

# **Expert Twisted**

## **Event-Driven and Asynchronous Programming with Python**

**Mark Williams**

**Cory Benfield**

**Brian Warner**

**Moshe Zadka**

**Dustin Mitchell**

**Kevin Samuel**

**Pierre Tardy**

**Apress®**

## ***Expert Twisted***

Mark Williams  
Pasadena, CA, USA

Brian Warner  
New York, USA

Dustin Mitchell  
New York, USA

Pierre Tardy  
Toulouse, France

Cory Benfield  
London, UK

Moshe Zadka  
New York, USA

Kevin Samuel  
Nice, France

ISBN-13 (pbk): 978-1-4842-3741-0  
<https://doi.org/10.1007/978-1-4842-3742-7>

ISBN-13 (electronic): 978-1-4842-3742-7

Library of Congress Control Number: 2018965166

Copyright © Mark Williams, Cory Benfield, Brian Warner, Moshe Zadka,  
Dustin Mitchell, Kevin Samuel, Pierre Tardy 2019, corrected publication 2019

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: Jonathan Gennick  
Development Editor: James Markham  
Coordinating Editor: Jill Balzano

Cover image designed by Freepik ([www.freepik.com](http://www.freepik.com))

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](mailto: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 [rights@apress.com](mailto:rights@apress.com), or visit <http://www.apress.com/rights-permissions>.

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/9781484237410](http://www.apress.com/9781484237410). For more detailed information, please visit <http://www.apress.com/source-code>.

Printed on acid-free paper

*Dedicated to AZ, NZ, and TS: Twisted prevails,  
and we're looking forward to the next  
generation of maintainers.*

*—Moshe Zadka*

# Table of Contents

<b>About the Authors</b> .....	<b>xiii</b>
<b>About the Technical Reviewers</b> .....	<b>xv</b>
<b>Acknowledgments</b> .....	<b>xvii</b>
<b>Introduction</b> .....	<b>xix</b>
<b>Part 1: Foundations</b> .....	<b>1</b>
<b>Chapter 1: An Introduction to Event-Driven Programming with Twisted</b> .....	<b>3</b>
A Note About Python Versions .....	4
What Is Event-Driven Programming? .....	4
Multiple Events .....	5
Application(tkinter.Tk()).mainloop() .....	6
Multiplexing and Demultiplexing.....	7
The select Multiplexer.....	9
Its History, Its Siblings, and Its Purpose.....	9
select and Sockets .....	9
The How and Why of Socket Events .....	11
Handling Events.....	12
An Event Loop with select .....	13
Event-Driven Clients and Servers.....	15
Non-blocking I/O .....	18
Knowing When to Stop .....	18
Tracking State.....	19
State Makes Programs Complex .....	23
Managing Complexity with Transports and Protocols .....	23
Reactors: Working with Transports.....	24

TABLE OF CONTENTS

- Transports: Working with Protocols..... 25
  - Playing Ping-Pong with Protocols and Transports ..... 26
  - Clients and Servers with Protocols and Transports ..... 31
  - Twisted and Reactors, Protocols, and Transports ..... 33
- The Value of Event-Driven Programming ..... 33
- Twisted and the Real World..... 36
- Events in Time..... 41
  - Repeated Events with LoopingCall ..... 44
- Event Interfaces with zope.interface..... 46
- Flow Control in Event-Driven Programs ..... 49
- Flow Control in Twisted with Producers and Consumers ..... 50
  - Push Producers ..... 51
  - Consumers ..... 54
  - Pull Producers ..... 57
- Summary..... 57
- Chapter 2: An Introduction to Asynchronous Programming with Twisted ..... 59**
  - Event Handlers and Composition ..... 59
  - What Is Asynchronous Programming? ..... 63
  - Placeholders for Future Values ..... 63
  - Asynchronous Exception Handling..... 66
  - An Introduction to Twisted's Deferred ..... 71
    - Callbacks ..... 71
    - Errbacks and Failures..... 73
    - Composing Deferreds ..... 76
  - Generators and InlineCallbacks ..... 80
    - yield ..... 80
    - send ..... 81
    - throw ..... 84
  - Asynchronous Programming with inlineCallbacks ..... 85

Coroutines in Python .....	88
Coroutines with yield from .....	88
Coroutines async and await .....	90
Awaiting Deferreds .....	95
Coroutines to Deferreds with ensureDeferred .....	97
Multiplexing Deferreds.....	99
Testing Deferreds.....	102
Summary.....	106
<b>Chapter 3: Applications with treq and Klein.....</b>	<b>109</b>
Why Libraries? .....	109
Feed Aggregation.....	110
Introducing treq .....	111
Introducing Klein.....	115
Klein and Deferreds.....	117
Klein Templates with Plating .....	118
A First Draft of Feed Aggregation.....	121
Test-Driven Development with Klein and treq.....	128
Running Test on an Installable Project .....	128
Testing Klein with StubTreq.....	131
Testing treq with Klein.....	140
Logging with twisted.logger.....	143
Running Twisted Applications with twist.....	149
Summary.....	154
<b>Part 2: Projects.....</b>	<b>155</b>
<b>Chapter 4: Twisted in Docker.....</b>	<b>157</b>
Intro to Docker .....	157
Containers .....	157
Container Images .....	158
Runc and Containerd .....	159

