



# Practical WebDriverIO

Learn to Automate Effectively  
Using WebDriverIO APIs

---

Shashank Shukla

Apress®

# **Practical WebDriverIO**

**Learn to Automate Effectively  
Using WebDriverIO APIs**

**Shashank Shukla**

**Apress®**

## ***Practical WebDriverIO***

Shashank Shukla  
Mandla, India

ISBN-13 (pbk): 978-1-4842-6660-1  
<https://doi.org/10.1007/978-1-4842-6661-8>

ISBN-13 (electronic): 978-1-4842-6661-8

Copyright © 2021 by Shashank Shukla

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: Louise Corrigan  
Development Editor: James Markham  
Coordinating Editor: Nancy Chen

Cover designed by eStudioCalamar

Cover image by Jimmy Chan from Pexels

Distributed to the book trade worldwide by Springer Science+Business Media New York, 1 New York Plaza, New York, NY 10004. Phone 1-800-SPRINGER, fax (201) 348-4505, e-mail orders-ny@springer-sbm.com, or visit [www.springeronline.com](http://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](mailto:booktranslations@springernature.com); for reprint, paperback, or audio rights, please e-mail [bookpermissions@springernature.com](mailto: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/9781484266601](http://www.apress.com/9781484266601). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*This book is dedicated to my mom and dad  
for always loving and supporting me.*

# Table of Contents

**About the Author ..... xxv**

**About the Technical Reviewer ..... xxvii**

**Acknowledgments ..... xxix**

**Introduction ..... xxxi**

**Chapter 1: Getting Started ..... 1**

    Introduction..... 1

    Installation ..... 2

        Prerequisites ..... 3

        Installation process ..... 4

        Additional Information ..... 13

    Summary..... 14

**Chapter 2: Web Locators ..... 15**

    ID..... 20

        Syntax..... 21

        Output..... 21

    Class ..... 21

        Syntax..... 21

        Output..... 22

    Notes ..... 22

TABLE OF CONTENTS

**Name Attribute** ..... 25

    Syntax..... 26

    Output..... 26

    Notes ..... 26

**Tag Name** ..... 26

    Syntax..... 27

    Output..... 27

**Link Text**..... 28

    Syntax..... 28

    Output..... 28

    Note ..... 28

**Partial Link Text**..... 29

    Syntax..... 29

    Output..... 29

    Note ..... 29

**Elements with Certain Text** ..... 30

    Syntax..... 30

**CSS Query Selector** ..... 31

    Syntax..... 31

    Output..... 32

    Note ..... 32

**XPath**..... 32

    Syntax..... 33

    Output..... 33

    Note ..... 33

**JS Function** ..... 34

    Syntax..... 34

    Output..... 35

Chain Selectors .....	35
Syntax.....	35
Output.....	35
Note .....	36
React Selectors .....	36
Syntax.....	37
Notes .....	37
Custom Selectors .....	38
Syntax.....	38
Output.....	39
Notes .....	39
Summary.....	39
<b>Chapter 3: Browser APIs .....</b>	<b>41</b>
Debugging.....	42
Syntax.....	43
Output.....	43
Notes .....	44
Loading URL and Basic Authentication .....	44
Output.....	45
Notes .....	46
Getting a Count of the Elements Returned from an Array of Elements .....	46
Syntax.....	46
Output.....	47
Notes .....	47
Getting the First Element Returned from an Array of Elements .....	47
Syntax.....	48
Output.....	48
Notes .....	48

## TABLE OF CONTENTS

Getting the Text of an Element .....	49
Syntax.....	49
Output.....	50
Notes .....	50
Getting the Text of any Element Returned from an Array of Elements .....	50
Syntax.....	50
Output.....	51
Notes .....	51
Getting the Last Element Returned from an Array of Elements .....	51
Syntax.....	51
Output.....	52
Iterating All Elements .....	52
Syntax.....	52
Output.....	53
Notes .....	54
Getting All the Links on a Page .....	54
Syntax.....	54
Code Snippet .....	54
Output.....	55
Notes .....	55
Map Function .....	55
Syntax.....	55
Output.....	56
Notes .....	56
Scrolling an Element into View .....	58
Syntax.....	58
Output.....	58
Notes .....	59



