
Testing with Behat And Mink

— Presenter: Miss Mansee Jadhav —
Masters - Computer Science

Introduction to Automated testing

- Automated testing tools are capable of executing tests, reporting outcomes and comparing results with earlier test runs.
- Tests can be run repeatedly.
- You can take a look at your code, alter it, and continue to run tests until the actual and expected outcomes align.

Automated testing Versus Manual testing

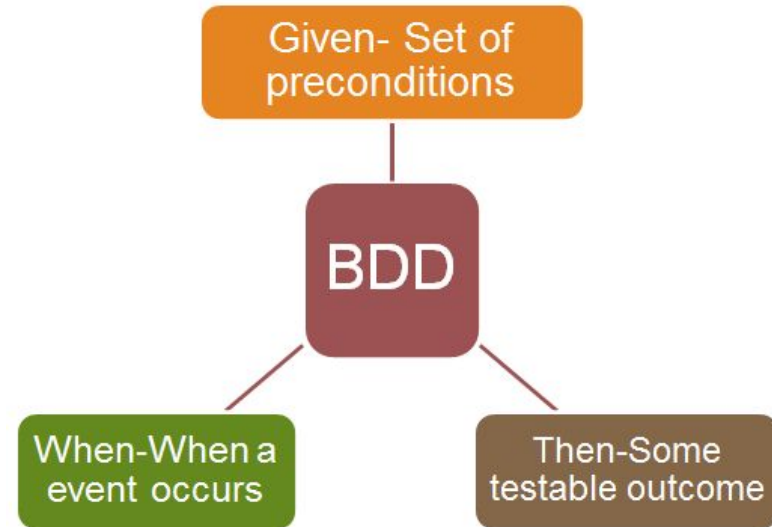
Advantages:

1. Runs tests quickly and effectively.
2. Can be cost effective.
3. More interesting.
4. Everyone can see results.

Behavior Driven testing : Requirements

A good BDD test script should meet following requirements :

1. Define characteristics for the feature.
2. Prioritize features.
3. Describe them with readable scenarios.
4. Then Implement them.



Automated testing with Behat and Cucumber

Behat is an open source Behavior Driven Development framework for PHP.

Cucumber is for Ruby and other web applications.

Both Cucumber and Behat support:

- Documentation availability
- Flexibility in passing parameters
- Steps, scenario and feature scoping
- Hooks and pre-conditions
- Binding to code
- Formatting flexibility

Which one to choose????



Behat IT IS...

It is easily portable and very efficient compared to Cucumber.

Cucumber is outdated now.

GHERKIN is a structured language to define feature in Behat.

MINK: It is a standalone library to use PP to command a "browser". One easy API that can be used to command **Selenium server** etc.

Drupal 7 supports plugin for Behat.

Drupal 8 Behat is in core.

Defining .Feature File

GIVEN- defines the original state of the system.

WHEN- defines the actual action that user takes.

THEN- defines the observable system state after the action is performed.



example.feature

```
1 Feature: Proof of cucumber works
2 Scenario: My first test
3 Given this is my first step
4 When this is my second step
5 Then this is my final step
```


Defining FeatureContext.php File

BEHAT: It maps every step in feature file to PHP Callback.

Behat executes each step in your scenario by calling the functions associated with it.

A **FeatureContext.php** file that holds the PHP callbacks for each step

Defining config file(.yml)



dream

behat.yml

```
1  default:
2    extensions:
3      Behat\MinkExtension\Extension:
4        base_url: http://library.ucla.edu
5        goutte: ~
6        selenium2: ~
7    paths:
8      features: features
9      bootstrap: features/bootstrap
10   annotations:
11     paths:
12       features: features/annotations
13   closures:
14     paths:
15       features: features/closures
```

Installing Behat , Mink and Selenium server

1. PHP

- a. Download XAMPP for Windows or MAMP for Mac or use your preferred method.

2. XAMPP

- ## 3. COMPOSER: Dependency Manager for PHP.
- It is a tool for dependency management in PHP. It allows you to declare the libraries your project depends on and it will manage (install/update) them for you.

For Windows:

1. Download link <https://getcomposer.org/Composer-Setup.exe>
2. Run the executable specifying "C:\xampp\php\php.exe" for "Where is PHP.exe located".

