

Python Testing with Selenium

Learn to Implement Different
Testing Techniques Using the
Selenium WebDriver

—
Sujoy Raghavendra

Apress®

Python Testing with Selenium

**Learn to Implement Different
Testing Techniques Using the
Selenium WebDriver**

Sujay Raghavendra

Apress®

Python Testing with Selenium

Sujay Raghavendra
Dharwad, Karnataka, India

ISBN-13 (pbk): 978-1-4842-6248-1
<https://doi.org/10.1007/978-1-4842-6249-8>

ISBN-13 (electronic): 978-1-4842-6249-8

Copyright © Sujay Raghavendra 2021, corrected publication 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

Trademarked names, logos, and images may appear in this book. Rather than use a trademark symbol with every occurrence of a trademarked name, logo, or image we use the names, logos, and images only in an editorial fashion and to the benefit of the trademark owner, with no intention of infringement of the trademark.

The use in this publication of trade names, trademarks, service marks, and similar terms, even if they are not identified as such, is not to be taken as an expression of opinion as to whether or not they are subject to proprietary rights.

While the advice and information in this book are believed to be true and accurate at the date of publication, neither the authors nor the editors nor the publisher can accept any legal responsibility for any errors or omissions that may be made. The publisher makes no warranty, express or implied, with respect to the material contained herein.

Managing Director, Apress Media LLC: Welmoed Spahr
Acquisitions Editor: Celestin Suresh John
Development Editor: James Markham
Coordinating Editor: Aditee Mirashi

Cover designed by eStudioCalamar

Cover image designed by Freepik (www.freepik.com)

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, Suite 4600, New York, NY 10004-1562, USA. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit www.springeronline.com. Apress Media, LLC is a California LLC and the sole member (owner) is Springer Science + Business Media Finance Inc (SSBM Finance Inc). SSBM Finance Inc is a **Delaware** corporation.

For information on translations, please e-mail booktranslations@springernature.com; for reprint, paperback, or audio rights, please e-mail bookpermissions@springernature.com.

Apress titles may be purchased in bulk for academic, corporate, or promotional use. eBook versions and licenses are also available for most titles. For more information, reference our Print and eBook Bulk Sales web page at <http://www.apress.com/bulk-sales>.

Any source code or other supplementary material referenced by the author in this book is available to readers on GitHub via the book's product page, located at www.apress.com/978-1-4842-6248-1. For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

To my Baba
Late Raghavendra A S

Table of Contents

About the Author	xv
About the Technical Reviewer	xvii
Acknowledgments	xix
Introduction	xxi
Chapter 1: Introduction to Selenium.....	1
What Is Selenium?	1
Selenium Tools and Versions.....	1
Selenium RC (Remote Control)	3
Selenium IDE	3
Selenium Grid	3
Selenium WebDriver	4
Selenium 2.0	4
Selenium 3.0	5
Selenium WebDriver Architecture	5
Client Library	5
JSON Wire Protocol.....	6
Web Drivers	6
Web Browsers	6
Why Selenium?	6
Open Source	7
Platforms	7
Language Support	7
Browser	7
Reuse.....	7
Easy Implementation	8

TABLE OF CONTENTS

- Flexible 8
- Hardware Resources 8
- Simulation 8
- Other Testing Tools 8
- Integrating Python with Selenium..... 10
- Summary..... 14
- Chapter 2: Getting Started 15**
- Installing Python 15
- Installing Selenium 16
- Installing Drivers 16
- Browser Commands..... 17
- Opening a Web Browser 17
- Mozilla Firefox 18
- Google Chrome 18
- Microsoft Edge 18
- Internet Explorer 18
- Closing a Browser 19
- Quit Browser..... 19
- Opening a Web Page..... 20
- Setting the Browser Size 21
- Setting the Browser Position 22
- Setting the Size Using Coordinates 23
- Getting the Browser Position 23
- Getting the Window Size..... 24
- Navigation Commands..... 24
- Running a Python Test Case..... 26
- Summary..... 26
- Chapter 3: Mouse and Keyboard Actions..... 27**
- Action Chains 27
- Mouse 28
- Click..... 28