Click an Element .....	59
Syntax.....	60
Output.....	60
Notes .....	60
Double-Click an Element.....	61
Syntax.....	62
Output.....	62
Notes .....	63
Right-Clicking an Element .....	63
Syntax.....	63
Output.....	63
Notes .....	64
Sending Text to an Input Field.....	64
Syntax.....	64
Output.....	64
Notes .....	65
Sending a Text to an Input Field via addValue.....	65
Syntax.....	65
Output.....	66
Sending Keyboard Keys to an Element .....	66
Syntax.....	66
Output.....	67
Notes .....	67
Getting the Value of an Element.....	67
Syntax.....	67
Output.....	68
Notes .....	68

TABLE OF CONTENTS

Clearing the Text Inside an Input Field .....68

    Syntax.....68

    Output.....69

Hovering the Mouse on an Element .....69

    Syntax.....69

    Output.....70

    Notes .....70

Navigating to a New URL in a Browser .....71

    Syntax.....71

    Output.....71

    Notes .....71

Navigating Back in a Browser.....72

    Syntax.....72

    Output.....72

    Notes .....73

Navigating Forward in a Browser.....73

    Syntax.....73

    Output.....74

Refreshing a Web Page .....74

    Syntax.....74

    Output.....75

    Notes .....75

Restarting a Browser .....75

    Syntax.....76

    Output.....76

Getting and Setting Window Size and Position .....	76
Syntax.....	77
Output.....	79
Notes .....	79
Getting Element Size.....	79
Syntax.....	79
Output.....	80
Notes .....	80
Maximizing the Browser .....	80
Syntax.....	81
Output.....	82
Notes .....	83
Minimizing the Browser .....	83
Syntax.....	83
Output.....	84
Notes .....	84
Browser Fullscreen Mode .....	84
Syntax.....	84
Output.....	85
Opening a New Window .....	85
Syntax.....	85
Output.....	86
Notes .....	86
Getting the URL of the Current Page .....	86
Syntax.....	86
Output.....	87

TABLE OF CONTENTS

Sending JavaScript to do a Task: Vanilla JS Code.....87

    Syntax.....87

    Output.....88

    Notes .....88

Sending JavaScript to do a Task: Handling Datepicker .....88

    Syntax.....89

    Output.....89

    Notes .....89

Taking a Full-Page Screenshot .....90

    Syntax.....90

    Output.....90

    Notes .....90

Switching Between Windows.....90

    Syntax.....91

    Output.....91

    Notes .....91

Switching Between Frames .....92

    Syntax.....92

    Output.....93

    Notes .....94

Closing the Page .....94

    Syntax.....94

    Output.....95

    Notes .....95

Closing the Browser.....96

    Syntax.....96

    Output.....96

    Notes .....96

Alerts: Accepting an Alert.....	97
Syntax.....	97
Output.....	98
Notes .....	98
Alerts: Dismissing an Alert.....	98
Syntax.....	98
Output.....	99
Notes .....	99
Alerts: Sending a Message to an Alert .....	100
Syntax.....	100
Output.....	101
Notes .....	101
Alerts: Reading an Alert Message .....	101
Syntax.....	102
Output.....	102
Notes .....	102
Selecting from a Drop-Down.....	102
Syntax.....	103
Output.....	104
Drag and Drop.....	105
Syntax.....	105
Output.....	105
Notes .....	106
Uploading a File .....	107
Output.....	108
Notes .....	108
Submitting a Form .....	108
Notes .....	108

