COURSEWARE

- Professional Skills
- Agile Fundamentals
- Jira
- Git
- Databases Introduction
- Java Beginner
- Maven
- Testing (Foundation)
- Java Intermediate
- HTML
- CSS
- Javascript
- Spring Boot
- Selenium
  - What is Selenium?
  - Selenium IDE
  - Webdrivers
  - Opening A Web Browser With Selenium
  - Browser manipulation
  - Interacting With Elements In Selenium
  - The POM Design Pattern
  - Actions
  - Waits
- Sonarqube
- Advanced Testing (Theory)
- Cucumber
- MongoDB
- Express
- NodeJS
- React

What is Selenium?

Contents

- Overview
- What is Selenium?
  - What is the purpose of Selenium?
  - What is test automation and why is it done?
- What other technologies can be used in conjunction with Selenium?
- I've heard Selenium is expensive to run...
- Types and levels of testing
- Key points
- Tutorial
- Exercises

Overview

Selenium is a collection of tools and libraries used to automate web browsers, usually used for testing purposes.

Webdrivers can be incredibly complex, and trying to interact with them directly would be very tedious. Therefore Selenium provides a single interface through which we will interact with - acting as a facade from which the complex webdriver's extend.

What is Selenium?

The Selenium Browser Automation Project is a free **toolset** that allows the automated control of browser instances alongside emulating user interactions. This could be as simple as opening a chrome browser, navigating to a website, and logging in... or as complex as a full weekly shop.

The Selenium toolset consists of:

- Selenium IDE (An interactive web browser extension for no/low-code testing)
- WebDriver (A vendor-supplied driver that abstracts the process of communicating with the different kinds of browser)
- Grid (A tool for running parallel tests in Selenium, potentially across multiple devices)

What is the purpose of Selenium?

The primary purpose of Selenium is to automate the front-end testing of websites, the Selenium documentation describes it as being a *browser user agent library*. Selenium is a useful tool for automating **User Acceptance Testing**, a form of end-user testing that validates the software system meets the required specification.

What is test automation and why is it done?

Have you ever tried to manually test a website? Needless to say, it is a laborious process involving many thousands of mouse clicks and key presses... Test automation, in terms of Selenium, makes this process automatic; it is carried out by the computer as many times as we want.

Express-Testing
Networking
Security
Cloud Fundamentals
AWS Foundations
AWS Intermediate
Linux
DevOps
Jenkins Introduction
Jenkins Pipeline
Markdown
IDE Cheatsheet

> A simple set of manual tests for one web page could take hours, and that's just for a single browser. If the whole world of technology operated in this fashion, our webpages would probably still look like something from the early 2000s as we'd likely still be testing them.

All jokes aside, manually testing websites is a waste of money, time, and resources in most cases – reserve manual testing for websites that may be subject to frequent change, this prevents the overapplying of automated testing. Using automated testing when it is not necessary can also lead to high costs like manual testing can.

The Selenium Browser Automation Project can be used to automate the testing process of a websites functionality, this means a computer program will carry out the actions against a website instead of an actual person – Selenium emulates the user to perform user acceptance testing (UAT), an acceptance level functional test.

What other technologies can be used in conjunction with Selenium?

We can also use Selenium in combination with other libraries such as Cucumber (Executable specification tool). Cucumber can be used to create automatable executable specifications for testing that can be understood across many different areas of the business domain, it particularly aims to close the gap between business analysts, software developers and testers by using the domain specific language called Gherkin. These executable specifications can be used to automate our Selenium tests.

I've heard Selenium is expensive to run...

Functional browser testing is an expensive process, but the Selenium software is free! The expense comes from power used over time; Selenium tests automate what the user would be doing on actual running instances of a browser, when testing across multiple different browsers and devices, the expenses soon start to mount up. Don't despair though, if browser testing is required within our software systems, more money is often saved by running tests and finding defects earlier on, before they have the chance to impact the business outcomes and thus the value perceived by a customer.

Types and levels of testing

The following information is not essential to Selenium, but will provide a good basis for understanding how Selenium fits into the overall topic of testing.

There are many different types and levels of test, these help us to categorise what each type of test we write is doing. There are four basic **test levels** to be aware of as indicated by the ISTQB syllabus:

- Component testing (focused on separately testable components, often performed by the developer)
- Integration testing (focused on interactions between components or systems)
- System testing (focused on the behaviour and capabilities of a system)
- Acceptance testing (focused on the behaviours and capabilities of a system in terms of satisfying user and legal requirements)

Test levels have the following *attributes* as defined by the ISTQB syllabus:

- Specific objectives
- Test basis, referenced to derive test cases
- Test object

- Typical defects and failures
- Specific approaches and responsibilities

The following **test types** can be performed at any of the above-mentioned test levels:

- Functional testing (evaluates the behaviours of a system against a functional specification)
- Non-functional testing (evaluates the characteristics of a system such as usability, security, performance, etc...)
- White-box testing (tests based on the internal structure or implementation of a system, thoroughness of the tests can be captured via code coverage. Generally unit tests)
- Black-box testing (software tests where the internal structure or implementation is not known, i.e. does the software do what it is supposed to)
- Change-related testing (confirms changes have corrected defects or have not caused any new defects)

Selenium user acceptance testing can be viewed as a type of functional, black-box test operating at the acceptance level as a websites behaviours are tested against the users acceptance criteria (functional requirements) listed in the services specification *without* knowing how the website works internally (Selenium emulates user interactions with a browser, not with code).

Key points

- Selenium is a free toolset for automating browser-based tests
- Selenium is particularly useful for automating User Acceptance Testing (UAT)
- UAT with Selenium is a type of black-box functional test at the Acceptance level
- Many different types and levels of test exist

Tutorial

There is no tutorial for this module

Exercises

There are no exercises for this module