Extracted from:

The Way of the Web Tester

A Beginner’s Guide to Automating Tests

This PDF file contains pages extracted from *The Way of the Web Tester*, published by the Pragmatic Bookshelf. For more information or to purchase a paperback or PDF copy, please visit http://www.pragprog.com.

Note: This extract contains some colored text (particularly in code listing). This is available only in online versions of the books. The printed versions are black and white. Pagination might vary between the online and printed versions; the content is otherwise identical.

Copyright © 2016 The Pragmatic Programmers, LLC.

All rights reserved.

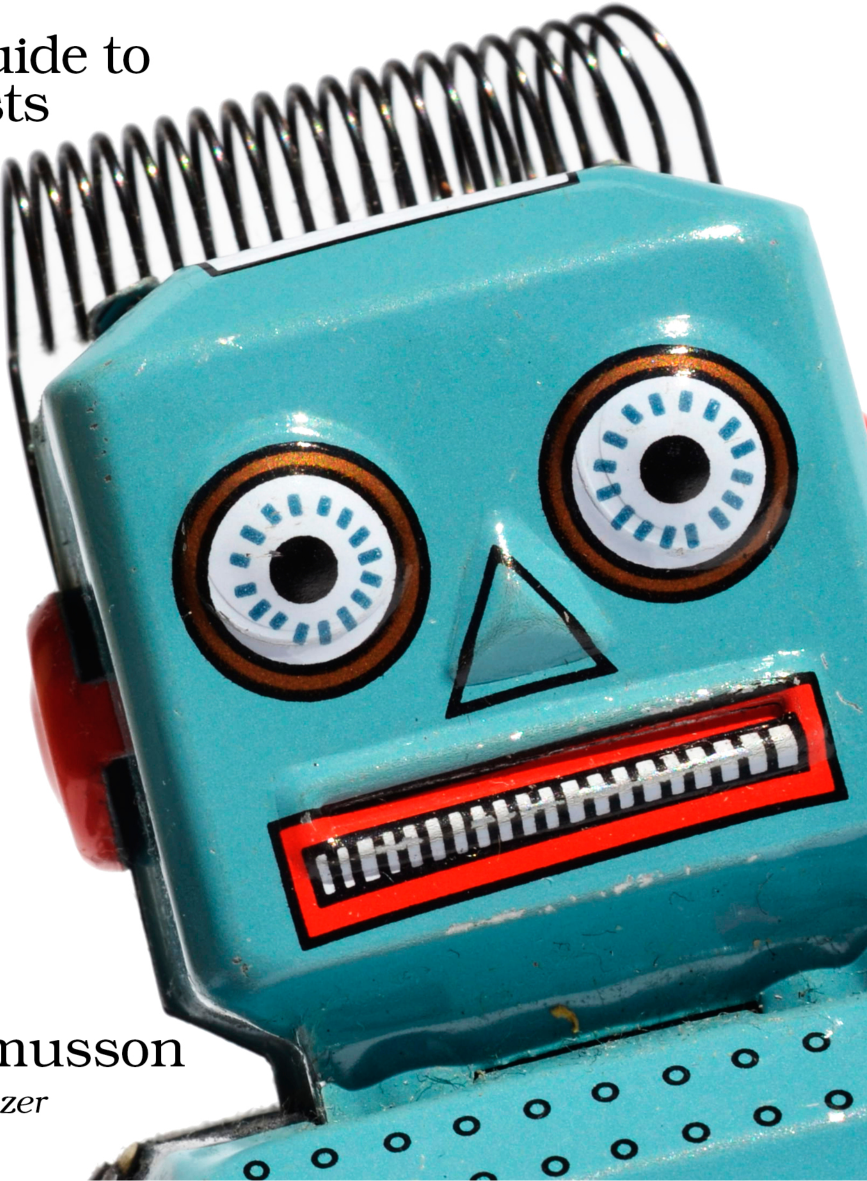
No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

The Pragmatic Bookshelf
Raleigh, North Carolina

The Pragmatic Programmers

The Way of the Web Tester

A Beginner's Guide to Automating Tests



Jonathan Rasmusson

edited by Susannah Pfalzer

The Way of the Web Tester

A Beginner's Guide to Automating Tests

Jonathan Rasmusson

The Pragmatic Bookshelf
Raleigh, North Carolina

Pragmatic Bookshelf

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic books, screencasts, and audio books can help you and your team create better software and have more fun. Visit us at *https://pragprog.com*.

The yellow adhesive note graphic in Chapter 11 is designed by Layerace from Freepik.com.

The team that produced this book includes:

Susannah Davidson Pfalzer (editor)
Potomac Indexing, LLC (index)
Nicole Abramowitz (copyedit)
Gilson Graphics (layout)
Janet Furlow (producer)

For sales, volume licensing, and support, please contact *support@pragprog.com*.

For international rights, please contact *rights@pragprog.com*.

Copyright © 2016 The Pragmatic Programmers, LLC.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior consent of the publisher.

Printed in the United States of America.
ISBN-13: 978-1-68050-183-4
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—September 2016

It’s Good to See You!

This is a book about how to write automated tests for the web. It’s a book for anyone who has ever wanted to learn:

- How automated testing on the web works
- What the different kinds of automated tests are
- And, most importantly, how to get started writing them, even if you have little or no programming background or experience

Be warned—this isn’t your typical deep dive tutorial book. We aren’t going to spend hundreds of pages walking you through how to set up this kind of test framework or that. The technology changes too quickly.

Instead, we are going to focus on the fundamentals. Those things that simply don’t change. These you will be able to take with you and apply to any project —regardless of which automated test framework or platform you choose to use.

And it is good to see you because automated testing is one of the greatest levers we’ve got for scaling the most valuable asset any software project has—you.