## TABLE OF CONTENTS

Display Cookies.....	108
Syntax.....	109
Output.....	109
Notes .....	110
Delete Cookies .....	110
Syntax.....	110
Output.....	110
Notes .....	111
Set Cookies .....	111
Syntax.....	111
Output.....	112
Notes .....	112
Geolocations .....	113
Syntax.....	114
Output.....	115
Notes .....	115
Summary.....	116
<b>Chapter 4: Element APIs and WebDriverIO Assertions.....</b>	<b>117</b>
Is the Element Present?: <code>isExisting()</code> .....	118
Syntax.....	118
Output.....	119
Notes .....	120
Is the Element Present?: <code>toExist()</code> .....	120
Syntax.....	120
Output.....	121
Notes .....	121

Is the Element Present?: toBePresent() .....	122
Syntax.....	122
Output.....	122
Note .....	123
Is the Element Present in DOM?: ToBeExisting().....	123
Syntax.....	123
Output.....	124
Notes .....	124
Is the Element Enabled?: isEnabled() .....	124
Syntax.....	125
Output.....	126
Notes .....	126
Is the Element Enabled?: toBeEnabled().....	126
Syntax.....	127
Output.....	127
Notes .....	127
Is the Element Disabled?: toBeDisabled().....	128
Syntax.....	128
Output.....	129
Notes .....	129
Is the Element Visible?: isDisplayed() .....	129
Syntax.....	130
Output.....	130
Notes .....	131
Is the Element Visible?: toBeDisplayed().....	132
Syntax.....	132
Output.....	132
Notes .....	133

TABLE OF CONTENTS

Is the Element Visible?: toBeVisible()..... 133

    Syntax..... 133

    Output..... 133

Is the Element Visible on the screen?: toBeDisplayedInViewport()..... 134

    Syntax..... 134

    Output..... 134

    Notes ..... 135

Is the Element Visible on the Screen?: toBeVisibleInViewport() ..... 135

    Syntax..... 135

    Output..... 136

Is the Element Selected?: isSelected() ..... 136

    Syntax..... 137

    Output..... 138

    Notes ..... 138

Is the Element Selected?: toBeSelected()..... 138

    Syntax..... 138

    Output..... 139

Is the Element Selected?: toBeChecked()..... 139

    Syntax..... 139

    Output..... 140

Is the Element Clickable?: isClickable() ..... 141

Syntax ..... 141

    Output..... 142

    Notes ..... 142

Is the Element Clickable?: toBeClickable() ..... 142

    Syntax..... 142

    Output..... 143

    Notes ..... 143



To Sum It Up.....	143
Summary.....	146
<b>Chapter 5: Additional WebDriverIO Methods .....</b>	<b>147</b>
Is the Element Focused?: isFocused .....	148
Syntax.....	148
Output.....	149
Note .....	149
Is the Element Focused?: toBeFocused .....	149
Syntax.....	150
Output.....	150
Note .....	150
Does the Element Have a Specific Attribute?: toHaveAttribute .....	151
Syntax.....	151
Output.....	152
Notes .....	152
Does the Element Have a Specific Attribute?: toHaveAttr .....	152
Syntax.....	153
Output.....	153
Does the Element Contain a Specific Text Attribute?: toHaveAttributeContaining .....	154
Syntax.....	154
Output.....	155
Notes .....	155
Does the Element Have a Specific Class?: toHaveClass .....	155
Syntax.....	156
Output.....	156
Notes .....	157

TABLE OF CONTENTS

Does the Element Contain Specific Text in Class?: toHaveClassContaining..... 157

    Syntax..... 157

    Output..... 158

    Notes ..... 159

Does the Element Have a Specific Property?: toHaveElementProperty ..... 159

    Syntax..... 160

    Output..... 160

    Notes ..... 160

Does the Element Have a Specific Value?: toHaveValue ..... 161

    Syntax..... 161

    Output..... 161

    Notes ..... 162

Does the Element Have a Specific href?: toHaveHref ..... 162

    Syntax..... 162

    Output..... 162

    Notes ..... 163

Does the Element Contain a Specific Text in the href?:  
toHaveHrefContaining ..... 163

