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

Waits

Contents

Overview

If we try to interact with a `WebElement` in Selenium which isn't on the page, it will return a `NoSuchElementException`. To remedy this we can tell the Selenium driver to wait until an element appears.

Having Selenium to wait is much more efficient then using something like `Thread.sleep()` because this will always wait for however many milliseconds you hard-code into it - even if the element you need appears well within time.

With a wait, Selenium will wait *up to ten seconds*, but if the element you need appears before then, it will continue. This saves a huge amount of time in testing, especially if you were originally running multiple tests with `Thread.sleep()`!

A note on wait timings

It is important to note that while setting the timings for your waits to the full ten seconds are more likely to lead to successful tests, you must still bear in mind the short attention-span of your users - if a Web site takes a long time to load, users would grow impatient, and thus user acceptance testing would fail instead!

Therefore, while Selenium can add extra benefit by automating whether items will load on a screen in a timely manner, you are still responsible for the timings of your test suites.

Wait types

There are three types of wait:

- Implicit Wait
- Explicit Wait
- Fluent Wait

Implicit Wait

An *implicit wait* is used when we want to set a global wait for all elements. By default the implicit wait is set to `0`. If an element is not found on the Web page DOM, it will always wait for the time specified in the implicit wait.

Explicit Wait

An *explicit wait* is used on an element-by-element basis. Before interacting with any element in the DOM it is possible to have a wait applied for that individual element. This is useful when an element in particular is expected to

Express-Testing

Networking

Security

Cloud Fundamentals

AWS Foundations

AWS Intermediate

Linux

DevOps

Jenkins Introduction

Jenkins Pipeline

Markdown

IDE Cheatsheet

have delays before it is visible e.g. after loading has taken place.

Fluent Wait

A *fluent wait* is similar to an explicit wait, with the exception that instead of constantly checking the DOM like the explicit wait would, a fluent wait will check to see if the element is available, and, if not, it will wait a certain amount of time before checking again. This is very useful for applications which will read information in and then gradually build the front-end, such as `ajax`.

Tutorial

Let's take a quick look at the syntax for each wait type:

Implicit Waits

Implicit waits are applied to the driver itself:

▶ Implicit Wait Example

Explicit Waits

Explicit waits are applied to an individual element:

▶ Explicit Wait Example

Fluent Waits

Fluent waits are applied to an individual element and are polled during the waiting time:

▶ Fluent Wait Example

Exercises

Play around with each type of wait on the automation practice Web site.

What do you notice about the different types of wait?