

behat

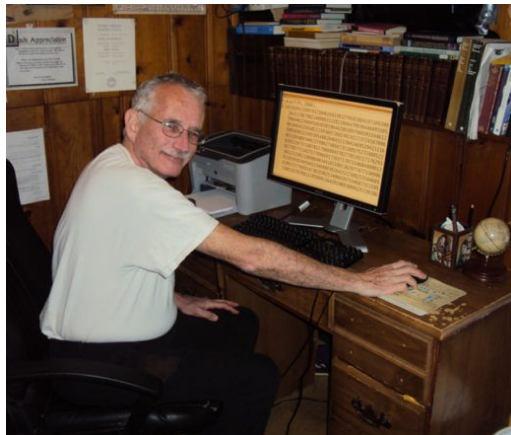
Munk

```
⚙ Feature: Multiple site support

Background:
  Given a global administrator named "Greg"
  And a blog named "Greg's anti-tax rants"
  And a customer named "Wilson"
  And a blog named "Expensive Therapy" owned by "Wilson"

Scenario: Wilson posts to his own blog
  Given I am logged in as Wilson
  When I try to post to "Expensive Therapy"
  Then I should see "Your article was published."

Scenario: Greg posts to a client's blog
  Given I am logged in as Greg
  When I try to post to "Expensive Therapy"
  Then I should see "Your article was published."
```



All code in <https://github.com/Diana0604/moodle-local-custombehattests>

How does behat code look like...?

```
Given the following "courses" exist:  
  | fullname | shortname | category |  
  | Course fullname | C_shortname | 0 |  
And I log in as "admin"  
And I am on site homepage
```

Distribution of code

```
@tag1 @tag2
Feature: Title of feature
  Description of feature

  Background:
    Background steps run for all scenarios

  @javascript
  Scenario: First scenario run with javascript
    Steps for first scenario

  Scenario: Second scenario will be round as background process
    Steps for second scenario
```

Mink extension

And I log in as "admin"



```
/**
 * Logs in the user. There should exist a user with the same value as username and password.
 *
 * @Given I log in as :username
 */
public function i_log_in_as($username) {
    // Visit login page.
    $this->getSession()->visit($this->locate_path('login/index.php'));

    // Enter username and password.
    $this->execute('behat_forms::i_set_the_field_to', array('Username', $this->escape($username)));
    $this->execute('behat_forms::i_set_the_field_to', array('Password', $this->escape($username)));

    // Press log in button, no need to check for exceptions as it will be checked after this step execution.
    $this->execute('behat_forms::press_button', get_string('login'));
}
```

Mink extension

```
<?php
class my_behat_class extends Behat\MinkExtension\Context\RawMinkContext {

    /**
     * Description of first function
     */

    protected function get_some_info($arg) {
        //manipulate $arg to get $info
        return $info;
    }

    /**
     * @Given I do step one on :param
     */
    public function do_this_step($param) {
        //some code
        $var_needed = get_some_info($param);
        //some more code
    }
}
```

Login test

▼ Steps definitions

Type

All the available steps definitions

Area

All the available steps definitions

Contains

log

Filter

```
Logs in the user. There should exist a user with the same value as username and password.  
Given I log in as "USERNAME_STRING"  
behat_auth::i_log_in_as()  
  
Logs out of the system.  
Given I log out  
behat_auth::i_log_out()  
  
Accepts the currently displayed alert dialog. This step does not work in all the browsers.
```

All defined steps are stored in /admin/tool/behat/index.php

Login test

In a few steps we're capable of creating this simple test

```
Feature: Simple Login
  In order to interact with moodle
  I need to log in

  @javascript
  Scenario: Log in with the predefined admin user
    Given I log in as "admin"
```

Docker moodle

☐ Remember username

[Forgotten your username or password?](#)

Cookies must be enabled in your browser

?

Some courses may allow guest access

Log in

Log in as a guest

Login test

In a few steps we're capable of creating this simple test

```
Feature: Simple Login
  In order to interact with moodle
  I need to log in

  @javascript
  Scenario: Log in with the predefined admin user
    Given I log in as "admin"
    And I change window size to "large"
    Then I should see "You are logged in as Admin User" in the "page-footer" "region"
```

 [Moodle Docs for this page](#)

You are logged in as [Admin User](#) ([Log out](#))

[Home](#)

Turn filter on

Manage filters

Filter	Active?	Order	Apply to	Settings	Uninstall
Email protection	Disabled ▾		Content ▾		Uninstall
Display emoticons as images	Disabled ▾		Content ▾	Settings	Uninstall
Glossary auto-linking	Disabled ▾		Content ▾		Uninstall
Multi-Language Content	Disabled ▾		Content ▾	Settings	Uninstall
TeX notation	Disabled ▾		Content ▾	Settings	Uninstall
HTML tidy	Disabled ▾		Content ▾		Uninstall
Convert URLs into links and images	Disabled ▾		Content ▾	Settings	Uninstall
MathType by WIRIS	Disabled ▾		Content ▾	Settings	Uninstall

Turn filter on

The image shows a two-part interface of a rich text editor. The top part displays the source code for a fraction: `<math><mfrac><mn>1</mn><mn>2</mn></mfrac></math>`. The bottom part, indicated by a downward arrow, shows the same editor with the 'Turn filter on' option selected. The source code is hidden, and the rendered fraction $\frac{1}{2}$ is displayed in the text area.