    Syntax..... 163

    Output..... 164

Does the Element Have a Specific Link?: toHaveLink ..... 164

    Syntax..... 164

    Output..... 165

    Notes ..... 165

Does the Element Contain a Specific Text in the Link?:  
toHaveLinkContaining ..... 165

    Syntax..... 165

    Output..... 166

    Notes ..... 166

Does the Element Have a Specific Text?: <code>toHaveText</code> .....	166
Syntax.....	166
Output.....	167
Notes .....	167
Does the Element Contain a Specific Text?: <code>toHaveTextContaining()</code> .....	167
Syntax.....	168
Output.....	168
Does the Element Have a Specific ID? .....	168
Syntax.....	168
Output.....	169
Element Count.....	169
Syntax.....	170
Output.....	170
Notes .....	170
To Sum It Up.....	171
Summary.....	174
<b>Chapter 6: Other Useful APIs .....</b>	<b>175</b>
Dealing with a Shadow DOM.....	175
Syntax.....	177
Output.....	178
Getting the Page Source .....	181
Syntax.....	181
Output.....	181
Getting an Active Element.....	181
Syntax.....	182
Output.....	182

TABLE OF CONTENTS

Getting the Property of an Element..... 182

    Syntax..... 182

    Output..... 183

Getting the CSS Property of an Element ..... 184

    Syntax..... 184

    Output..... 185

Getting the Tag Name of an Element..... 186

    Syntax..... 186

    Output..... 186

Getting the Location of an Element..... 186

    Syntax..... 187

    Output..... 187

Getting the Size of an Element..... 188

    Syntax..... 188

    Output..... 189

Getting the HTML Build of an Element ..... 189

    Syntax..... 189

    Output..... 190

Summary..... 190

**Chapter 7: Waits ..... 191**

    Hard and Explicit Waits at a Glance ..... 191

    Hard Sleep ..... 192

        Syntax..... 192

        Output..... 192

    Wait for an Element to Be Clickable..... 193

        Syntax..... 193

        Output..... 194

Wait for an Element to Be Displayed.....	194
Syntax.....	195
Output.....	196
Wait for an Element to Be Enabled .....	197
Syntax.....	197
Output.....	198
Wait for an Element to Exist.....	198
Syntax.....	199
Output.....	200
Wait Until.....	200
Syntax.....	200
Output.....	201
Summary.....	202
<b>Chapter 8: Timeouts .....</b>	<b>203</b>
Setting and Getting Various Timeouts .....	204
Output.....	204
Session Implicit Wait Timeout.....	205
Session Page Load Timeout .....	206
Output .....	207
Session Script Timeout .....	207
WebdriverIO-related Timeouts: waitForTimeout .....	208
Output .....	209
Framework-related Timeouts.....	209
Summary.....	211

