



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences



Automated testing in Moodle

Selenium, Behat, Gherkin, Mink – what is this all about?

► Department for Higher Education Didactics & E-Learning

Software development tec

However... yc

- ▶ Test-driven development



e testing

- ▶ Behavior-driven development

Why automated testing?

- ▶ Manually repeating these tests is costly and time consuming
- ▶ Executing repetitive tasks with automated software testing frees time for other things
- ▶ (like writing more automated tests)

Selenium

the Selenium (Java) way

https://www.chalkstreet.com/selenium-online-tutorial/

The screenshot shows a web browser displaying the ChalkStreet website. The URL in the address bar is https://www.chalkstreet.com/selenium-online-tutorial/. The page features a navigation bar with the ChalkStreet logo, a search bar, and links for 'Login' and 'Signup'. Below the navigation bar, the course is categorized under 'Technology / Software Testing'. Two buttons are visible: 'Watch Demo' and 'Join Course'. The price is listed as 'Rs. 1999 Rs. 599'. A video player shows a woman speaking, with a play button overlay. The video title is 'Learn Selenium for Automated Web Testing'. The description states: 'Created by Stanford alumni team, this comprehensive tutorial teaches Selenium using 45 solved examples on Automating Web Testing. You will learn to work with Loonycorn'. The video duration is 05h:44m, and it offers 'Lifetime access'. There are 36 learners enrolled. The 'About this course' section describes the course content, mentioning Selenium's Java API, 45 examples, and topics like web application testing, scraping, and Selenium Grid. A green button at the bottom right says 'Need some help?'.

Technology / Software Testing

Watch Demo Join Course

Rs. 1999 Rs. 599

Learn Selenium for Automated Web Testing

Created by Stanford alumni team, this comprehensive tutorial teaches Selenium using 45 solved examples on Automating Web Testing. You will learn to work with Loonycorn

05h:44m Lifetime access 36 learners

About this course

In this Selenium for Automated Web Testing course, you'll be working with Selenium's Java API to test browser functionality and automate tasks using nearly 45 solved examples on Automating Web Testing. This course will teach you to test the functionality of web application and to simplify a boring or a repetitive task.

By the end of the course, you will also learn to scrape websites for specific elements identified by HTML tags and CSS selectors. Starting from checking the title of a web page and exploring a web page with developer tools to interacting with a video player and running a cross-browser test with selenium grid, this course teaches everything you need to know to automate web testing using Selenium.

Need some help?

Selenium the Selenium (Java) way

- ▶ From your Java IDE (Integrated Development Environment)
- ▶ Having a browser installed
- ▶ Run Java code which invokes Selenium and executes actions in the browser

Selenium the Selenium (Java) way

The screenshot displays the Eclipse IDE interface with the following components:

- Package Explorer:** Shows the project structure with a package named `p6_Screenshots` and a class `ex35`.
- JUnit Test Results:** Indicates the test `p6_Screenshots.ex35` finished after 5.253 seconds with 1/1 runs, 0 errors, and 0 failures.
- Source Editor:** Contains the Java code for `ex35.java`, including package declarations, imports, and test methods like `setUp()`, `testDriver()`, and `cleanUp()`.
- Outline:** Lists the classes and methods in the current file, such as `driver : WebDriver`, `setUp() : void`, `testDriver() : void`, and `cleanUp() : void`.
- Console:** Shows the execution output, including the message `Starting ChromeDriver 2.29.461585` and the Selenium version `org.openqa.selenium.remote.ProtocolHandshake createSession`.
- Task List:** A panel on the right side of the IDE, currently empty.
- Updates Available:** A notification in the bottom right corner stating "Updates are available. Click to review. Set up Remind".

Look and see

Automated testing

the Moodle way

Automated testing the Moodle way

- ▶ I suppose there was no way Moodle developers would write clunky and long Java files to cover all their tests.