TABLE OF CONTENTS

- Client ..... 159
- Registry ..... 160
- Build ..... 160
- Multi-stage Build ..... 161
- Python on Docker ..... 163
  - Deployment Options ..... 163
  - Full env ..... 163
  - Virtualenv ..... 169
  - Pex..... 170
  - Build Options ..... 172
  - One Big Bag ..... 172
  - Copying Wheels Between Stages ..... 172
  - Copying Environment Between Stages..... 173
  - Copying the Pex Executable Between Stages ..... 173
  - Automation with Dockerpy ..... 173
- Twisted on Docker..... 174
  - ENTRYPOINT and PID 1 ..... 174
  - Custom Plugins..... 174
  - NColony ..... 175
- Summary..... 178
- Chapter 5: Using Twisted as a WSGI Server..... 179**
  - Introduction to WSGI ..... 179
    - PEP ..... 180
    - Raw Example..... 181
    - Reference Implementation ..... 183
    - WebOb Example..... 185
    - Pyramid Example..... 186
  - Getting Started ..... 187
    - WSGI Server..... 188
    - Finding Code..... 191

Default Path.....	191
PYTHONPATH .....	192
setup.py.....	192
Why Twisted .....	192
Production vs. Development.....	192
TLS.....	194
Server Name Indication .....	195
Static Files.....	197
Resource Model.....	197
Pure Static.....	198
Combining Static Files with WSGI.....	200
Built-In Scheduled Tasks .....	203
Control Channels .....	206
Strategies for Using Multiple Cores .....	208
Load Balancer.....	208
Opening Socket in Shared Mode .....	210
Other Options.....	213
Dynamic Configuration.....	214
A/B Testable Pyramid App.....	214
Custom Plugin with AMP .....	216
Control Program .....	219
Summary.....	221
<b>Chapter 6: Tahoe-LAFS: The Least-Authority File System .....</b>	<b>223</b>
How Tahoe-LAFS Works .....	224
System Architecture.....	227
How It Uses Twisted.....	229
Problems We've Run Into .....	230
Daemonization Tools.....	231
Internal FileNode Interfaces.....	232
Front-End Protocol Integration.....	233



TABLE OF CONTENTS

- The Web Front End ..... 234
  - File Types, Content-Type, /name/ ..... 237
  - Saving to Disk..... 238
  - Range Headers ..... 238
  - Error Conversion on the Return Side ..... 240
  - Rendering UI Elements: Nevow Templates ..... 241
- The FTP Front End..... 242
- The SFTP Front End..... 248
- Backward-Incompatible Twisted APIs ..... 248
- Summary..... 251
- References ..... 251
- Chapter 7: Magic Wormhole ..... 253**
  - What It Looks Like..... 254
  - How It Works ..... 255
  - Network Protocols, Transfer Latency, Client Compatibility ..... 257
    - Network Protocols and Client Compatibility ..... 258
  - Server Architecture ..... 260
    - Persistent Database ..... 262
  - Transit Client: Cancelable Deferreds ..... 262
  - Transit Relay Server ..... 265
  - Wormhole Client Architecture ..... 267
  - Deferreds vs State Machines, One-Shot Observer..... 268
  - One-Shot Observers..... 271
  - Promises/Futures vs. Deferreds..... 272
  - Eventual-Send, Synchronous Testing..... 275
  - Asynchronous Testing with Deferreds..... 277
  - Synchronous Testing with Deferreds ..... 278
    - Synchronous Testing and Eventual Send..... 281
  - Summary..... 283
  - References ..... 283

<b>Chapter 8: Push Data to Browsers and Micro-services with WebSocket .....</b>	<b>285</b>
Why WebSocket? .....	285
WebSocket and Twisted .....	286
Raw WebSocket, from Python to Python .....	288
Raw WebSocket, Between Python and JavaScript.....	292
More Powerful WebSocket with WAMP .....	294
Summary.....	303
<b>Chapter 9: Applications with asyncio and Twisted.....</b>	<b>305</b>
Core Concepts .....	305
Promises .....	306
Event Loops .....	307
Guidelines .....	308
Case Study: A Proxy with aiohttp and treq .....	312
Summary.....	316
<b>Chapter 10: Buildbot and Twisted.....</b>	<b>317</b>
History of Buildbot .....	317
The Evolution of Buildbot's Async Python.....	318
Migrating Synchronous APIs.....	321
Async Build Steps.....	322
Buildbot's Code .....	323
Async Utilities.....	323
Debounce .....	323
Async Services .....	324
LRU Cache .....	326
Eventual.....	327
Interfacing with Synchronous Code.....	327
SQLAlchemy .....	328
requests.....	329
Docker .....	332
Concurrent Access to Shared Resources .....	333