TABLE OF CONTENTS

**Chapter 9: Framework Options and Design Patterns.....213**

    Introduction to Frameworks..... 213

    WebdriverIO with Cucumber ..... 214

    WebdriverIO with TypeScript..... 217

    WebdriverIO with Jasmine ..... 218

        WebdriverIO with Mocha ..... 218

    Design Pattern Introduction ..... 219

        base.js ..... 221

        login.page.js ..... 222

        landing.page.js ..... 224

        example.e2e.js..... 225

    Summary..... 226

**Chapter 10: Assertions .....229**

    Determining If Strings Match by Value..... 230

    Determining If Strings Match by Value and Type..... 230

    Determining If a Value Is Truthy ..... 231

    Determining If a Value Is Falsy..... 231

    Determining If a Value Is Equal (==) ..... 231

    Determining If a Value and Type Both Are Equal (===)..... 232

    Determining If a Value Is Not Equal (!==)..... 233

    Determining If a Value and Type Are Not Equal (!==)..... 233

    Determining If a Value Is Higher Than Expected ..... 234

    Determining If a Value Is Lower ..... 234

    Determining If Expected Is True ..... 235

    Determining If Expected Is False ..... 235

    Determining If Expected Result Is an Array..... 235

    Determining If an Actual Result Is a String ..... 236

Determining If an Array Contains a Value.....	236
Verifying the Length of an Array.....	237
Summary.....	237
<b>Chapter 11: Configuration File.....</b>	<b>239</b>
Runner .....	240
Specs .....	241
Exclude .....	241
LogLevel.....	242
Services .....	242
Reporters .....	244
Capabilities .....	248
Summary.....	253
<b>Chapter 12: Conclusion.....</b>	<b>255</b>
Advantages of WebDriverIO .....	256
Disadvantages of WebDriverIO .....	259
Challenges of Using WebDriverIO .....	260
<b>Index.....</b>	<b>263</b>

# About the Author



**Shashank Shukla** has been working in software testing for more than a decade and is passionate about tools and technologies that can be leveraged to enrich the testing experience and optimize the quality of delivery. This is his first book.



# About the Technical Reviewer



**Kanika Sud** has been working on the Web for more than ten years. Her work spans enterprise CMSes in Java, back-end technologies in the LAMP stack, and the MEAN stack. She has also worked on open source e-commerce CMSes and UX strategy. Solution design remains her favorite job. Market research on mobile apps and plugins led her to experiment with a bootstrapped technology startup called Codnestic Solutions. Find her on LinkedIn at [www.linkedin.com/in/kanikasud](http://www.linkedin.com/in/kanikasud).

# Acknowledgments

I would like to thank all the generous people in the open source community for tirelessly contributing to making WebdriverIO easy to use, and helping others implement smarter testing approaches.

I want to thank my wife, Anuja, who encouraged me to write this book and supported me throughout the writing process, and my son, Riyaarth, for teaching me not to let “the lack of time” hold me back.

I would also like to thank my test manager, Eileen Stevenson, for her leadership and guidance, and also my test team, Anju, Gomathi, Shilpa, Mueez, Sreekanth, Kay and Lucie for always being there to support me.

And lastly, I want to thank my editorial team for their interest and effort, which fine-tuned and shaped this book.

# Introduction

WebdriverIO is probably the coolest NodeJS-based framework gathering traction in the market. It has simplified complex, promise-based testing that stems due to the asynchronous nature of JavaScript in NodeJS-based frameworks. This tool saves time in automating websites and is fun to use. This book attempts to condense WebdriverIO's API documentation with practical, easy-to-understand examples. This book is a one-stop reference guide on almost every desktop API provided by WebdriverIO.

The journey starts with setting up the WebdriverIO test tool. You learn how to install it and its related dependencies and run a demo spec file in Chapter 1. In Chapter 2, you learn the methods to locate elements using various selector strategies provided by WebdriverIO, which are essential in interacting with web elements.

Chapters 3, 4, 5, and 6 explore various WebdriverIO API methods through easy-to-understand examples of automating a variety of user actions on located elements and the web app. You also learn some of WebdriverIO's built-in assertions.

In Chapter 7, you learn the importance of the wait command in automation testing and implement various wait commands. In Chapter 8, you learn about timeouts. After covering enough groundwork, the book touches upon various WebdriverIO framework options.

You learn about the page object model design pattern in Chapter 9. In Chapter 10, you learn about an external assertion library called the Chai Assertion Library. You then learn about WebdriverIO configuration settings and how to integrate a reporter and parallelly execute tests in Chapter 11.

The journey concludes in Chapter 12 by looking at some of the pros and cons of WebdriverIO.

# CHAPTER 1

# Getting Started

If you're interested in browser automation through Node.js and understanding the various methods that can be employed to effectively automate user actions on your web site, this book is for you. WebdriverIO is a fast-growing automation tool that is hard to ignore as Node.js/JavaScript technologies assert their dominance in front-end and back-end web development.

