

Test Automation with Java and Selenium Training

Duration: 2 Days

Description

This course is designed for participants who want to automate software testing processes. Starting with the fundamentals of test automation, it covers the Selenium ecosystem, JUnit 5, TestNG and design patterns.

Participants gain hands-on experience in developing test scenarios, building automation frameworks, reporting, running tests in parallel and applying professional approaches such as the Page Object Model in Java projects. The course combines a theoretical foundation with practical work based on real scenarios.

Who Should Attend?

- Test engineers, quality assurance specialists and business analysts
- Systems analysts, programmers and software specialists
- Project managers
- Enterprise architecture specialists
- Software architects

Outline

Introduction to Test Automation

- What is software testing?
- The purpose and benefits of software testing
- Test design and preparing test cases
- Why write test cases?

What is Selenium?

Selenium IDE

- Panels and the recording mechanism
- Test debugging and step-by-step execution

- Using breakpoints and the debugger command
- Pause on Exception
- Find Target in a Page and Select Target in a Page
- Running tests

Selenium WebDriver

- Navigating applications with Selenium
- Chrome Developer Tools
- Locator types and their use
- Synchronization: implicit waits, explicit waits and Fluent Wait
- Identifying complex structures with locators
- Working with different elements in practical scenarios
- Radio buttons and checkboxes
- Selecting images and multiple elements
- LinkText and PartialLinkText
- Dropdowns
- Mouse actions: MoveToElement, DragAndDrop, ContextClick and DoubleClick
- Uploading and downloading files
- Pop-up messages and tabs
- Validating tooltips and scrolling pages
- Broken links
- The Actions class
- Taking screenshots
- Debugging and identifying errors

JUnit 5

- Setting up the framework
- Annotations and writing tests
- Hard assertions

TestNG

- Setting up the framework
- Differences from JUnit 5 and advantages
- Managing and packaging tests with XML
- Setting test priorities and timeouts
- Test dependencies and repeated tests
- Parameterized tests and Data Providers
- Running tests in parallel

- Test reporting and soft assertions

Page Object Model (POM)

- A brief introduction to object-oriented programming
- Definition and purpose of the POM pattern
- Implementing POM

Page Factory

- Definition and purpose of the Page Factory pattern
- Implementing Page Factory
- Simplifying code with @FindAll