TABLE OF CONTENTS

- Yield as a Concurrency Barrier ..... 333
- Thread-Pool Functions Should Not Mutate State ..... 334
- DeferredLocks ..... 336
- Testing ..... 336
- Fakes ..... 338
- Summary..... 338
- Chapter 11: Twisted and HTTP/2..... 339**
- Introduction..... 339
- Design Goals ..... 341
  - Seamless Integration..... 341
  - Most-Optimized Behavior by Default..... 343
  - Separating Concerns and Reusing Code ..... 343
- Implementation Concerns..... 344
  - What Is a Connection Anyway? The Value of Standard Interfaces..... 345
  - Multiplexing and Priority..... 348
  - Backpressure..... 355
    - Backpressure in Twisted ..... 356
    - Backpressure in HTTP/2 ..... 359
- Current Status and Future Expansion ..... 362
- Summary..... 363
- Chapter 12: Twisted and Django Channels ..... 365**
- Introduction..... 365
- Channels Building Blocks..... 367
- Message Brokers and Queues ..... 368
- Distributed Multi-Layer Systems in Twisted ..... 369
- Current Status and Future Expansion ..... 371
- Summary..... 371
- Correction to: Expert Twisted: Event-Driven and Asynchronous Programming with Python ..... C1**
- Index..... 373**

# About the Authors

**Mark Williams** works on Twisted. At eBay and PayPal, he worked on high-performance Python web services (over a billion requests a day!), application and information security, and porting enterprise, Java-only libraries to Python.

**Cory Benfield** is an open source Python developer heavily involved in the Python HTTP community. He's a Requests core contributor, a urllib3 core contributor, and the lead maintainer of the Hyper Project, a collection of HTTP and HTTP/2 tools for Python. For his sins, he also helps out with the Python Cryptographic Authority on PyOpenSSL.

**Brian Warner** is a security engineer and software developer, having worked at Mozilla on Firefox Sync, the Add-On SDK, and Persona. He is co-founder of the Tahoe-LAFS distributed secure filesystem, and develops secure storage and communication tools.

**Moshe Zadka** has been part of the open source community since 1995, made his first core Python contributions in 1998, and is a founding member of the Twisted open source project. He also loves to teach Twisted and Python, having given tutorials at several conferences as well as regularly blogging.

**Dustin Mitchell** has contributed to Buildbot and is a member of the TaskCluster team at Mozilla, having also worked on the Release Engineering, Release Operations, and Infrastructure teams.

**Kevin Samuel** has been a Dev and trainer since Python 2.4 and has been putting his skills to work in East Europe, North America, Asia, and West Africa. He has been working closely with the Crossbar.io team and is an active member of the French Python community.

**Pierre Tardy** is a continuous integration specialist with Renault Software Labs, and he is currently the lead committer for Buildbot.

# About the Technical Reviewers



**Julian Berman** is a New York-based software developer and open source contributor. He is the author of the `jsonschema` Python library, an occasional contributor to the Twisted ecosystem, and an active member of the Python community.

**Shawn Shojaie** lives in the clement chaparral of California's Bay Area, where he works as a back-end software engineer. He has worked at Intel, NetApp, and now SimpleLegal, where he happily builds web-based applications for legal services. He spends weekdays writing Django and tuning PostgreSQL, and his weekends contributing to open source projects like `django-pylint`, occasionally editing technical essays. Find out more at him at [shawnshojaie.com](http://shawnshojaie.com).

**Tom Most** is a software engineer in the telecommunications industry. He is a Twisted committer with 10 years of experience of applying Twisted to web services, client libraries, and command-line applications. He is the maintainer of `Afkak`, the Twisted Kafka client. He can be found online at [freecog.net](http://freecog.net) and reached at [twm@freecog.net](mailto:twm@freecog.net).

# Acknowledgments

Thanks to my wife, Jennifer Zadka, without whose support I could not have done it.

Thanks to my parents, Yaacov and Pnina Zadka, who taught me how to learn.

Thanks to my advisor, Yael Karshon, for teaching me how to write.

Thanks to Mahmoud Hashemi, for inspiration and encouragement.

Thanks to Mark Williams, for always being there for me.

Thanks to Glyph Lefkowitz, for teaching me things about Python, about programming, and about being a good person.

—Moshe Zadka

Thanks to Mahmoud Hashemi and David Karapetyan for their feedback. Thanks to Annie for putting up with me while I wrote

—Mark Williams

---

The original version of this book was revised. A correction to this book is available at [https://doi.org/10.1007/978-1-4842-3742-7\\_13](https://doi.org/10.1007/978-1-4842-3742-7_13)

# Introduction

Twisted has recently celebrated its sweet sixteen birthday. It has been around for a while; and in that time, it grew to be a powerful library. In that time, some interesting applications have been built on top of it. In that time, many of us learned a lot about how to use Twisted well, how to think about networking code, and how to architect event-based programs.

After going through the introductory materials that we have on the Twisted site, a common thing to hear is “What now? How can I learn more about Twisted?” The usual way we answered that question is with a question: “What do you want to do with Twisted?” This book shows how to do interesting things with Twisted.

Each of the contributors to this book has done slightly different things with Twisted and learned different lessons. We are excited to present all of these lessons, with the goals of making them common knowledge in the community.

Enjoy!