

Оглавление

От издательства.....	16
О научном редакторе русского издания	16
Предисловие	17
Введение.....	19
Благодарности	20
О книге	22
Об авторах.....	26
Иллюстрация на обложке.....	27

Часть 1

Введение в контрактное тестирование

Глава 1. Общий обзор контрактного тестирования	31
1.1. Общий обзор контрактного тестирования	32
1.2. Почему важно применять контрактное тестирование	34
1.3. Когда имеет смысл использовать контрактное тестирование?.....	37
1.4. Общие принципы контрактного тестирования	38
1.4.1. Контрактное тестирование со стороны потребителя	40
1.4.2. Контрактное тестирование со стороны провайдера.....	41
1.5. Контрактное тестирование в действии.....	43
1.5.1. Пример контрактного теста потребителя	43

1.5.2. Пример контрактного теста провайдера	45
Итоги.....	46
Глава 2. Место контрактных тестов в общей стратегии тестирования.....	47
2.1. Контрактное тестирование и другие виды тестов	50
2.1.1. Юнит-тесты.....	51
2.1.2. Интеграционные тесты	51
2.1.3. Сквозные тесты.....	52
2.1.4. Перевернутая пирамида автоматизации тестирования.....	52
2.1.5. Контрактные тесты в пирамиде автоматизации тестирования ..	53
2.2. Может ли контрактное тестирование заменить другие проверки	54
2.3. Преимущества контрактного тестирования	56
2.3.1. Снижение издержек	56
2.3.2. Улучшение качества	57
2.3.3. Преимущества для команды	59
2.4. Знакомим команды с контрактным тестированием	59
2.4.1. Начните с вопроса «зачем»	60
2.4.2. Начните с малого и продвигайтесь постепенно	61
2.4.3. Рассказывайте командам о прогрессе и преимуществах.....	61
2.4.4. Найдите союзников и энтузиастов	62
2.5. Трудности контрактного тестирования.....	63
2.5.1. Временные затраты	63
2.5.2. Изменение мышления разработчиков	64
2.5.3. Недостаточная вовлеченность команд	65
2.5.4. Объемная техническая часть.....	65
2.5.5. Сотрудничество с внешними командами	66
2.6. Практическое руководство по внедрению контрактного тестирования	66
2.6.1. Изучите инструменты	67
2.6.2. Определите потребителей и провайдеров	68

2.6.3. Создайте контракт	68
2.6.4. Напишите первый тест для потребителя и провайдера	69
2.6.5. Добавьте контрактное тестирование в процесс CI/CD	70
Итоги.....	72
Глава 3. Технический обзор контрактного тестирования	74
3.1. Потребитель.....	77
3.2. Провайдер.....	78
3.3. Контракт.....	79
3.4. Брокер контрактов.....	81
3.5. Жизненный цикл тестирования, ориентированного на потребителя.....	84
3.6. Инструменты для контрактного тестирования	87
3.6.1. Инструменты, ориентированные на контракты.....	88
3.7. Подходы к контрактному тестированию	89
3.7.1. Подход, ориентированный на потребителя	90
3.7.2. Подход, ориентированный на провайдера	90
3.8. Какие типы коммуникации поддерживает контрактное тестирование.....	94
3.8.1. GraphQL.....	95
3.8.2. Системы, управляемые событиями	95
Итоги.....	96

Часть 2

Контрактное тестирование, ориентированное на потребителя

Глава 4. Разработка контрактных тестов потребителя для веб-приложений	101
4.1. Принципы написания контрактных тестов потребителя	102
4.1.1. Основная цель теста потребителя.....	103

10 Оглавление

- 4.1.2. Почему важно использовать моки вместо стабов 103
- 4.1.3. Использование нечеткого сопоставления для ослабления связанности 104
- 4.1.4. Почему важно выполнять контрактные тесты изолированно 106
- 4.1.5. Can-i-deploy. Проверка: можно ли приступить к развертыванию 108
- 4.2. Установка инструментов контрактного тестирования на стороне потребителя 108
 - 4.2.1. Требования 109
- 4.3. Контрактные тесты для веб-приложения на стороне потребителя..... 109
 - 4.3.1. Общий обзор API фильмов 110
 - 4.3.2. Общий обзор веб-приложения 115
 - 4.3.3. Контракт потребителя 116
 - 4.3.4. Первый контрактный тест потребителя..... 117
 - 4.3.5. Другие языки программирования..... 123
- 4.4. Публикация контракта в Pact Broker..... 123
 - 4.4.1. Отправляем изменения в GitHub..... 126
- Итоги..... 127

Глава 5. Разработка контрактных тестов потребителя для мобильных клиентов 128

- 5.1. Настройка потребителя Pact для Android-приложений 130
 - 5.1.1. Требования 130
 - 5.1.2. Реализация мобильного приложения 132
- 5.2. Разработка теста потребителя для Android 133
- 5.3. Создание контракта потребителя для Android 141
 - 5.3.1. Сопоставители Pact 144
 - 5.3.2. POST-запросы Pact 146

