Sela.

seleniumDepthWS

Selenium in Depth Workshop

college@sela.co.il 03-6176666

Selenium in Depth Workshop

seleniumDepthWS - Version: 1

1 days Course

Description:

Selenium is a very simple and straight-forward technology to use. However, if you dig deeper into it, you'll find some hidden gems that can be very handy in making your tests more robust and maintainable. In addition, knowing Selenium more deeply will help you investigate and understand failures, and resolve many of them which you might consider as not-reproducible or unexplained. Even if you think that you know Selenium, you'd be surprised by some of the demos!

In this workshop we'll dive deeply into Selenium and understand:

* How it works
* The difference between ImplicitWait and Explicit wait – when to use what (if any), and how it's related to the web-page's JavaScript
* The different exception types that can be thrown from Selenium, and what they mean
* How to collect valuable information that will help you investigate failing tests, including the browser logs, screenshots, page source and more
* How to execute JavaScript code from your tests, synchronously or asynchronously, and how to pass parameters to it
* And more…

Note: This course mainly focuses on the technical sides of Selenium. For a course on best practices for test automation see the course Advanced Test Automation Practices. It is recommended to combine these courses together.

Intended audience:

Prerequisites:

Basic experience working with Selenium WebDriver.

Objectives:

Gain deep understanding of how Selenium works
Know how to choose the most appropriate locator
Know how and when to use Implicit and Explicit wait, and how to use them effectively for making your tests more reliable
Understand exceptions thrown by Selenium (including StaleElementReferenceException) and know how to solve them
Know how to execute JavaScript code from within the test code to achieve things that you cannot otherwise
Know how to work with multiple windows and iFrames
Gain tips for investigating failures more effectively using Selenium
Know how to implement the Page Object pattern correctly
Understand Selenium Grid, its benefits and limitations, and how to use it

Topics:

Selenium Overview

- Introduction to Selenium
- Overview of the DOM
- Selenium Architecture: bindings and drivers

Locators

- XPath and CssSelectors – when to use and why
- Searching for elements within other elements

- Finding multiple elements
- Best practices for choosing locators

Delays and waits

- Why is my browser stuck?
- Are waits always necessary?
- Understanding browser's asynchronous operations
- The drawbacks of fixed delays
- When Implicit Wait helps and when it doesn't?
- How FindElements behaves when ImplicitWait is used?
- Using Explicit Wait
- Writing custom wait conditions

Understanding Selenium exceptions

- NoSuchElementException
- Understanding StaleElementReferenceException
- Reasons for StaleElementReferenceException
- WebDriverTimeoutException

Frames and Windows

- Frames and the DOM
- Working with multiple Frames and Windows
- Understanding SwitchTo
- Why do I get StaleElementReferenceException now?
- The TestAutomationEssentials solution

Executing JavaScript

- Overview on ExecteJavaScript
- Passing arguments to ExecuteJavaScript
- Returning values from ExecuteJavaScript
- Passing and returning elements to/from ExecuteJavaScript
- Executing asynchronous scripts

Investigating Failures

- Taking screenshots
- Saving the Page source
- Getting the browser's logs
- Using EventFiringWebDriver to write log entries

The Page Object Model pattern

- Overview on POM
- Do's and Don'ts
- Nested Page Objects
- Reusable Page Objects
- Page Objects and Inheritance
- Limitations of Page Objects and alternatives

Selenium Grid

- Selenium Grid Architecture
- When to use Selenium Grid and alternative solutions
- Selenium Grid cloud providers

Other best practices (if time allows)

- Exception handling tips
- Fitting the tests to the SUT’s architecture