

УДК 004.438
ББК 32.973.22
315

**Задка М., Уильямс М., Бенфилд К., Уорнер Б.,
Митчелл Д., Сэмюэл К., Тарди П.**

315 Twisted из первых рук / пер. с англ. А. Н. Киселева. – М.: ДМК Пресс, 2020. – 338 с.: ил.

ISBN 978-5-97060-795-4

Эта книга посвящена Twisted – событийно-ориентированному сетевому фреймворку на Python, в котором можно создавать уникальные проекты. В первой части рассматриваются особенности Twisted; на практических примерах показано, как его архитектура способствует тестированию, решает общие проблемы надежности, отладки и упрощает выявление причинно-следственных связей, присущих событийно-ориентированному программированию. Детально описываются приемы асинхронного программирования, подчеркивается важность отложенного вызова функций и сопрограмм. На примере использования двух популярных приложений, `treq` и `klein`, демонстрируются сложности, возникающие при реализации веб-API с Twisted, и способы их преодоления.

Вторая часть книги посвящена конкретным проектам, использующим Twisted. В число примеров входят использование Twisted с Docker, применение Twisted в поликонтейнера WSGI, организация обмена файлами и многое другое.

Читатель должен иметь некоторый опыт работы с Python и понимать основы контейнеров и протоколов. Знакомство с Twisted и с проектами, описанными в книге, не требуется.

УДК 004.438
ББК 32.973.22

Authorized Russian translation of the English edition of Expert Twisted ISBN 978-1-4842-3741-0 © 2019 Moshe Zadka, Mark Williams, Cory Benfield, Brian Warner, Dustin Mitchell, Kevin Samuel, Pierre Tardy.

This translation is published and sold by permission of Packt Publishing, which owns or controls all rights to publish and sell the same.

Все права защищены. Любая часть этой книги не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами без письменного разрешения владельцев авторских прав.

ISBN 978-1-4842-3741-0 (англ.)

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

ISBN 978-5-97060-795-4 (рус.)

© Оформление, издание, перевод, ДМК Пресс, 2020

Содержание

Об авторах	12
Благодарности	14
Введение	15
От издательства	16
Часть I. ОСНОВЫ	17
Глава 1. Введение в событийно-ориентированное программирование с помощью Twisted	18
Примечание о версиях Python	19
Событийно-ориентированное программирование – что это?.....	19
Многократные события.....	20
Мультиплексирование и демultipлексирование	22
Мультиплексор select.....	23
История, аналоги и назначение	23
Сокеты и select	24
События сокета – как, что и почему	25
Обработка событий	26
Цикл обработки событий с select	27
Управляемые событиями клиенты и серверы.....	29
Неблокирующий ввод/вывод	31
Знаем, когда нужно остановиться	31
Отслеживание состояния	32
Наличие информации о состоянии усложняет программы	35
Управление сложностью с помощью транспортов и протоколов	36
Реакторы: работа с транспортом.....	37
Транспорты: работа с протоколами	37
Игра в пинг-понг с протоколами и транспортами	38
Клиенты и серверы со своими реализациями протоколов и транспортов	42
Реакторы Twisted и протоколы с транспортами	43
Преимущества событийно-ориентированного программирования	44
Twisted и реальный мир	46
События и время.....	50
Повторение событий с LoopingCall	53
Интерфейсы событий в zope.interface.....	55

Управление потоком в событийно-ориентированных программах	57
Управление потоком в Twisted с помощью производителей и потребителей	58
Активные производители	59
Потребители	61
Пассивные производители	64
Итоги	64

Глава 2. Введение в асинхронное программирование с Twisted..... 66

Обработчики событий и их композиция	66
Что такое асинхронное программирование?	69
Заполнители для будущих значений	70
Асинхронная обработка исключений	72
Введение в Twisted Deferred	76
Обычные обработчики	76
Ошибки и их обработчики	77
Композиция экземпляров Deferred	80
Генераторы и inlineCallbacks	83
yield	83
send	84
throw	86
Асинхронное программирование с inlineCallbacks	87
Сопрограммы в Python	89
Сопрограммы с выражением yield from	90
Сопрограммы async и await	91
Ожидание завершения экземпляров Deferred	96
Преобразование сопрограмм в Deferred с помощью ensureDeferred	97
Мультиплексирование объектов Deferred	98
Тестирование объектов Deferred	102
Итоги	105

Глава 3. Создание приложений с библиотеками treq и Klein..... 107

Насколько важную роль играют эти библиотеки?	107
Агрегирование каналов	108
Введение в treq	109
Введение в Klein	112
Klein и Deferred	113
Механизм шаблонов Plating в Klein	115
Первая версия агрегатора каналов	117
Разработка через тестирование с использованием Klein и treq	123
Выполнение тестов на устанавливаемом проекте	123
Тестирование Klein с помощью StubTreq	126
Тестирование treq с помощью Klein	133