For Mac:

1. Go to: <http://www.abeautifulsite.net/installing-composer-on-os-x/> and follow the on-screen instructions.
2. You may already have composer installed, to check run the command 'Composer' from the Terminal.

What to write where???

- Create a directory at C:\Behat on your local machine (or whatever directory structure you prefer).
- In the Behat folder, create a file called "composer.json" and add the following into it:

composer.json



composer.json

```
{
  "require": {
    "behat/behat": "~3.0",
    "behat/mink": "~1.5",
    "behat/mink-extension": "*",
    "behat/mink-goutte-driver": "*",
    "behat/mink-selenium2-driver": "*"
  },
  "minimum-stability": "dev",
  "config": {
    "bin-dir": "bin/"
  }
}
```

What to write where???

- Install **composer** and **Behat** in newly created folder Behat by typing composer install under ~/Behat directory in command prompt.
- This will create a '**vendor**' folder in your behat folder.

Behat folder structure

```
:behat $ ls -la
drwxrwxrwx  8 root  wheel   272 Mar  4 09:32 .
drwxr-xr-x@ 5 root  wheel   170 Mar  4 08:46 ..
-rw-r--r--@ 1 Ambi  wheel  6148 Mar  4 09:32 .DS_Store
drwxr-xr-x  3 Ambi  wheel   102 Mar  4 09:32 bin
-rw-r--r--  1 Ambi  wheel   336 Mar  4 09:30 composer.json
-rw-r--r--  1 Ambi  wheel  51400 Mar  4 09:32 composer.lock
-rwxr-xr-x  1 Ambi  wheel 1577346 Mar  4 09:20 composer.phar
drwxr-xr-x 11 Ambi  wheel   374 Mar  4 09:32 vendor
```

What to write where???

- Add the Behat path "C:\Behat\bin" => YOUR BEHAT DIR to your system path variable.
- Make Behat Global
 - Create a global symlink to access behat from anywhere by setting a sym link: ("/opt/behat/" = YOUR BEHAT DIR)

```
:behat $ sudo ln -s /opt/behat/bin/behat /usr/local/bin/behat
```

- Test that Behat has installed correctly



```
:bin $ ./bin/behat --version  
behat version 3.1-dev
```

behat --init Command

This will create:

- a features folder,
- a bootstrap folder inside the features folder and
- the FeatureContext.php file inside the bootstrap folder.

```
behat --init
```

```
:behat $ behat --init  
+d features - place your *.feature files here  
+d features/bootstrap - place your context classes here  
+f features/bootstrap/FeatureContext.php - place your definitions, transformations and hooks here
```

Installing Behat , Mink and Selenium server- Complete Documentation..

<https://docs.library.ucla.edu/pages/viewpage.action?spaceKey=~mansee8&title=How+to+install+behat+and+mink>

Installing selenium server

- Required for running Remote Selenium WebDriver.
- Making use of the latest and greatest features of the WebDriver InternetExplorerDriver.
- install selenium standalone server from <http://www.seleniumhq.org/download/>
- **Point to remember:** Kindly download Selenium server compatible with PHP. The one we downloaded is **PHP by Lukasz Kolczynski**.

Installing selenium server..continued

- Now open a new Command prompt and set path= Path where your **JAVA compiler and JAVA jre** is located.
 - For example on my computer the path where JAVA files are located is "C:\Program Files\Java\jre1.8.0_60\bin";
- So the command in command prompt will be set path= "C:\Program Files\Java\jre1.8.0_60\bin";
- Start the selenium server using:

Command Prompt

```
C:\xampp\php>java -jar selenium-server-standalone-2.52.0.jar
```

Selenium server Running!!!