You see, you’re kind of a big deal. We need more of you. We need more of your critical thinking. We need more of your creativity. And we need more of your time. And by learning how to write automated tests, that’s really what you are giving yourself and the others on your team. More time.

If you are a traditional software tester who has little or no programming experience, this is the perfect book for getting started. Together we are going to start from the ground up and give you everything you need to create and start writing your very own automated tests today.

If you are a developer, but haven’t thought a ton about how automated testing works, this is your crash course on how to move fast without breaking stuff. That means more time working on fun things, like adding new features, and less time working on the boring stuff, like fixing old bugs.

And if you are a team lead, this is your Rosetta Stone. This book will not only help you bridge the gap between traditional testers and developers, it will give you and your team the time, the language, and the framework to set your automated tests up right, while avoiding much of the duplication and wasted effort that usually comes to teams when they are just starting out.

How to Read This Book

If there are two chapters everyone on your team should read, they are Chapter 1, *The Testing Pyramid*, on page ?, and Chapter 8, *Climbing the Pyramid*, on page ?. These will give a nice overview of how automated testing works, along with the different kinds of tests and where and when to use each one.

For the rest of you, who actually want to know how this stuff works, the book is broken into two parts.

In Part I we go over the basics of how automated testing on the web works.

In Chapter 1, *The Testing Pyramid*, on page ?, we introduce the concept of the testing pyramid: a model all teams can use to coordinate their testing efforts and ensure we are all on the same page.

In Chapter 2, *Smoking User Interface Tests*, on page ?, we introduce the concept of the user interface (UI) test and see how this test helps us test our systems just like a regular user would. And in Chapter 3, *Adding UI Tests to Legacy Systems*, on page ?, we put the theory into practice and see what it takes to add UI tests to an existing legacy system.

In Chapter 4, *Connecting the Dots with Integration Tests*, on page ?, we then dive deeper into the world of the web and see how to test web services directly. We then apply that knowledge in Chapter 5, *Integration Testing RESTful Web Services*, on page ?, where we see how to test what is arguably the most popular kind of web API out there today: RESTful web services.

In Chapter 6, *Covering Our Bases with Unit Tests*, on page ?, we see why unit tests play such an important role in test automation today. And, specifically, we explore how to unit test JavaScript in the browser in Chapter 7, *Unit Testing in the Browser with JavaScript*, on page ?.

And in Chapter 8, *Climbing the Pyramid*, on page ?, we bring it all together and see how the pyramid works in action, starting at the top and then working our way down to the bottom, highlighting a few of the challenges you're likely to meet along the way.

In Part II we move beyond the basics and get into some of the more advanced topics that come with the territory.

In Chapter 9, *Programming 101*, on page ?, we look at the techniques programmers use to write good code, and see how to adapt those strategies for writing good tests.

In Chapter 9, *Programming 101*, on page ?, we look at the techniques testers and developers can use to write better tests, and see what kind of strategies help us write tests that are easier to maintain.

In Chapter 10, *Organizing Tests:*, on page ?, we take a look at all those tests you're going to write, and see how we can go about organizing them in a nice and easy way.

In Chapter 11, *Effective Mocking*, on page ?, we look at some of the pitfalls developers can fall into when relying heavily on mocks and how to avoid them.

And in Chapter 12, *Writing Tests First*, on page ?, we see what writing tests first is like, and how it can help us deal with the complexity and design challenges we face when writing our very first tests.

A Few Conventions

One of the trickier things about writing a book like this is choosing the computer language to do the examples in. I picked Ruby, along with its sidekick Ruby on Rails, and JavaScript for a couple of reasons.

Ruby was chosen because there is a lot to be learned from this community about automated testing and web development in general. That, and Ruby is a fairly easy language to understand and learn—even if you aren't a programmer.

JavaScript is in here because so much of the web is powered by it today. We have an entire chapter dedicated to how it can be used to test functionality in the browser.

But this shouldn't be viewed as an authoritative book on JavaScript or Ruby. These are simply the tools we use today.

Much more important are the fundamentals. Things like HTML, CSS, and HTTP. So we are going to spend more time on these, and less on any particular tool or framework.

The goal here is for you to get so good at the fundamentals that the frameworks and tools they are built upon won't matter.

Fun Bits with Purpose

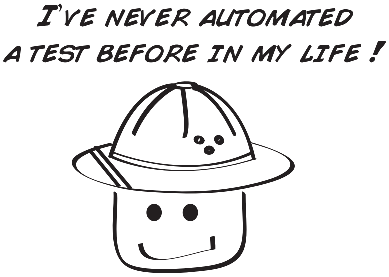
You can't take this stuff too seriously, and it helps if you can approach the material with a sense of humor.

To that end, I've tried to lighten things up with pictures, stories, and anecdotes to share with you along the way.

War stories are real-life, from-the-trenches experiences about some of the successes and failures I and others have had while writing automated tests. You'll know it's a war story when you see the archer.

The Now You Try exercises are there to snap you out of reading and get you into thinking and doing. So keep a pen or pencil handy.

Then there are Tim and Diane—our traditional tester and developer. Tim and Diane don't have a lot of experience when it comes to automated testing. But what they lack in experience, they more than make up for in enthusiasm and questions.



Tim the Tester

Diane the Developer

• Click HERE to purchase this book now. discuss

And when you see a thumb, get ready for a timely tip or piece of advice.

Write good automated tests so you will have more time to do exploratory testing.

Online Resources

You can always reach me, or just send feedback on how the book could be better, at the book's web page, pragprog.com/book/jrtest. Here you can find source code, ask questions, report bugs, and generally discuss all things book related. Make sure you drop by and say hi.

Alright. Let's begin.