Python Unit Test Automation

Practical Techniques for Python Developers and Testers

Ashwin Pajankar

Apress®

Python Unit Test Automation: Practical Techniques for Python Developers and Testers

Ashwin Pajankar
Nashik, Maharashtra, India

ISBN-13 (pbk): 978-1-4842-2676-6 ISBN-13 (electronic): 978-1-4842-2677-3
DOI 10.1007/978-1-4842-2677-3

Library of Congress Control Number: 2017933075

Managing Director: Welmoed Spahr
Lead Editor: Celestin Suresh John
Technical Reviewers: Unmesh Gundecha and Jojo Moolayil
Editorial Board: Steve Anglin, Pramila Balan, Laura Berendson, Aaron Black, Louise Corrigan, Jonathan Gennick, Robert Hutchinson, Celestin Suresh John, Nikhil Karkal, James Markham, Susan McDermott, Matthew Moodie, Natalie Pao, Gwenan Spearing
Coordinating Editor: Sanchita Mandal
Copy Editor: Kezia Endsley
Compositor: SPi Global
Indexer: SPi Global
Artist: SPi Global

Distributed to the book trade worldwide by Springer Science+Business Media New York, 233 Spring Street, 6th Floor, New York, NY 10013. 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 rights@apress.com, or visit www.apress.com.

Apress and friends of ED books 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 Special Bulk Sales–eBook Licensing web page at www.apress.com/bulk-sales.

Any source code or other supplementary materials referenced by the author in this text are available to readers at www.apress.com. For detailed information about how to locate your book's source code, go to www.apress.com/source-code/. Readers can also access source code at SpringerLink in the Supplementary Material section for each chapter.

Printed on acid-free paper

Contents at a Glance

Contents

About the Author

Ashwin Pajankar is a Polymath. He is a Programmer, a Maker, an Author, a Youtuber, and a Science Popularizer. He graduated from IIIT Hyderabad with MTech in Computer Science and Engineering. He has keen intrest in promotion of Science, Technology, Engineering, and Mathematics (STEM) education. He has written 3 books with Packt Publication, 6 books with Leanpub, and has also reviewed four books for Packt Publications. This is his first book with Apress Publication and he's working on couple of more books with Apress.

His personal web site is
www.AshwinPajankar.com
His LinkedIn profile is
https://in.linkedin.com/in/ashwinpajankar

About the Technical Reviewers

Jojo Moolayil is a data scientist and the author of the book, *Smarter Decisions – The Intersection of Internet of Things and Decision Science*. With over four years of industrial experience in data science, decision science and IoT, he has worked with industry leaders on high-impact and critical projects across multiple verticals. He is currently associated with General Electric, the pioneer and leader in data science for Industrial IoT. He lives in Bengaluru—the silicon valley of India.

He was born and raised in Pune, India and graduated from the University of Pune with a major in Information Technology Engineering. He started his career with Mu Sigma Inc., the world's largest pure play analytics provider and worked with the leaders of many Fortune 50 clients. One of the early enthusiasts to venture into IoT analytics, he converged his knowledge from decision science to bring the problem-solving frameworks and his knowledge from data and decision science to IoT analytics.

To cement his foundation in data science for industrial IoT and scale the impact of the problem-solving experiments, he joined a fast-growing IoT analytics startup called Flutura, based in Bangalore and headquartered in the valley. After a short stint with Flutura, Jojo moved on to work with the leaders of Industrial IoT—General Electric—in Bangalore, where he focused on solving decision science problems for industrial IoT use cases. As a part of his role at GE, Jojo also focuses on developing data science and decision science products and platforms for industrial IoT.

Apart from authoring books on decision science and IoT, Jojo has also been a technical reviewer for various books on machine learning, deep learning, and business analytics with Apress. He is an active data science tutor and maintains a blog at http://www.jojomoolayil.com/web/blog/.

His profile:

http://www.jojomoolayil.com/

https://www.linkedin.com/in/jojo62000

He would like to thank his family, friends, and mentors.

Unmesh Gundecha has a master's degree in software engineering and over 15 years of experience in Agile software development, test automation, and technical QA. He is an Agile, open source, and DevOps evangelist with a rich experience in a diverse set of tools and technologies. Presently, he is working as an automation architect for a multinational company in Pune, India. Unmesh also authored the *Selenium Testing Tools Cookbook* and *Learning Selenium Testing Tools with Python* books, published by Packt Publishing.

Acknowledgments

Writing this book is a journey that I am glad I undertook. First, I would like to thank my wife, Kavitha, without whose support this journey would not have been possible. The journey spanned a few months, but the experience will last a life time. I had my wife Kavitha with me onboard this journey and I wish to express my deepest gratitude to her. Without her unwavering support and affection, I couldn't have pulled it off.

I would also like to extend my thanks to my siblings-Savita, Gayatri, and Prasad.

I am grateful to the student and teacher community which, with their continual bombardment of queries, impelled me to learn more, simplify my findings, and place them neatly in the book. This book is for them.

I wish to thank my friends and colleagues—the practitioners from the field—for their good counsel and for filling me in on the latest in the field of test automation.

A special thanks to the technical reviewers—Unmesh and Jojo—for their vigilant review and filling in with their expert opinions.

I have been fortunate to have the support of my team, which sometimes knowingly and at other times unknowingly contributed to the making of the book by lending me their steady support.