Command Prompt

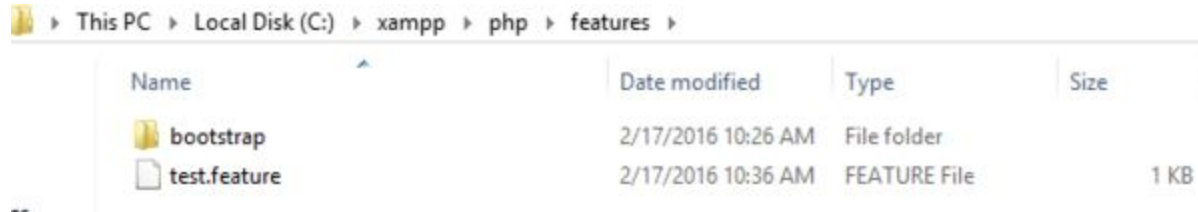
```
C:\xampp\php>java -jar selenium-server-standalone-2.52.0.jar
10:34:32.927 INFO - Launching a standalone Selenium Server
10:34:33.002 INFO - Java: Oracle Corporation 25.60-b23
10:34:33.002 INFO - OS: Windows 8.1 6.3 amd64
10:34:33.015 INFO - v2.52.0, with Core v2.52.0. Built from revision 4c2593c
10:34:33.081 INFO - Driver class not found: com.opera.core.systems.OperaDriver
10:34:33.081 INFO - Driver provider com.opera.core.systems.OperaDriver is not registered
10:34:33.092 INFO - Driver provider org.openqa.selenium.safari.SafariDriver registration is skipped:
registration capabilities Capabilities [{browserName=safari, version=, platform=MAC}] does not match the current platform WIN8_1
10:34:33.209 INFO - RemoteWebDriver instances should connect to: http://127.0.0.1:4444/wd/hub
10:34:33.209 INFO - Selenium Server is up and running
```



Now You're All Set!

Examples - Behat testing with Selenium

- **Test.feature file** which check the search functionality by populating the main search box of library site.
- Steps that this test performs are:
 - 1. It opens the browser.
 - 2. Navigates to Library.ucla.edu site.
 - 3. Populates the site search field with books.
 - 4. Searches the site for books.

Placement of this test.feature file



Name	Date modified	Type	Size
 bootstrap	2/17/2016 10:26 AM	File folder	
 test.feature	2/17/2016 10:36 AM	FEATURE File	1 KB

Test.feature file

@javascript is mandatory for features file to coordinate with selenium and open up the browser and test functionality.

test.feature

```
1 Feature: Book search
2   In order to find books on library website
3   As a Library user
4   I need to be able to use Library services
5   @javascript
6   Scenario:
7     Given I go to "http://library.ucla.edu/"
8     And I fill in "Site Search" with "books"
9     And I press "Search"
10    Then I should see ""
```

Running your test

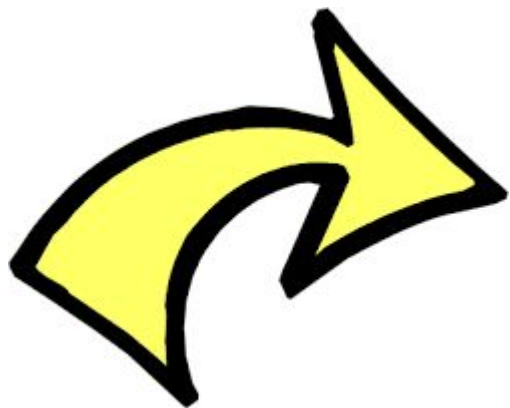


Command Prompt

```
1 C:\xampp\php>set path="C:\manseefinal\bin";
2 C:\xampp\php>behat --config behat.yml features\test.feature
3 Feature: Book search
4   In order to find books on library website
5     As a Library user
6     I need to be able to use Library services
7     @javascript
8     Scenario:                                     # features\test.feature:7
9       Given I go to "http://library.ucla.edu/" # FeatureContext::visit()
10      And I fill in "Site Search" with "books" # FeatureContext::fillField()
11      And I press "Search"                      # FeatureContext::pressButton()
12      Then I should see ""                     # FeatureContext::assertPageContain
13 sText()
14 1 scenario (1 passed)
15 4 steps (4 passed)
16 0m18.056s
```


Making changes to FeatureContext.php

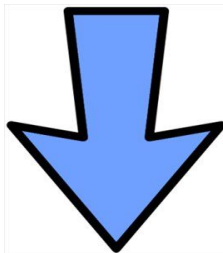
Feature: I wait for suggestion box to appear



```
/**
 * @Given /^I wait for the suggestion box to appear$/
 */
public function iWaitForTheSuggestionBoxToAppear()
{
    $this->getSession()->wait(5000,
        "$('.suggestions-results').children().length > 0"
    );
}
}
```