Automated testing the Moodle way

- ▶ [https://mootus.moodle.org/pluginfile.php/2182/mod_data/content/8143/Behat for Beginners by a Beginner.pdf](https://mootus.moodle.org/pluginfile.php/2182/mod_data/content/8143/Behat%20for%20Beginners%20by%20a%20Beginner.pdf)
- ▶ [https://mootau.moodle.org/pluginfile.php/2201/mod_data/content/12810/Automated web acceptance testing with Behat - MOOTAU15.pdf](https://mootau.moodle.org/pluginfile.php/2201/mod_data/content/12810/Automated%20web%20acceptance%20testing%20with%20Behat%20-%20MOOTAU15.pdf)

NC STATE UNIVERSITY


Behat for Beginners by a Beginner

Presented By: Steve Bader

```
@mootus16 @behat
Feature: Presentation
  In order to spread basic knowledge of Behat
  As a developer / presenter named Steve Bader
  I need to cover several topics

@javascript
Scenario: Explain Why Behat?
  Given there are more than 2 members in audience
  And I finished preparing the presentation
  And I present the following topics:
    | Topic Name | Number of Slides |
    | What is Behat? | 11 |
    | Moodle: Why use Behat? | 2 |
    | Setting up Behat with Moodle | 8 |
    | Running Moodle Feature Files | 5 |
    | Looking ahead to Behat 3.0 | 3 |
  When I finish the presentation
  Then at least 2 audience members will understand Behat
```

Distance Education and Learning Technology Applications (DELTA)



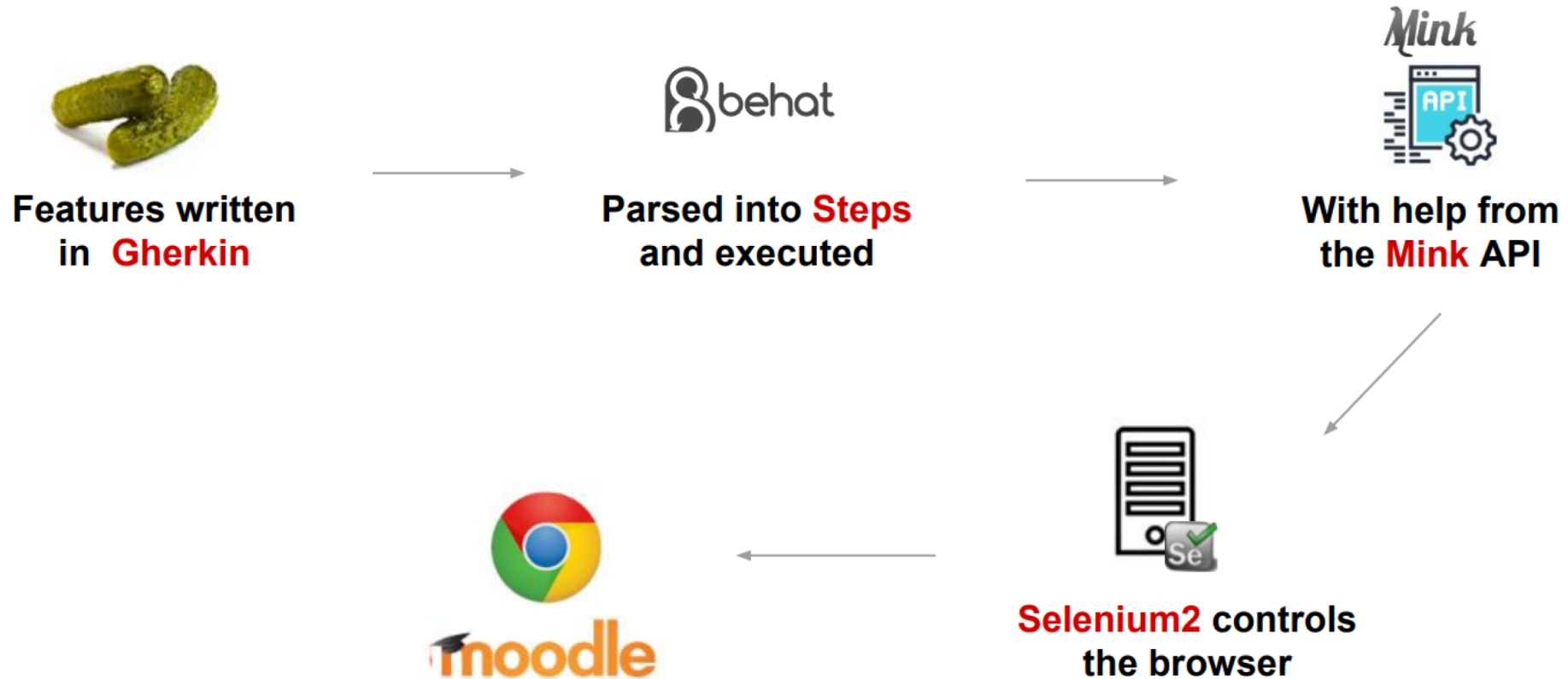
the world's open source learning platform