I consider myself very fortunate for the editorial assistance provided by Apress. This is my first book with Apress and the collaboration with them has been fabulous. I am thankful to Celestin Suresh John, Senior Manager, editorial acquisitions, Apress and Springer Science and BusinessMedia Company, for giving me this long-desired opportunity to collaborate with Apress. I wish to acknowledge and appreciate Sanchita Mandal, coordinating editor, James Markham, content development editor, Anila Vincent, content development editor, and the team of associates from Apress who adeptly guided me through the entire process of preparation and publication.

Introduction

Why This Book?

I have been using Python for more than 10 years for a variety of stuff. Initially, I used it for GUI applications. Then I quickly moved to the scientific usage as my academic projects demanded it. When I entered professional life, I used it for automation first and then for implementation of alert mechanisms. I have been using Python for the last six years in the various fields of scientific computing, Internet of Things (IoT), and single board computers. I have written plenty of Python code over these years. I always prefer it to bash scripting, which offers limited capabilities to users like me. Over the period of the last 10 years, I worked as a developer, an R&D engineer, a maker, an author, and a QA specialist. I used Python in every single role.

Whenever I write any code, I unit test it thoroughly every time. Earlier I used to unit test all my Python modules in the good old manual way. I used to run all the scripts once and compare the outcome with what's expected. However, I experienced that when your codebase grows larger, it's pretty difficult to perform the activity of testing the scripts manually. Also, all the scripts have to be tested, re-tested, and tested for regression whenever a small part of the codebase changes. I was looking for a way to run all the tests automatically and then I started reading about test automation. It immediately aroused my curiosity and, after a couple of days, I was running my own automated Python tests. After acquainting myself with the philosophy of test automation, I applied the knowledge to automate unit and integration testing to web, mobile, GUI, API, and a variety of other types of applications using programming languages like C++, Python, Java, and PHP.

I wrote this book to share my knowledge and experiences while automating the unit tests in Python 3. I explore different frameworks and plugins in this book. I learned about the tools and techniques explained in this book by spending numerous hours learning, coding, discussing, and actively participating in diverse Internet forums. I have condensed the knowledge to the basics of the unit test automation frameworks in this book. I hope readers will enjoy reading and following the book as much as I enjoyed writing it. This book includes the following:

- An introduction to Python and various IDEs
- Various test automation frameworks for Python 3, including `doctest`, `unittest`, `nose`, `nose2`, and `pytest`
- Coding standards for Python 3 test automation and implementation of test driven development with `pytest` in Python 3

Who This Book Is For

The main audience of this book is Python 3 programmers who want to automate their unit tests. This includes a large and diverse set of people, including developers, test automators, students, researchers, and novice learners. The book is for those who have some knowledge of the Python programming language. The test automation engineers who have already worked with other programming frameworks, such as Java and C++, will find this book immensely useful to learn how test automation is done in Python 3. If you are just beginning with Python 3 programming and want to quickly get into automating the unit tests of your modules and packages, you will find this book helpful.

What this book is not. This book is not a book for learning Python 3 programming and syntax from scratch. It is also not a DIY cookbook for development projects. If your understanding of coding is limited, you will find it difficult to follow this book.

How This Book Is Organized

This book has six chapters. Here is a sneak peek into the chapters of the book:

Chapter 1: This chapter introduces the readers to the history and philosophy of Python. It teaches you how to install Python and how to set up the environment for Python 3 programming. It also briefly explores the new features of Python 3 and introduces the readers to a few popular Python 3 IDEs.

Chapter 2: The aim of this chapter is to quickly get the readers started with unit test automation in Python 3. The chapter revises the understanding of testing concepts and quickly moves into implementing those concepts with `docstring` and `doctest`.

Chapter 3: This chapter serves to introduce `xUnit` and its philosophy to the readers. Then it proceeds to teach the readers how to to implement concepts of `xUnit` with `unittest`, a `xUnit` port for Python.

Chapter 4: This chapter explores the inadequacies in unittest. Then it explores a better unit testing framework, called `nose`. It explains the installation of plugins for `nose` to generate reports. It also discusses `nose2`, which is `nose`'s next-generation version that's under active development.

Chapter 5: This chapter introduces the readers to a modular, easy-to-use, and latest unit test framework for Python, `pytest`. It discusses the drawbacks of `nose` and compares `nose`, `unittest`, and `pytest`.

Chapter 6: This chapter helps the readers understand the coding and filenaming conventions for facilitating easier test discovery across various unit test frameworks in Python. The chapter concludes the book by implementing a test driven development in Python 3 using `pytest`.

How to Get the Most Out of This Book

It is easy to leverage the book to gain the most out of it by religiously abiding by the following:

- Read the chapters thoroughly. Use the chapters hands-on by following the step-by-step instructions stated in the code examples. Do *not* skip any of the code examples. If need be, repeat them a second time or until the concept is firmly etched in your mind.
- Join a Python community or discussion forum.
- Read the online documentation available for various test automation frameworks for Python 3.
- Read test automation, Python 3, migration to Python 3 from Python 2, and test driven development blogs.

Where Next?

I have endeavored to unleash the power of the unit test automation libraries for Python 3 as an aid to the developer and tester community. I recommend you read the book from cover to cover without skipping any of the chapters, text, or code examples.

I wish you well in exploring Python!

A Quick Word for the Instructors' Fraternity

Attention has been paid in arriving at the sequence of chapters and also to the flow of topics within each chapter. This is done particularly with an objective to assist my fellow instructors and academicians in carving out a syllabus from the Table of Contents (ToC) of the book. The complete ToC can complement the syllabus of "Software Testing," if students were introduced to programming their freshman year with the help of Python.

We have ensured that each concept discussed in the book includes adequate hands-on content to enable you to teach better and to provide ample hands-on practice to your students.

Happy Learning!!!
Ashwin Pajankar