Top toolbar (Source Code view):

- Buttons: Bold (B), Italic (I), List (bulleted), Link (chain), Unlink (chain with slash), Image (picture), Document (file), Microphone (audio), Video (camera), Embed (code), Underline (U), Strikethrough (ABC), Subscript (x₂), Superscript (x²), Indent (left), Outdent (right), Table (grid), Table of Contents (book icon), Undo (curved arrow left), Redo (curved arrow right), Source Code (code icon), and a toggle switch.

Bottom toolbar (Filtered view):

- Buttons: Bold (B), Italic (I), List (bulleted), Link (chain), Unlink (chain with slash), Image (picture), Document (file), Microphone (audio), Video (camera), Embed (code), Underline (U), Strikethrough (ABC), Subscript (x₂), Superscript (x²), Indent (left), Outdent (right), Table (grid), Table of Contents (book icon), Undo (curved arrow left), Redo (curved arrow right), Source Code (code icon), and a toggle switch.

Text area (Top):

```
<math><mfrac><mn>1</mn><mn>2</mn></mfrac></math>
```

Text area (Bottom):

$$\frac{1}{2}$$

Turn filter on

We can use their functions and the test works fine

```
@javascript
```

```
Scenario: Turn filter on using moodle methods
```

```
And I navigate to "Plugins" in site administration
```

```
And I click on "Manage filters" "link"
```

```
And I click on "On" "option" in the "MathType by WIRIS" "table_row"
```

```
And I open my profile in edit mode
```

```
And I click on "HTML" "button"
```

```
And I set the field "Description" to "<math><mfrac><mn>1</mn><mn>2</mn></mfrac></math>"
```

```
And I click on "HTML" "button"
```

```
Then "//img[@alt='1 half']" "xpath_element" should exist
```

Turn filter on

Let's try to write it in a more understandable way...

```
@javascript
Scenario: Turn filter on using moodle methods
  And I navigate to "Plugins" in site administration
  And I click on "Manage filters" "link"
  And I turn "MathType by WIRIS" filter on
  And I open my profile in edit mode
  And I click on "HTML" "button"
  And I set the field "Description" to mathml "1 half"
  And I click on "HTML" "button"
  Then I should see MathType image "1 half"
```

Turn filter on

Let's try to write it in a more understandable way...

```
When I click on "On" "option" in the "MathType by WIRIS" "table_row"  
When I turn "MathType by WIRIS" filter on  
  
And I set the field "Description" to "<math><mfrac><mn>1</mn><mn>2</mn></mfrac></math>"  
And I set the field "Description" to mathml "1 half"  
  
Then "//img[@alt='1 half']" "xpath_element" should exist  
Then I should see MathType image "1 half"
```

Turn filter on

Steps to program

```
<td class="leftalign cell c0" style>MathType by WIRIS</td>
▼<td class="leftalign cell c1" style id="yui_3_17_2_1_1530167990277_20">
  ▼<div class="singleselect d-inline-block" id="yui_3_17_2_1_1530167990277_19">
    ▼<form method="get" action="http://localhost:8000/admin/filters.php" class="form-inline" id="activewiris">
      <input type="hidden" name="sesskey" value="7QsrlMaAAw">
      <input type="hidden" name="filterpath" value="wiris">
      <input type="hidden" name="action" value="setstate">
      <label for="single_select5b3482b4688c644" class="accesshide">
        Active?
      </label>
      ▼<select id="single_select5b3482b4688c644" class="custom-select singleselect" name="newstate">
        <option value="-9999" selected>Disabled</option>
        <option value="-1">Off, but available</option>
        <option value="1">On</option> == $0
      </select>
      ▶<noscript>...</noscript>
    </form>
  </div>
</td>
<td class="centeralign cell c2" style id="yui_3_17_2_1_1530167990277_26"></td>
```