Automated web acceptance testing with Behat

Rajesh Taneja
@TanejaRaji

MoodleMoot Australia 2015
#mootau15

Automated testing the Moodle way



Automated testing the Moodle way

All those *.feature files in
test/behat directories



Features written
in **Gherkin**

Run behat tests

```
php admin/tool/behat/cli/run.php ...
```



Parsed into **Steps**
and executed



With help from
the **Mink** API



Selenium2 controls
the browser

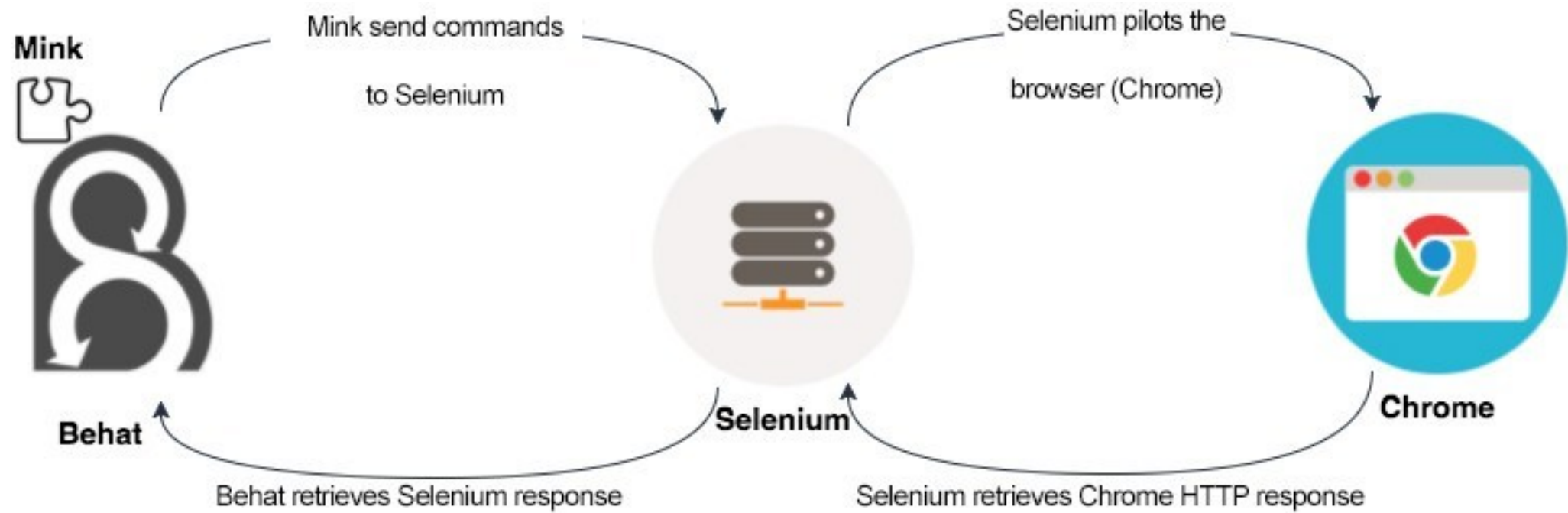
Start your Selenium Server
(which waits for commands)



