store</td> <td>EmphanosTester</td> <td>drupaluser</td> </tr></pre>	<pre><tr> <td>type</td> <td>id=edit-name</td> <td>\${drupaluser}</td> </tr></pre>



1 2 3 4 5 6 7 8 9 10

What about Conditionals?

Selenium IDE records interactions in HTML format (Selenese). Since HTML doesn't have loops or if-then conditionals (that's why JavaScript was cobbled together early on ;-), neither does Selenese.

Some Selenium IDE extensions purport to do conditionals like "sideflow.js" which can be loaded as Selenium Core extensions, but they use ugly GOTO code to duct-tape on conditionals and don't work with the runScript command. **RECOMMENDATION: AVOID**

For true conditionals you'll have to translate Selenese to a Selenium Remote Control or Selenium Webdriver script in PHP or Python (**recommended**).



1 2 3 4 5 6 7 8 9 10

Some Tricks using JavaScript

runScript can use arbitrary JavaScript snippet inside a step, incl. JQuery

BRITTLE and UNSAFE

assertValue	id=edit-active-tags	off
click	id=edit-active-tags	
assertValue	id=edit-active-tags	on

ROBUST and SAFE

runScript	<code>\$('#input[id=edit-active-tags]').attr('checked', true)</code>	
assertValue	id=edit-active-tags	on



1 2 3 4 5 6 7 8 9 10

Best Practices using Selenium

1. Use **version control** on your Selenium directory, easily spot diffs
2. In teams, use Dropbox to sync your Selenium script folder
3. **"Fix" scripts as you record them**, i.e. record, replay, edit, repeat
4. Create a reset-snapshot of your DB when fiddling with scripts
5. Just as in coding, **if you see it repeated, refactor your scripts**
6. Use assert statements especially when submitting data
7. **Fully use IDE with keyboard shortcuts** to debug snippets
8. Use variables to make scripts reusable (change in one script!)
9. If you move to Selenium Server using a language specific Webdriver binding, use a headless X-windows setup (Xvfb)
10. Continuous integration with Jenkins works great and will avoid port 4444 locking issues and you can queue your config jobs!



1 2 3 4 5 6 7 8 9 10

Best Practices using Selenium

11. Write **one suite per test case**, rather than stringing multiple unrelated test cases together into a large suite. This allows specific testing of each case when needed.
12. Use **one test case to set variables for account credentials**, and always use only variables for these values in tests.
13. Document test cases in English (e.g., in a Google Doc), **numbered sequentially**. Then use filenames like "SUITE_01_description" and "CASE_01_a_description". This makes files easy to find.
14. Insert comments in tests showing **which step is being executed** (e.g., "1.a.iv"), to ease communication: Testers can say "the test is failing at step 1.a.iv" instead of "about half-way through".



1 2 3 4 5 6 7 8 9 10

Best Practices using Selenium

15. Have **multiple versions of a suite**? Only one should be in use, and other(s) should be moved to an "archive" directory. All testers are using the same suite for any given issue, & testers never have to ask "Which version is failing?"
16. **Manage local paths** for example files used to test file upload features, which will differ for testers on different platforms. Testers can use the script "copy_selenium.sh" (https://github.com/twomice/misc/tree/master/copy_selenium) to automate the process of copying the Selenium test files to a separate location and replacing the file path with one that works in the local environment.



1 2 3 4 5 6 7 8 9 10

Best Practices using Selenium

17. Use the "pause" command only for testing based on the passing of time (which is rare). **Instead use a "waitFor..." command** such as "waitForElementPresent", "waitForTextPresent", etc. To make this work well, use a pattern like this:
- - `assertElementNotPresent | id=foo`
 - - `click | link="get foo"`
 - - `waitForElementPresent | id=foo`
 - - `[do something with foo]`
- **Pro:** The "pause" time will eventually be too short or too long, based on variations in the testing environment; if too short, the test will break; if it's too long, the test will be slower than necessary.



1 2 3 4 5 6 7 8 9 10

Thank you, questions welcome.

<http://seleniumhq.org>

http://seleniumhq.org/docs/02_selenium_ide.html

<http://saucelabs.com> *check out scout!*

<http://en.wikipedia.org/wiki/Xvfb>

<http://jenkins-ci.org>

<https://wiki.jenkins-ci.org/display/JENKINS/Xvfb+Plugin>

Young-Jin Kim

youngjin@emphanos.com

Emphanos LLC, Chicago, IL



emphanos

open innovation

Useful Selenium extensions

<https://addons.mozilla.org/en-US/firefox/addon/power-debugger-selenium-ide/>

Super useful extension that pauses the execution of Test SUITE if a subtest fails! A Must-have!

<https://addons.mozilla.org/en-US/firefox/addon/stored-variables-viewer-seleni/>

If you're using a lot of stored variables and TESTs that act as functions TESTFUNC with a preceding PREP_step, then this is really helpful

<https://addons.mozilla.org/en-US/firefox/addon/selenium-expert-selenium-ide/>

Inspects your Selenium tests for very common store Target Value mix-ups (variable assignment is backwards) and tries to catch other common gotchas

<https://addons.mozilla.org/en-US/firefox/addon/test-results-selenium-ide/>

Exports the test results of a SUITE or TEST into a colorful HTML file similar to the output seen on Selenium IDEs screen

<https://addons.mozilla.org/en-US/firefox/addon/file-logging-selenium-ide/>

<https://addons.mozilla.org/en-US/firefox/addon/page-coverage-selenium-ide/>

<https://addons.mozilla.org/en-US/firefox/addon/log-search-bar-selenium-ide/>

<https://addons.mozilla.org/en-US/firefox/addon/highlight-elements-selenium-id/>