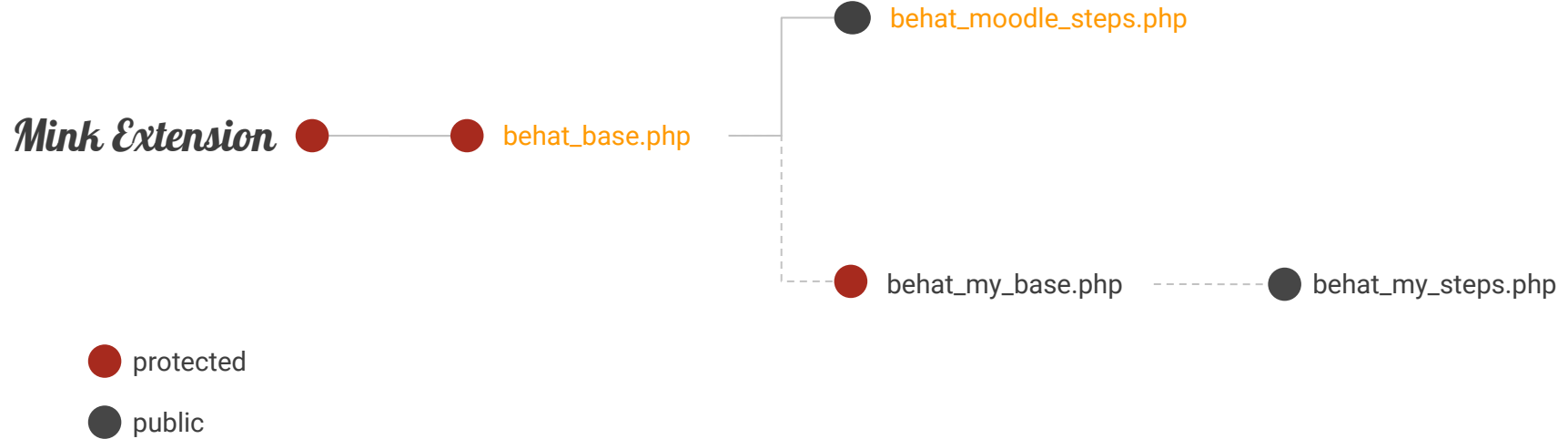
Turn filter on

Steps to program

1. **Reproduce moodle structure**
2. Apply secret technique
3. Run command `php admin/tool/behat/cli/util.php --enable`

Turn filter on

Structure of code



Turn filter on

Simplified version of our code

```
<?php
/*
 * Protected methods
 */

require_once(__DIR__ . '/relative/path/to/lib/behat/behat_base.php');

class behat_custom_base extends behat_base {
    //We don't need any auxiliar methods at the moment
}
```

behat_custom_base.php

```
<?php
/*
 * Public methods to interact with behat
 */

require_once(__DIR__ . '/behat_custom_base.php');

class behat_custom_filter extends behat_custom_base {
    // Public methods to interact with behat
}
```

behat_custom_filter.php

Turn filter on

Steps to program

1. Reproduce moodle structure
2. **Apply secret technique**
3. Run command `php admin/tool/behat/cli/util.php --enable`

```
When I click on "On" "option" in the "MathType by WIRIS" "table_row"  
When I turn "MathType by WIRIS" filter on  
  
And I set the field "Description" to "<math><mfrac><mn>1</mn><mn>2</mn></mfrac></math>"  
And I set the field "Description" to mathml "1 half"  
  
Then "//img[@alt='1 half']" "xpath_element" should exist  
Then I should see MathType image "1 half"
```

Turn filter on

Steps to program

▼ Steps definitions

Type

All the available steps definitions

Area

All the available steps definitions

Contains

click

Filter

Click on the element of the specified type which is located inside the second element.

When I click on "ELEMENT_STRING" "Choose..." in the

"ELEMENT_CONTAINER_STRING" "Choose..."

behat_general::i_click_on_in_the()

Turn filter on

Steps to program

1. Reproduce moodle structure
2. **Apply secret technique**
3. Run command `php admin/tool/behat/cli/util.php --enable`

```
When I click on "On" "option" in the "MathType by WIRIS" "table_row"  
When I turn "MathType by WIRIS" filter on
```

Turn filter on

Simplified version of our code

```
/**
 * Click on the element of the specified type which is located inside the second element.
 *
 * @When I click on :arg1 :arg2 in the :arg3 :arg4
 * @param string $element Element we look for
 * @param string $selectortype The type of what we look for
 * @param string $nodeelement Element we look in
 * @param string $nodeselectortype The type of selector where we look in
 */
public function i_click_on_in_the($element, $selectortype, $nodeelement, $nodeselectortype) {
    $node = $this->get_node_in_container($selectortype, $element, $nodeselectortype, $nodeelement);
    $this->ensure_node_is_visible($node);
    $node->click();
}
```

