Automated Testing with Selenium WebDriver for Java Developers

Course Number: TN-SEL-WEB-JAVA

Duration: 3 days

Course Description

This Selenium WebDriver training teaches Java Developers how to create Selenium automated test scripts for simple to complex SPA web applications. Learners discover how to use the flexibility of Java test scripts to direct Selenium in testing all aspects of a web application. The training starts off with locating and exercising simple elements on a web page. Learners then proceed to complex tasks including efficiently waiting for Ajax calls to complete, structuring projects to the POM for maintainability, and moving to new windows within the same application.

- This course is over 65% hands-on labs where learners apply their newly acquired real-world skills to develop Selenium automated Java tests.
- Note: This class uses Selenium v. 3 and 4 in the Windows environment.

Skills Gained

- Explain the major building blocks of the Selenium tool suite.
- Use Selenium WebDriver to write web application automated test scripts in Java.
- Control test script execution with waits.
- Leverage Selenium Advanced User Interactions.
- Utilize advanced Selenium features including popup-windows, Automating Ajax applications, and taking screenshots.
- Exercise the Katalon IDE recorder as a prototype productivity tool.

- Conform test scripts to the Selenium POM Pattern and PageFactory to produce maintainable, reusable, automated tests scripts.
- Recognize Selenium version considerations and the migration process between Selenium 2, Selenium 3, and Selenium 4.
- Interpret Selenium's Roadmap including the ramifications of the WebDriver W3C Specification.

Who Can Benefit

Experienced Java developers working as Web Developers, Test Engineers or other IT Personnel who want to learn how to use Selenium to automate Web Application tests.

Prerequisites

This is a hands-on programming course designed for experienced Web and Test Engineer Developers who are new to Selenium. **Students must be able to develop Java Application in an IDE like Eclipse.** Knowledge of the following is helpful:

- Web technologies
- Testing concepts
- Knowledge of HTML DOM and Object-Oriented Development

Course Details

Introduction to Test Automation

- Variations of Test Automation
- Test Harness Tools
- Benefits of Test Automation
- What to test
- The Testing Pyramid
- Test Data

Introduction to Selenium

- Selenium Suite of Testing Tools
- History of Selenium
- Selenium Browser Drivers
- Version Considerations
- W3C WebDriver Specification
- W3C Browser Automation Protocol
- Selenium Roadmap
- Downloading and Installing Selenium

- Lab: Working with Selenium Drivers

Selenium WebDriver

- Introduction to WebDriver
- Test script as Java code
- The Selenium WebDriver library
- The Main WebDriver classes
- WebDriver Capabilities
- Setting up Selenium WebDriver projects
- SeleniumManager
- Using Maven and Gradle build tools
- Lab: First WebDriver Project

Working with Selenium WebElements

- WebElements as Nodes in the DOM
- Locating WebElements on the Web Page
- Using Katalon Recorder as a Prototype Time Saver
- Interacting with WebElements
- Retrieving WebElements
- Verifying WebElements
- Submitting a Form
- Lab: Locating and Updating WebElements

Working with Selenium Clickable WebElements

- Working with Clickable WebElements
- Handling Radio buttons and Check Boxes
- Working with Select Drop Down Elements
- Lab: Working with Clickable WebElements

Accessing Links Using Selenium WebDriver

- Exact and Partial Link Text Match
- Case-sensitivity
- Using CSS to Locate WebElements
- Lab: Working with Links

Accessing Tables Using Selenium WebDriver

- HTML Tables Structure
- Advanced Strategies for Locating WebElements
- Using XPath to find WebElements
- Table Utilities
- Lab: Working with Tables

Page Object Model (POM) Pattern

- Introduction to POM (Page Object Model)
- Making Automated Test Scripts Maintainable
- POM Benefits
- How to Refactor a Project to POM
- Lab: Refactoring to the Page Object Model

Testing Frameworks

- Testing Frameworks
- Modular
- Data Driven
- Keyword-Driven
- Hybrid
- Using Apache POI to Access Test Data
- Optional Lab: Refactoring to Keyword-Driven Framework

The Selenium Page Factory

- Introduction to the Page Factory
- PageFactory implicit Instantiations
- Page Factory Annotations
- Lazy Instantiations and Caching
- Lab: Refactoring to the PageFactory

Working with Multiple Windows

- Working with alert pop-ups
- Switching between multiple windows
- Working with iFrames
- Dealing with OS Generated pop-ups
- Lab: Working with Multiple Windows

Controlling Test Execution

- Waiting on a Web Page Element
- Implicit Waits
- Explicit Waits
- Specifying Expected Conditions
- Working with Ajax Controls Using WebDriver
- Overview of Ajax
- Issues testing Ajax model
- Pros and Cons of different methods to wait on Ajax
- Lab: Controlling Test Execution

Advanced User Interactions

- Introduction to Advanced User Interactions
- Keyboard & Mouse Events
- Building a Series of Multiple Actions
- Lab: Advanced User Interaction

Advanced Web Application Access

- Taking Screenshot Using WebDriver
- Working with Colors
- HTML5 Considerations
- Handling HTML5 Date Time Picker Using Selenium
- Working with Non-Text Pressable Keys or Unicode PUA Code points
- Menu Selection
- Lab: Advanced Web Application Access

Selenium Wisdom

- Selenium Guidelines and Recommendations
- Selenium Worst Practices to Avoid
- Selenium References