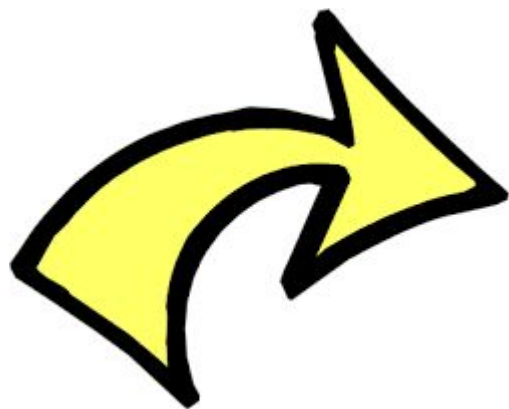
Making changes to FeatureContext.php

Feature: I hover over the element



```
/**
 * @When /^I hover over the element "([^"]*)"$/
 */
public function iHoverOverTheElement($locator)
{
    $session = $this->getSession(); // get the mink session
    $element = $session->getPage()->find('css', $locator); // runs the actual query and returns the element
    // errors must not pass silently
    if (null === $element) {
        throw new \InvalidArgumentException(sprintf('Could not evaluate CSS selector: "%s"', $locator));
    }
    // ok, let's hover it
    $element->mouseOver();
}
```

Feature: I select
The radio button.



```
/**
 * @When /^I select the "([^"]*)" radio button$/
 */
public function iSelectTheRadioButton($labelText)
{
    // Find the label by its text, then use that to get the radio item's ID
    $radioId = null;
    $ctx = $this->getMainContext();
    /** @var $label NodeElement */
    foreach ($ctx->getSession()->getPage()->findAll('css', 'label') as $label) {
        if ($labelText === $label->getText()) {
            if ($label->hasAttribute('for')) {
                $radioId = $label->getAttribute('for');
                break;
            } else {
                throw new \Exception("Radio button's label needs the 'for' attribute to be set");
            }
        }
    }
    if (!$radioId) {
        throw new \InvalidArgumentException("Label '$labelText' not found.");
    }
    // Now use the ID to retrieve the button and click it
    /** @var NodeElement $radioButton */
    $radioButton = $ctx->getSession()->getPage()->find('css', " #$radioId");
    if (!$radioButton) {
        throw new \Exception("$labelText radio button not found.");
    }
    $ctx->fillField($radioId, $radioButton->getAttribute('value'));
}
```

Making changes to FeatureContext.php

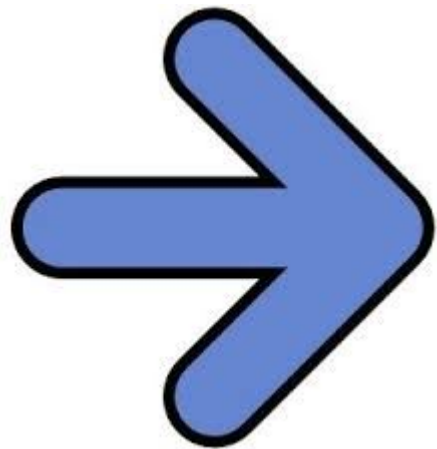
Feature: I press back/forward/reload button in the browser



```
/**
 * @Then /^I press the "([^"]*)" button in the browser$/
 */
public function iPressTheButtonInTheBrowser($arg1)
{
    $session = $this->getSession();
    if ($arg1 == 'back') {
        $session->back();
    } else if ($arg1 == 'forward') {
        $session->forward();
    } else if ($arg1 == 'reload') {
        $session->reload();
    } else {
        throw new ExpectationException('Unknown browser button.', $session);
    }
}
```

Making changes to FeatureContext.php

Feature: Wait * seconds



```
/**
 * @Given /^wait (\d+) second$/
 */
public function waitSecond($arg1)
{
    $this->getSession()->wait(1000*$arg1);
    //throw new PendingException();
}
```

DEMO

Limitations and future Work

Getting it to work with Drupal and Behat plugin and Travis.

Performance issues

Finding people to write user stories.

Future Scope:

**THANK
YOU!**

Questions?????