This chapter briefly introduces WebdriverIO. You learn how to install it and quickly get to a point where you can start hands-on work. The chapter also discusses the following.

- Why WebdriverIO is gaining popularity among a new generation of test engineers
- How to run a test in 15 minutes
- Steps to take if the installation fails
- Demo web sites to practice test automation

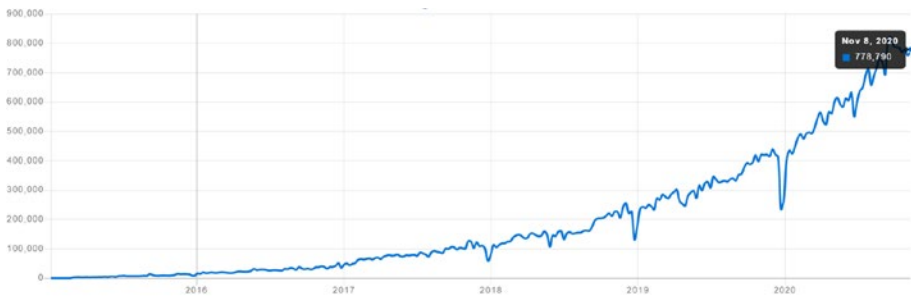
## Introduction

WebdriverIO is an independent and customized implementation of WebDriverJS (Selenium WebDriver) created by Christian Bromann. WebDriverJS is the official JavaScript implementation of Selenium API, packaged as 'selenium-webdriver' in npm, which runs on Node.js. WebdriverIO abstracts the lengthy syntax and complex asynchronous

promise management of JavaScript and presents the user with easy-to-read action commands. It makes every test step synchronous, meaning the user doesn't have to worry about any missed steps in the test code. It is very flexible, allowing users to choose assertion libraries, reporting tools, and various other components of the framework.

WebdriverIO is packaged and installed through npm and runs on Node.js, which is a JavaScript run-time environment that allows you to run JavaScript outside your browser. It can run on macOS, Linux, and Windows.

The principal reason WebdriverIO is gaining traction is that it is open source. If you have experience working with other JavaScript frameworks, you can start using this tool in no time. Figure 1-1 shows its number of downloads over the past few years, depicting its growing popularity.



**Figure 1-1.** The growing popularity of WebdriverIO (from [www.npmtrends.com](http://www.npmtrends.com))

## Installation

The WebdriverIO installation process is very easy. This book explains installation in the Windows 10 operating system. You need an active Internet connection and enough space on your machine to accommodate the setup.

## Prerequisites

This section describes the basic tools required for a barebones WebdriverIO project setup.

### Node.js

The official Node.js download page is at <https://nodejs.org/en/download/>. You can download the LTS (latest stable version) or the current version with the latest Node.js features. It is strongly advised to download the LTS version so that you can avoid unforeseen errors due to experimental features in the package. If you are using Node.js for the first time, it is advisable to download to the location suggested by your operating system. This applies to all the tools I recommend you install in this book. I have used Node.js version 12.16.3 (<https://nodejs.org/en/blog/release/v12.16.3/>) in this book. I recommend you use the same to avoid any errors due to version mismatch.

### VS Code

In the JavaScript universe, VS Code is a widely used, freely available code editor. It provides good integration and support with WebdriverIO for our test development. It is frequently updated with new features to make a developer's life easy. I use version 1.49 in this book. You are free to use the latest available version because it does not impact your execution (<https://code.visualstudio.com/download>).

### Chrome

WebdriverIO provides flexibility to use a wide variety of browsers. Chrome ([www.google.com/chrome](http://www.google.com/chrome)) is used in this book because it is most convenient. Specifically, this book uses Chrome version 87.

Note that if you are using the latest version of Node.js, you might be asked to install Python or JDK as part of the installation process. I recommend installing the versions used in this book.