



THE LEADER IN DRUPAL PLATFORM DESIGN AND DEVELOPMENT

TESTING DRUPAL WITH GHOSTS AND GHERKINS

USING CASPERJS AND BEHAT TO TEST DRUPAL SITES



STEVEN MERRILL

Director of Engineering

smerrill@phase2technology.com

[@stevenmerrill](#)

<https://github.com/smerrill>

ABOUT ME

- Fan of Jenkins and continuous integration
- Running Jenkins in production for about 4 years

ABOUT THIS PRESENTATION

- Originally titled "Testing Your Site With Friendly Ghosts"
- Developed by Eric Duran and me
- Thanks, Eric!

A TWITTER EXCHANGE

PEOPLE WANT TO TEST THEIR DRUPAL MODULES WITH REAL BROWSERS



**Is there precedent for including selenium tests in a
#drupal contrib module? #lazyweb**

@stevector





**@stevector <https://github.com/ericduran/>
views load more is a neat example of including
@casperjs_org tests and running them using @travis-ci.**

@stevenmerrill





@stevenmerrill Thanks! That approach looks very viable for the what I want to do (check for text in a CTools modal).

@stevector



YOU NEED TO TEST YOUR SITE

SIMPLETEST

- Let's test complex AJAX in a real browser!



SIMPLETEST

phase://
TECHNOLOGY

SELENIUM

- Let's add variables into a test that was recorded by clicking around the interface.



SELENIUM

phase://
TECHNOLOGY

FRIENDLY GHOSTS

- Website testing tools based around headless WebKit.
- PhantomJS (<https://github.com/ariya/phantomjs>)
- CasperJS (<https://github.com/n1k0/casperjs>)

FRIENDLY GHOSTS

- Headless website testing
- Site scraping
- Run QUnit

PHANTOMJS

- Headless Webkit browser
- Scriptable via a JavaScript API
 - Write data to the filesystem
 - Render <canvas> contents
- Use JavaScript or CoffeeScript

WHY NOT PHANTOMJS?

```
var page = require('webpage').create();

page.open(url1, function(status) {
  if (status == "fail") phantom.exit();
  page.open(url2, function(status) {
    if (status == "fail") phantom.exit();
    page.open(url3, function(status) {
      if (status == "fail") phantom.exit();
      page.open(url4, function(status) {
        if (status == "fail") phantom.exit();
        // Can I stop, now?
      });
    });
  });
});
```

```
var casper = require('casper').create();
```

```
casper.start(url1);
```

```
casper.thenOpen(url2);
```

```
casper.thenOpen(url3);
```

```
casper.thenOpen(url4);
```

```
casper.run();
```

THE CASPERJS API

- Take actions through a website
- Clicking/following links
- Filling/submitting forms
- Downloading resources
- Capturing screenshots
- Running DOM assertions

```
// Search for 'casperjs' from google.com.
casper.start('http://google.com/', function() {
  this.fill('form[action="/search"]',
    { q: 'casperjs' }, true);});

// Search for a 'casperjs' module from d.o.
casper.start('http://drupal.org/', function() {
  this.fill('#search-theme-form',
    {search_theme_form: 'casperjs',
     meta_type: 'module'}, true);});
```

TESTING

- You use JavaScript to build sites!
- Test real JavaScript interactions.
- Save screenshots or scraped data.
- Export JUnit XML and hook into Jenkins.

TRAVIS + CASPER AUTOMATED TESTING

- Host a mirror of your module on GitHub
 - https://github.com/ericduran/views_load_more
- Use TravisCI to automate your testing
 - https://github.com/ericduran/views_load_more/blob/7.x-1.x/.travis.yml
- Write tests and update your README with the build status!


```
language: php
```

```
php:
```

```
- 5.4
```

```
mysql:
```

```
  database: drupal
```

```
  username: root
```

```
  encoding: utf8
```

```
before_script:
```

- mysql -e 'create database drupal;'
- pyrus channel-discover pear.drush.org
- pyrus install drush/drush
- phpenv rehash

```
- wget http://ftp.drupal.org/files/projects/drupal-7.14.tar.gz
- tar -xf drupal-7.14.tar.gz
- cd drupal-7.14
- drush site-install standard --db-url=mysql://root:@localhost/drupal
--yes
- git clone --quiet --branch casper-test git://github.com/ericduran/
views_load_more.git ./sites/all/modules/views_load_more
- drush en views_load_more_test --yes
- drush cc all --yes
```

- "export PHANTOMJS_EXECUTABLE='phantomjs --local-to-remote-url-access=yes --ignore-ssl-errors=yes'"
- "export DISPLAY=:99.0"
- "sh -e /etc/init.d/xvfb start"
- sleep 3 # give xvfb some time to start
- drush runserver --server=builtin 8080 &
- sleep 3 # give Web server some time to bind to sockets, etc
- cd ..
- git clone git://github.com/n1k0/casperjs.git
- cd casperjs
- git checkout tags/0.6.10
- cd ./bin

script:

- "DISPLAY=:99.0 ./casperjs test ../../test/casperjs/"