Click and Hold.....	28
Context Click.....	29
Double Click.....	29
Move to an Element.....	30
Move Offset	31
Drag and Drop	32
Drag and Drop By	34
Release.....	35
Keyboard Actions	35
Key Down.....	36
Key Up	36
Send Keys.....	36
Send Keys to Element.....	37
Pause.....	37
Reset	37
Summary.....	37
Chapter 4: Web Elements.....	39
Elements/Web Elements	39
Web Locators	40
Locating Elements	40
ID Locator	40
Name Locator	41
XPath Locator	43
CSS Selectors.....	56
Link Locator.....	61
Partial Link Locator.....	62
Tag Name.....	63
Class Name.....	64
Locating Multiple Web Elements	65

TABLE OF CONTENTS

- Locator Issues..... 66
 - Attribute Change..... 67
 - No Web Element 67
 - Web Element Not Visible..... 67
 - Test Case Mismatch 68
 - iframe Web Element 68
- Summary..... 68
- Chapter 5: Navigation 69**
 - Hyperlinks 69
 - Testing Hyperlinks..... 70
 - Hyperlink by ID 71
 - Hyperlink by Text..... 71
 - Hyperlink by Partial Link 71
 - Hyperlink by XPath..... 71
 - Nth Hyperlink 71
 - Return All Hyperlinks 72
 - Check for a Valid Hyperlink 72
 - Check for Broken Images..... 73
 - Data Attributes Hyperlink 74
 - Summary..... 74
- Chapter 6: Buttons..... 75**
 - Default Button 75
 - Select 76
 - Submit/Form Button..... 78
 - Select by Visible Text 79
 - Image as a Button 79
 - Assert for Button 80
 - Radio Buttons..... 81
 - Select Radio Buttons 82
 - Unselect/Clear Selection 83

Assert for Radio Button	84
Checkbox	85
Select/Check	86
Assert If Checkbox.....	87
Assert Checkbox If Checked or Not	87
Clear/Unselect/Deselect Checkbox.....	87
Select List	88
Getting All the Options	90
Select	90
Unselect/Clear/Deselect.....	92
Deselect/Clear One Selected	92
Deselect by Visible Text	92
Deselect by Value	92
Deselect by Index	93
Multiple Select List	93
Verify	94
Deselect/Clear All Selected	95
Summary.....	96
Chapter 7: Frames and Textboxes	97
Iframe.....	97
Switching to Iframe.....	99
Switching Using ID	99
Switching Using Name	99
Switching Using Index.....	99
Switching as an Element.....	101
Switching to a Main Iframe.....	102
Default Content.....	102
Parent Frame.....	102
Frames with Waits	102
Textboxes	103
Single-line Textbox	103

TABLE OF CONTENTS

- Multiline Textbox..... 104
- Inserting Values 105
- Getting Values from Textbox and Textarea..... 105
- Summary..... 106
- Chapter 8: Assertions 107**
- The Need for Assertion..... 107
- Basic Asserts 108
 - assertTrue..... 108
 - assertFalse 108
 - assertIs..... 109
 - assertIsNot 109
 - assertIsNone..... 109
 - assertIsNotNone 110
 - assertIsInstance 110
 - assertNotIsInstance..... 110
- Compare Assert..... 110
 - assertEqual..... 111
 - assertNotEqual 111
 - assertGreater 112
 - assertGreaterEqual 112
 - assertLess 112
 - assertLessEqual 112
- Collection Assert 113
 - Lists 113
 - Tuples 113
 - Sets 114
 - Dictionary 114
 - assertIn..... 115
 - assertNotIn 115
- Summary..... 116

Chapter 9: Exceptions	117
What Is an Exception?.....	117
Why Use Exceptions?.....	117
Exceptions in Selenium.....	118
ConnectionClosedException	118
ElementClickInterceptedException.....	118
ElementNotInteractableException	118
ElementNotSelectableException.....	118
ElementNotVisibleException	118
ErrorInResponseException	119
ErrorHandler.UnknownServerException.....	119
ImeActivationFailedException.....	119
ImeNotAvailableException	119
InsecureCertificateException.....	119
InvalidArgumentException	120
InvalidCookieDomainException	120
InvalidCoordinateException	120
InvalidElementStateException	120
InvalidSelectorException	120
InvalidSessionIdException	120
InvalidSwitchToTargetException	121
JavascriptException	121
JsonException	121
MoveTargetOutOfBoundsException	121
NoAlertPresentException	121
NoSuchAttributeException.....	121
NoSuchCookieException.....	122
NoSuchElementException.....	122
NoSuchFrameException	122
NoSuchWindowException.....	122
NoSuchContextException	123