Журналирование с использованием twisted.logger.....	136
Запуск приложений Twisted с помощью twist.....	141
Итоги	144
Часть II. ПРОЕКТЫ	146
Глава 4. Twisted в Docker	147
Введение в Docker	147
Контейнеры.....	147
Образы контейнеров	148
runc и containerd	149
Клиент	149
Реестр	150
Сборка	150
Многоступенчатая сборка.....	151
Python в Docker	153
Варианты развертывания	153
В виртуальном окружении.....	157
В формате Pex	159
Варианты сборки	160
Один большой образ.....	160
Копирование пакетов wheel между этапами.....	161
Копирование окружения между этапами	161
Копирование файлов Pex между этапами.....	161
Автоматизация с использованием Dockerpy	161
Twisted в Docker	162
ENTRYPOINT и PID 1.....	162
Пользовательские плагины.....	162
NColony.....	162
Итоги	165
Глава 5. Использование Twisted в роли сервера WSGI	166
Введение в WSGI	166
PEP	167
Простой пример.....	168
Базовая реализация.....	170
Пример WebOb	172
Пример Pyramid	173
Начало	174
Сервер WSGI.....	174
Поиск кода.....	177
Путь по умолчанию	177
PYTHONPATH	177

setup.py	177
Почему Twisted?	178
Промышленная эксплуатация и разработка	178
TLS	179
Индикация имени сервера	180
Статические файлы	182
Модель ресурсов	182
Чисто статические ресурсы	183
Комбинирование статических файлов с WSGI	185
Встроенное планирование задач	186
Каналы управления	189
Стратегии параллельного выполнения	191
Балансировка нагрузки	191
Открытие сокета в режиме совместного использования	192
Другие варианты	195
Динамическая конфигурация	195
Приложение Pyramid с поддержкой A/B-тестирования	195
Плагин для поддержки AMP	197
Управляющая программа	200
Итоги	201

Глава 6. Tahoe-LAFS: децентрализованная файловая

система	202
Как работает Tahoe-LAFS	203
Архитектура системы	206
Как система Tahoe-LAFS использует Twisted	208
Часто встречающиеся проблемы	208
Инструменты поддержки выполнения в режиме демона	209
Внутренние интерфейсы FileNode	210
Интеграция интерфейсных протоколов	211
Веб-интерфейс	212
Типы файлов, Content-Type, /named/	214
Сохранение на диск	215
Заголовки Range	215
Преобразование ошибок на возвращающей стороне	216
Отображение элементов пользовательского интерфейса: шаблоны Nevow	217
Интерфейс FTP	218
Интерфейс SFTP	223
Обратная несовместимость Twisted API	223
Итоги	226
Ссылки	226

Глава 7. Magic Wormhole	227
Как это выглядит	228
Как это работает	229
Сетевые протоколы, задержки передачи, совместимость клиентов	231
Сетевые протоколы и совместимость клиентов	231
Архитектура сервера	234
База данных	235
Транзитный клиент: отменяемые отложенные операции	235
Сервер транзитной ретрансляции	238
Архитектура клиента	239
Отложенные вычисления и конечные автоматы, одноразовый наблюдатель	240
Одноразовые наблюдатели	243
Promise/Future и Deferred	244
Отсроченные вызовы, синхронное тестирование	247
Асинхронное тестирование с объектами Deferred	248
Синхронное тестирование с объектами Deferred	249
Синхронное тестирование и отсроченный вызов	252
Итоги	254
Ссылки	254
Глава 8. Передача данных в браузерах и микросервисах с использованием WebSocket	255
Нужен ли протокол WebSocket?	255
WebSocket и Twisted	256
WebSocket, из Python в Python	258
WebSocket, из Python в JavaScript	261
Более мощная поддержка WebSocket в WAMP	263
Итоги	269
Глава 9. Создание приложений с asyncio и Twisted	271
Основные понятия	271
Механизм обещаний	272
Циклы событий	273
Рекомендации	274
Пример: прокси с aiohttp и treq	277
Итоги	280
Глава 10. Buildbot и Twisted	282
История появления Buildbot	282
Эволюция асинхронного выполнения кода на Python в Buildbot	283
Миграция синхронных API	286
Этапы асинхронной сборки	287

Код Buildbot	287
Асинхронные утилиты	288
«Дребезг»	288
Асинхронные службы	288
Кеш LRU	291
Отложенный вызов функций	291
Взаимодействие с синхронным кодом	292
SQLAlchemy	292
requests	293
Docker	295
Конкурентный доступ к общим ресурсам	296
yield как барьер конкуренции	296
Функции из пула потоков не должны изменять общее состояние	297
Блокировки Deferred	298
Тестирование	298
Имитации	300
Итоги	300
Глава 11. Twisted и HTTP/2	301
Введение	301
Цели и задачи	303
Бесшовная интеграция	303
Оптимизация поведения по умолчанию	304
Разделение задач и повторное использование кода	305
Проблемы реализации	306
Что такое соединение? Ценность стандартных интерфейсов	306
Мультиплексирование и приоритеты	309
Противодавление	315
Противодавление в Twisted	317
Противодавление в HTTP/2	319
Текущее положение дел и возможность расширения в будущем	321
Итоги	322
Глава 12. Twisted и Django Channels	323
Введение	323
Основные компоненты Channels	325
Брокеры сообщений и очереди	325
Распределенные многоуровневые системы в Twisted	327
Текущее положение дел и возможность расширения в будущем	328
Итоги	329
Предметный указатель	330