DEMOS

INSTALLING PHANTOM AND CASPER

```
brew install phantomjs  
brew install casperjs
```

PHANTOM/CASPER LINKS

- <http://phantomjs.org/>
- <http://casperjs.org/index.html>
- <https://github.com/ericduran/friendly-ghosts-examples>

API DOCUMENTATION

- Client Utilities: <http://casperjs.org/api.html#client-utils>
- Testing API: <http://casperjs.org/api.html#tester>
- Utils: <http://casperjs.org/api.html#utils>
- CLI: <http://casperjs.org/cli.html>

BEHAVIORAL TESTING

BEHAT

- Behavior driven development framework for PHP 5.3+
- Uses Symfony components
- Uses the Gherkin format for feature definitions

MINK

- Browser testing for Behat
- Has a pure PHP test browser (Goutte)
- Can run Selenium or Selenium 2 tests
- Can farm testing out to Sauce Labs

GHERKIN

- Business-readable domain specific language
- Features
- Scenarios
- Step Definitions

GHERKIN STEP DEFINITIONS

- Given
 - Preconditions
- When
 - Actions
- Then
 - Assertions

Feature: In order to learn about automated testing tools
As a DrupalCamp New Jersey attendee
I want to attend Testing Drupal with Ghosts and Gherkins

Scenario: Attending the session

Given that the time is 10:30 AM

When I enter room Friend 008

Then the presenter should be Steven Merrill

And the session should be Testing Drupal

MINK STEP DEFINITIONS

- Navigate to pages
- Check HTTP status codes
- Fill in forms by label
- Click links
- Look for text

```
$ ./bin/behat -dl
```

```
Given /^(?:|I )am on homepage$/  
  When /^(?:|I )go to homepage$/  
Given /^(?:|I )am on "(?P<page>[^\"]+)"$/  
  When /^(?:|I )go to "(?P<page>[^\"]+)"$/  
  When /^(?:|I )reload the page$/  
  When /^(?:|I )move backward one page$/  
  When /^(?:|I )move forward one page$/  
  When /^(?:|I )press "(?P<button>(?:[^\"]|\\")*)"$/  
  When /^(?:|I )follow "(?P<link>(?:[^\"]|\\")*)"$/  
  When /^(?:|I )fill in "(?P<field>(?:[^\"]|\\")*)" "  
    with "(?P<value>(?:[^\"]|\\")*)"$/  
  When /^(?:|I )fill in "(?P<value>(?:[^\"]|\\")*)" "  
    for "(?P<field>(?:[^\"]|\\")*)"$/  
  When /^(?:|I )fill in the following:$/
```

```
When /^(?:|I )select "(?P<option>(?:[^\"]|\\")*)"
  from "(?P<select>(?:[^\"]|\\")*)"$/
When /^(?:|I )additionally select "(?P<option>(?:[^\"]|\\")*)"
  from "(?P<select>(?:[^\"]|\\")*)"$/
When /^(?:|I )attach the file "(?P[^\"]*)" to "(?P<field>(?:[^\"]|\\")*)"$/
Then /^(?:|I )should be on "(?P<page>[^\"]+)"$/
Then /^the (?i)url(?-i) should match (?P<pattern>"([^\"]|\\")*" )$/
Then /^the response status code should be (?P<code>\d+)$/
Then /^the response status code should not be (?P<code>\d+)$/
Then /^(?:|I )should see "(?P<text>(?:[^\"]|\\")*)"$/
Then /^(?:|I )should not see "(?P<text>(?:[^\"]|\\")*)"$/
Then /^(?:|I )should see text matching (?P<pattern>"(?:[^\"]|\\")*" )$/
Then /^(?:|I )should not see text matching (?P<pattern>"(?:[^\"]|\\")*" )$/
Then /^the response should contain "(?P<text>(?:[^\"]|\\")*)"$/
Then /^the response should not contain "(?P<text>(?:[^\"]|\\")*)"$/
Then /^(?:|I )should see "(?P<text>(?:[^\"]|\\")*)"
  in the "(?P<element>[^\"]*)" element$/
Then /^(?:|I )should not see "(?P<text>(?:[^\"]|\\")*)"
```



```
Then /^(:|I )should see "(?P<text>(?:[^\"]|\\")*)"
  in the "(?P<element>[^\"]*)" element$/
Then /^(:|I )should not see "(?P<text>(?:[^\"]|\\")*)"
Then /^(:|I )should not see an? "(?P<element>[^\"]*)" element$/
Then /^the "(?P<field>(?:[^\"]|\\")*)" field
  should contain "(?P<value>(?:[^\"]|\\")*)"$/
Then /^the "(?P<field>(?:[^\"]|\\")*)" field
  should not contain "(?P<value>(?:[^\"]|\\")*)"$/
Then /^the "(?P<checkbox>(?:[^\"]|\\")*)" checkbox should be checked$/
Then /^the "(?P<checkbox>(?:[^\"]|\\")*)" checkbox should not be checked$/
Then /^(:|I )should see (?P<num>\d+) "(?P<element>[^\"]*)" elements?$/
Then /^print last response$/
Then /^show last response$/
```