TABLE OF CONTENTS

- ScreenshotException 123
- StaleElementReferenceException 123
- TimeoutException 123
- UnableToSetCookieException 123
- UnexpectedAlertPresentException 123
- UnexpectedTagNameException 124
- UnknownMethodException 124
- WebDriverException 124
- Exception Handling 124
 - Timeout Exception 125
 - Element Not Found 125
 - Stale Element 126
- Summary..... 127
- Chapter 10: Waits 129**
 - Why the Need for Waits?..... 130
 - Types of Waits 130
 - Implicit..... 131
 - Explicit..... 132
 - Commonly Used ExpectedConditions in the Python Class 134
 - Differences Between Implicit and Explicit 139
 - Fluent 140
 - Summary..... 142
- Chapter 11: Page Objects 143**
 - Page Objects Overview 143
 - Page Object Model (POM) Overview..... 144
 - The Need for the Page Objects Model..... 144
 - Creating Page Objects..... 145
 - Test.py 145
 - Page.py 146

Elements.py	147
Locators.py	148
Advantages of Page Objects Model.....	149
Limitations of the Page Object Model	150
Summary.....	150
Chapter 12: Using Test Cases with a Screenshot	151
Test Outcomes	151
Pass/OK.....	151
Fail.....	152
Error.....	152
Test Case 1: Taking a Screenshot.....	152
save_screenshot ('name_of_file')	152
get_screenshot_as_file ('name_of_file').....	153
get_screenshot_as_png()	153
Testing Frameworks.....	154
The Unittest Framework	155
Test Case 2: Unittest	156
setUp().....	156
tearDown()	157
setUpClass	158
tearDownClass	158
setUpModuleandtearDownModule	159
Test Execution Order	161
Testing Tool Comparisons	162
Summary.....	163
Correction to: Introduction to Selenium	C1
Index.....	165

About the Author



Sujay Raghavendra works primarily in data science, machine/deep learning, and artificial intelligence. He is currently the Executive Director of Raghavendra Training & Consultancy (RTC), a startup company based in Dharwad, Karnataka, India. He co-founded RTC with Sumedh Raghavendra, his brother, in 2014. He plans and evaluates new technology projects in research and product development. His projects include analyzing a pap smear filter for microscopic medical images, thermal heat sensing in hospitals, OCR for handwritten characters, satellite

image analysis, and network automation for maps, forecasting models, text analytics, predictions, and more.

Raghavendra is also a consultant on building research centers for technical universities and colleges. His recent interests include automating test cases through machine learning. He has published numerous research articles in international journals and has been on reviewer committees for various journals and conferences.

About the Technical Reviewer



Supreeth Chandrashekhar is a seasoned software technology professional associated with Philips Healthcare, Bangalore. He has more than eight years of architect experience in building/running highly robust, massively scalable, and extremely secure systems, as well as setting up and managing mid-sized tech teams. He is extremely passionate about building great products that exceed customer expectations.

Acknowledgments

The person behind who I am today is my mother, Mrs. Indumati Raghavendra.

Introduction

This book focuses on how to implement testing techniques using Selenium WebDriver with the Python programming language. This quick reference provides simple, functional test cases with a syntax-based approach for Selenium WebDriver.

You'll begin by reviewing the basics of Selenium WebDriver and its architectural design history. Next, you move on to the configuration and installation of the Selenium library in various web browsers, including the basic commands needed to start test scripts. You'll review keyboard and mouse action commands for testing user interactions on a web page and see how hyperlinks are tested.

The book examines various web elements using the eight different locators provided by Selenium to help you choose the one best suited to your needs. All Python scripts are real ready-to-test examples that are explained thoroughly in problem statements. You'll use different Python design patterns to automate test scripts that can be incorporated with Selenium.

Python Testing with Selenium teaches the expertise to write your own test cases.

CHAPTER 1

Introduction to Selenium

Before Selenium, testing the functionality of a web application was done manually, which took many hours. The testing tended to rely on different scenarios. Each scenario was considered a test case to enact the behavior of the web app before its implementation. These test cases were deployed on various browsers to affirm any issues in the source code.

It requires a dedicated team of testers to check all test cases. Accuracy and time are major constraints in web development, which has led to automated test cases that can be used in different web applications without changing the source code. Selenium was developed to automate test cases.