Watch the tests running

Automated testing the Moodle way

In more details



Look and see

Automated testing the Moodle way

Useful sources

https://docs.moodle.org/dev/Behat_integration

https://docs.moodle.org/dev/Running_acceptance_test

https://docs.moodle.org/dev/Writing_acceptance_tests

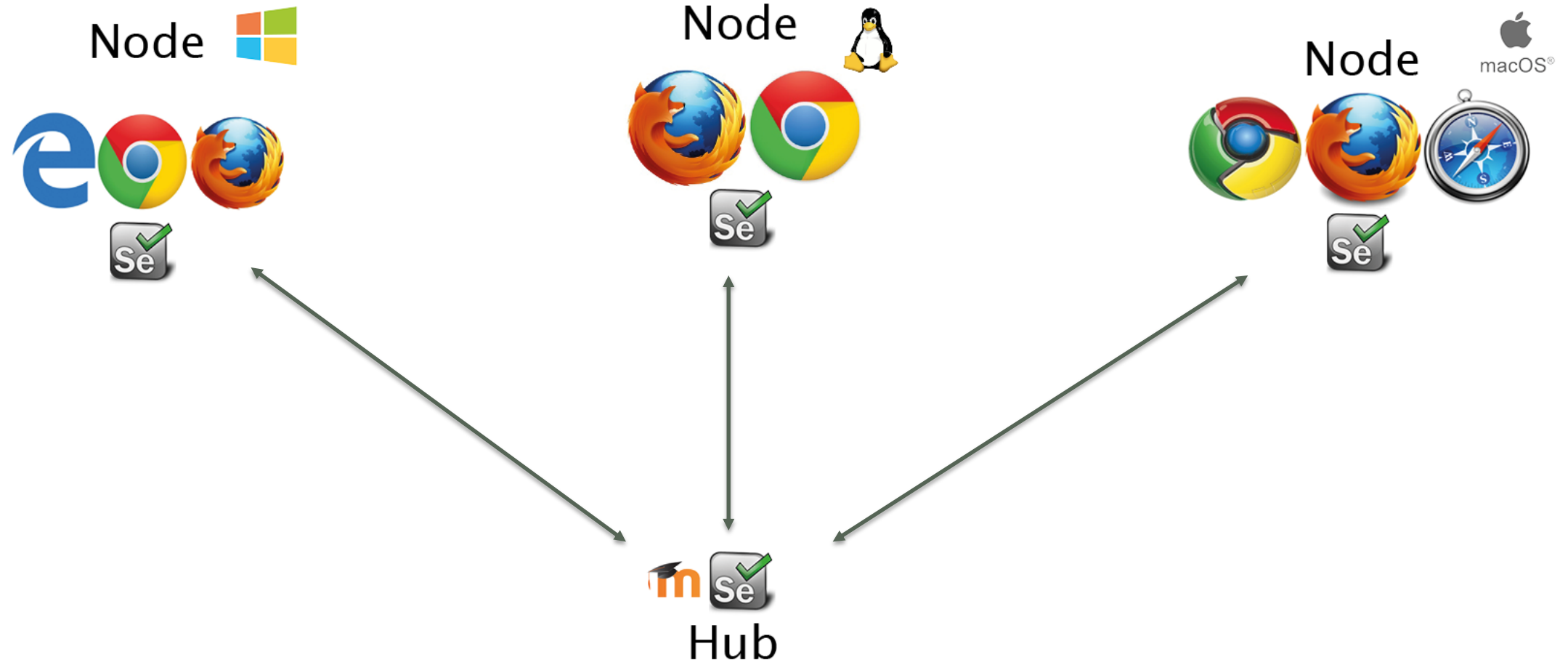
[https://mootus.moodlemoot.org/pluginfile.php/2182/mod_data/content/8143/Behat for Beginners by a Beginner.pdf](https://mootus.moodlemoot.org/pluginfile.php/2182/mod_data/content/8143/Behat_for_Beginners_by_a_Beginner.pdf)

[https://mootau.moodlemoot.org/pluginfile.php/2201/mod_data/content/12810/Automated web acceptance testing with Behat - MOOTAU15.pdf](https://mootau.moodlemoot.org/pluginfile.php/2201/mod_data/content/12810/Automated_web_acceptance_testing_with_Behat_-_MOOTAU15.pdf)

But this was just standalone

We want to distribute the tests and run browsers on other OS

Setup with hub and nodes



Look and see

Thank you

luca.boesch@bfh.ch