behat_general.php

```
/*
 * Turns MathType filter on in 'Manage Filters' menu
 * @Given I turn :arg filter on
 *
 * @throws Behat\Mink\Exception\ElementNotFoundException If $xpathnumber is not correctly computed.
 */
public function i_turn_filter_on($filtername) {
    $node = $this->get_node_in_container("option", "On", "table_row", $filtername);
    $this->ensure_node_is_visible($node);
    $node->click();
}
```

behat_custom_filter.php

Turn filter on

Steps to program

1. Reproduce moodle structure
2. **Apply secret technique**
3. Run command `php admin/tool/behat/cli/util.php --enable`

```
And I set the field "Description" to "<math><mfrac><mn>1</mn><mn>2</mn></mfrac></math>"
And I set the field "Description" to mathml "1 half"

Then "//img[@alt='1 half']" "xpath_element" should exist
Then I should see MathType image "1 half"
```

Turn filter on

Simplified version of our code

```
class behat_custom_base extends behat_base {  
  
    const MATHML = "<math><mfraction><mn>1</mn><mn>2</mn></mfraction></math>";  
  
    /**  
     * Return mathml given by alternative text  
     *  
     * @param string $alt -> alternative text  
     *  
     * @throws Exception If alternative is not recognized  
     */  
    protected function alternative_to_mathml($alt) {  
        if ($alt == "1 half") {  
            return self::MATHML;  
        }  
        throw new \Exception('Alternative text not recognized');  
    }  
}
```

behat_custom_base.php

```
class behat_custom_filter extends behat_custom_base {  
  
    /**  
     * @Given I set the field :field to mathml :alt  
     */  
    public function i_set_the_field($field, $alt) {  
        $mathml = $this->alternative_to_mathml($alt);  
        $this->execute('behat_forms::i_set_the_field_to', array($field, $mathml));  
    }  
  
    /**  
     * @Then I should see MathType image :alt  
     */  
    public function i_should_see_image($alt) {  
        $element = "//img[@alt='$alt']";  
        // Getting Mink selector and locator.  
        list($selector, $locator) = $this->transform_selector("xpath_element", $element);  
  
        // Will throw an ElementNotFoundException if it does not exist.  
        $this->find($selector, $locator);  
    }  
}
```

behat_custom_filter.php

Turn filter on

Simplified version of our code

```
<?php
require_once(__DIR__ . '/../../../lib/behat/behat_base.php');

class behat_custom_base extends behat_base {

    const MATHML = "<math><mfrac><mn>1</mn><mn>2</mn></mfrac></math>";

    /**
     * Return mathml given by alternative text
     *
     * @param string $alt -> alternative text
     * @throws Exception If alternative is not recognized
     */
    protected function alternative_to_mathml($alt){
        if($alt == "1 half") return self::MATHML;
        throw new \Exception('Alternative text not recognized');
    }
}
```

behat_custom_base.php

```
<?php

require_once(__DIR__ . '/behat_custom_base.php');

class behat_custom_filter extends behat_custom_base {

    /**
     * Turns MathType filter on in 'Manage Filters' menu
     *
     * @Given /^I turn filter on "(?P<element_string>(?:[^\"]|\\")*)"$/
     * @throws Behat\Mink\Exception\ElementNotFoundException If $xpathnumber is not correctly computed.
     */
    public function i_turn_filter_on($filtername) {
        $node = $this->get_node_in_container("option", "On", "table_row", $filtername);
        $this->ensure_node_is_visible($node);
        $node->click();
    }

    /**
     * @Given I set the field :field to mathml :alt
     */
    public function i_set_the_field($field, $alt)
    {
        $mathml = $this->alternative_to_mathml($alt);
        $this->execute('behat_forms::i_set_the_field_to', array($field, $mathml));
        // $this->set_field_value($field, $mathml);
    }

    /**
     * @Then I should see MathType image :alt
     */
    public function i_should_see_image($alt)
    {
        $element = "//img[@alt='$alt']";
        // Getting Mink selector and locator.
        $list($selector, $locator) = $this->transform_selector("xpath_element", $element);

        // Will throw an ElementNotFoundException if it does not exist.
        $this->find($selector, $locator);
    }
}
```

behat_custom_filter.php

Turn filter on

Let's try to write it in a more understandable way...

```
@javascript
Scenario: Turn filter on using moodle methods
  And I navigate to "Plugins" in site administration
  And I click on "Manage filters" "link"
  And I turn filter on "MathType by WIRIS"
  And I open my profile in edit mode
  And I click on "HTML" "button"
  And I set the field "Description" to mathml "1 half"
  And I click on "HTML" "button"
  Then I should see MathType image "1 half"
```

Turn filter on

Steps to program

1. Reproduce moodle structure
2. Apply secret technique
3. **Run command `php admin/tool/behat/cli/util.php --enable`**

Thank you



diana@wiris.com

All code in <https://github.com/Diana0604/moodle-local-custombehattests>

wiris