This first chapter of the book offers a complete overview of Selenium and its core architectural design. The subtopics explain using Selenium and compares it to other testing tools in the domain. Later in the chapter, integrating Python with Selenium is explained. Let's start with a brief history and description of the Selenium tool and the reasons to use it.

What Is Selenium?

Selenium came into existence in 2004 at ThoughtWorks to test a web application named Time and Expenses by Jason Huggins. The tool was developed to test the front-end behavior of an application in various browsers. The tool was popular and open source. The increase in demand for automated testing led to the development of several versions of Selenium over the years, which are discussed next.

Selenium Tools and Versions

ThoughtWorks has released four major versions of Selenium to test web applications. Figure 1-1 shows each version and its release year.

The original version of this chapter was revised. A correction to this chapter is available at https://doi.org/10.1007/978-1-4842-6249-8_13

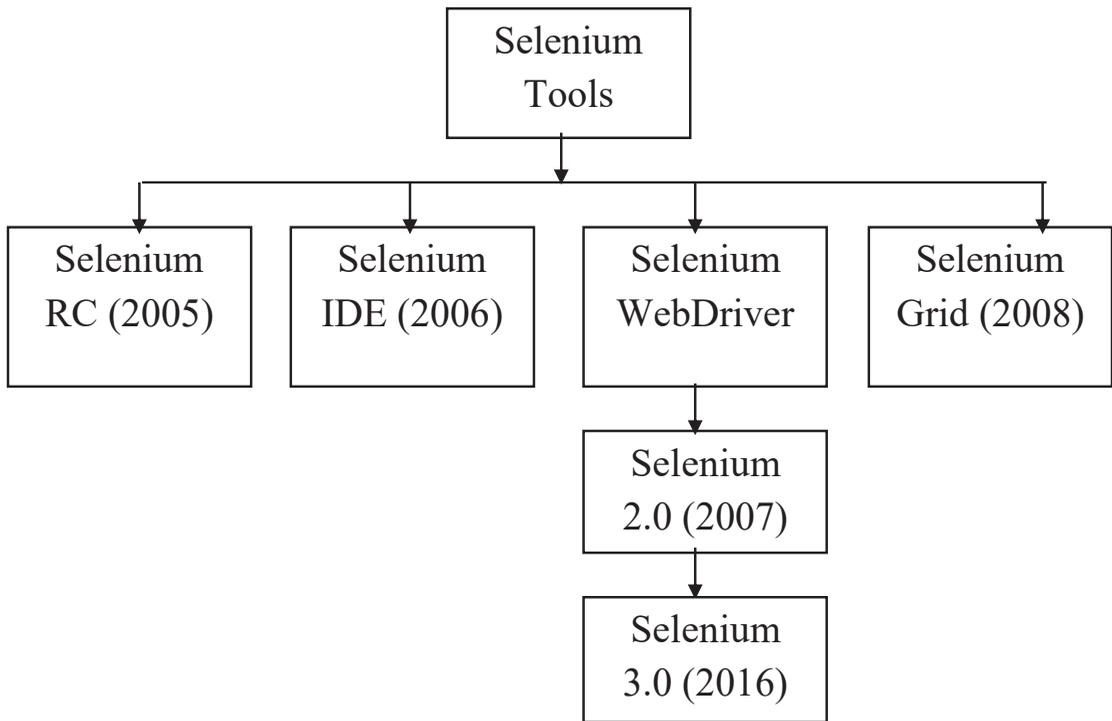


Figure 1-1. *Selenium suite*

Any one (or more) of these Selenium tools can be used by an organization; the choice depends on the test environment’s requirements. The first tool developed by ThoughtWorks was Selenium Core in 2004. It allowed the tester to write their own code/script to automate front-end interactions like keyboard or mouse activities. These activities were similar to a user interacting with the application.

In web application security, there is a policy that grants a test script permission to access data in web pages from the same origin; this policy is called a *same-host* policy. A same-host policy only allows a test case to access pages within the same domain. For example, a test script can access multiple pages within www.apress.com, such as www.apress.com/in/python and www.apress.com/in/about, because of the same-host policy; however, this policy does not allow access to pages from different sites, such as <https://google.com> or <https://wikipedia.org>.

Due to the same-host policy, access to code elements is denied or blocked when using external scripts. To avoid this complication, Huggins and Paul Hammant developed a server component that enables you to test a web app with a test script that makes the browser believe that both are from the same source. This core Selenium was eventually known as Driven Selenium or Selenium B.