DOING A REAL MINK TEST

<https://github.com/phase2/behaviour-phase2tech>

```
{
  "require": {
    "behat/behat": "2.4.*@stable",
    "behat/mink": "*",
    "behat/mink-extension": "*",
    "behat/mink-goutte-driver": "*",
    "behat/mink-selenium-driver": "*",
    "behat/mink-selenium2-driver": "*"
  },
  "minimum-stability": "dev",
  "config": {
    "bin-dir": "bin/"
  }
}
```

```
default:
  extensions:
    Behat\MinkExtension\Extension:
      base_url: http://www.phase2technology.com/
      goutte: ~
      selenium2:
        browser: "chrome"
      javascript_session: selenium2
```

Feature: Search

In order to look for articles on the Phase2 website
As a website user
I need to be able to search for articles

Scenario: Searching for responsive design articles

Given I am on `"/search"`

When I fill in `"Enter your keywords"` with `"Responsive design"`

And I press `"Search"`

Then I should see `"Blog Entry"`

Scenario: Searching for Drupal blog posts

Given I am on `"/search"`

When I fill in `"Enter your keywords"` with `"drupal"`

And I press `"Search"`

Then I should see `"Blog Entry"`

Scenario: Searching for Clojure articles

Given I am on `"/search"`

When I fill in `"Enter your keywords"` with `"Clojure"`

And I press `"Search"`

Then I should not see `"Blog Entry"`

Feature: Search

In order to look for articles on the Phase2 website

As a website user

I need to be able to search for articles

Scenario Outline: Searching the website

Given I am on `"/search"`

When I fill in `"Enter your keywords"` with `"<keyword>"`

And I press `"Search"`

Then I should see `"<result>"`

Scenario Outline: Searching the website

Given I am on `"/search"`

When I fill in `"Enter your keywords"` with `"<keyword>"`

And I press `"Search"`

Then I should see `"<result>"`

Examples:

keyword	result
Drupal	Blog Entry
smerrill	Presentation
sharepoint drupal	Blog Entry

IMPLEMENTING YOUR OWN STEPS

- behat -dl will show all step definitions
- Steps are written in PHP
- Place them in your Feature Context
- Call other steps inline

Feature: Favicon

In order to have a good user experience

As a website user

I need to be able to distinguish the site by its favicon

Scenario: The favicon is not missing

Given I am on homepage

Then the favicon should be found

```
use Behat\Behat\Context\Step;
```

```
class FeatureContext extends Behat\MinkExtension\Context\MinkContext {  
    /**  
     * Initializes context.  
     * Every scenario gets it's own context object.  
     *  
     * @param array $parameters  
     *     context parameters (set them up through behat.yml)  
     */  
    public function __construct(array $parameters) {  
    }  
}
```

```
/**
 * @Then /^the favicon should be found$/
 */
public function theFaviconShouldBeFound() {
    $favicon_url = "/favicon.ico";
    if ($favicon_link = $this->getSession()->getPage()
        ->find('css', 'link[rel="shortcut icon"]')) {
        $favicon_url = $favicon_link->getAttribute('href');
    }

    return array(
        new Step\Given(sprintf('I go to "%s"', $favicon_url)),
        new Step\Then('the response status code should not be 404'),
    );
}
}
```

DRUPAL BEHAT

<https://github.com/phase2/behhat-drupal-extension>

DRUPAL-SPECIFIC BEHAT STEPS

- Log in (by user/pass or as a test user with a role) or out
- Run cron
- Create nodes
- Install modules
- Teardown tasks to remove all test entries

```
default:
```

```
  extensions:
```

```
    Phase2\Behat\DrupalExtension\Extension:
```

```
      drupal_root: "/var/www/drupal-testing"
```

```
      base_url: "http://behat.dev/"
```

Feature: Permissions

In order to control access

Certain roles need to be restricted from logging in

Scenario: Publishers cannot add blog posts

Given I am logged in as a user with the "Publisher" role

When I go to "/node/add"

Then I should see "Article"

Scenario: Publishers cannot add blog posts

Given I am logged in as a user with the "Publisher" role

When I go to "/node/add"

Then I should not see "Blog Entry"

Scenario: Anonymous users may not add content

Given I am logged out

When I go to "/node/add"

Then the response status code should be 403

RIGHT TOOLS FOR THE JOB

CASPERJS / PHANTOMJS

- Test writers need to know JavaScript (or CoffeeScript)
- Interact with a fully-featured browser
- Use the JavaScript and DOM environment of the browser
- Save screenshots

BEHAT

- Tests can be written or read by non-technical stakeholders
- New step definitions are written in PHP
- Both APIs and browser-based assertions can be tested
- You can use a real browser

BOTH TOOLS

- Integrate well with continuous integration tools
- Can provide detailed assertion results in JUnit XML format
- Provide ways to alter variables per environment

GO FORTH AND TEST!

phase://
TECHNOLOGY

phase2technology.com

@phase2tech