5.4. Упражнение	146
Итоги	147
Глава 6. Разработка тестов провайдера в контрактном тестировании, ориентированном на потребителя	148
6.1. Основные принципы написания контрактных тестов провайдера.....	149
6.1.1. Цель теста провайдера.....	150
6.1.2. Состояния провайдера	152
6.1.3. Can-i-deploy. Проверка: можно ли приступить к развертыванию	154
6.2. Настройка инструментов контрактного тестирования на стороне провайдера.....	154
6.3. Контрактные тесты провайдера для веб-приложения	155
6.3.1. Пишем тест провайдера	155
6.3.2. Состояния провайдера	160
6.4. Контрактные тесты провайдера для мобильного приложения	167
6.5. Упражнение	168
Итоги.....	170
Глава 7. Контрактное тестирование для GraphQL	171
7.1. Краткое введение в GraphQL.....	172
7.2. Обзор приложения на GraphQL	175
7.2.1. Серверная часть приложения на GraphQL.....	176
7.2.2. Запуск сервера GraphQL.....	179
7.2.3. Обзор потребителя	180
7.3. Контрактные тесты потребителя между потребителем и слоем GraphQL.....	182
7.3.1. Импортируем зависимости.....	183
7.3.2. Настраиваем мок-провайдера.....	183
7.3.3. Регистрируем ожидания потребителя.....	184

12 Оглавление

- 7.3.4. Запускаем тест потребителя и создаем контракт..... 185
- 7.3.5. Публикация контракта на Pact Broker 188
- 7.4. Контрактные тесты провайдера между потребителем и слоем GraphQL..... 189
- 7.5. Упражнение 192
- Итоги..... 193

- Глава 8. Контрактное тестирование для архитектуры, управляемой событиями 194**
- 8.1. Введение в тестирование асинхронных API..... 196
- 8.2. Настройка сообщений потребителя с помощью Kafka 197
 - 8.2.1. Требования 198
- 8.3. Реализация потребителя Kafka 198
- 8.4. Реализация теста потребителя для обработчика событий 204
- 8.5. Создание контракта потребителя событий..... 207
- 8.6. Реализация производителя Kafka 209
- 8.7. Реализация теста производителя событий 212
- Итоги..... 215

- Глава 9. Сохранение, размещение и обеспечение безопасности контрактов 216**
- 9.1. Зачем нужен Pact Broker 218
 - 9.1.1. Основные функции Pact Broker..... 219
- 9.2. Дополнительные функции Pact Broker 222
- 9.3. Облачный сервис Pact Broker (PactFlow) 225
 - 9.3.1. Настройка PactFlow 225
 - 9.3.2. API-токен PactFlow 226
 - 9.3.3. Управление пользователями PactFlow 228
 - 9.3.4. Другие функции PactFlow 229
- 9.4. Внутренний сервер Pact Broker..... 232
 - 9.4.1. Разворачиваем Pact Broker с помощью Docker 232
 - 9.4.2. Разворачиваем Pact Broker с помощью Kubernetes 235

9.5. Сравнение облачного и внутреннего сервисов Pact Broker	236
Итоги.....	237
Глава 10. Интеграция контрактного тестирования в пайплайн CI/CD.....	239
10.1. Основные шаги настройки CI/CD для потребителя и провайдера	240
10.1.1. Настройка GitHub Actions	241
10.1.2. Пример настройки для других платформ CI/CD	243
10.2. Дополнительные функции для улучшения настройки CI/CD	243
10.2.1. Настройка версий потребителя и провайдера	244
10.2.2. Настройка веток	245
10.2.3. Создание окружений	250
10.2.4. Сохранение развертываний и/или поставок	252
10.2.5. Инструмент can-i-deploy	256
10.3. Упражнения	258
10.4. Вебхуки	260
10.4.1. Создаем вебхук Pact.....	261
10.4.2. Создаем отдельный процесс GitHub для проверки провайдера	265
10.4.3. Обновляем свойства провайдера.....	266
10.4.4. Выполняем команду can-i-deploy второй раз.....	266
10.4.5. Проверяем вебхук.....	267
Итоги.....	268

Часть 3

Контрактное тестирование, ориентированное на провайдера

Глава 11. Реализация контрактного тестирования, ориентированного на провайдера	270
11.1. Введение в контрактное тестирование, ориентированное на провайдера.....	272

14 Оглавление

11.1.1. Зачем нужно PDCT?	274
11.1.2. Как PDCT решает трудности CDCT	275
11.1.3. Недостатки PDCT	276
11.2. Реализация адаптеров теста потребителя	277
11.2.1. Адаптер потребителя Cypress	277
11.2.2. Адаптер потребителя WireMock	280
11.2.3. Публикация контрактов потребителя	283
11.3. Реализация тестов провайдера с помощью инструментов тестирования API	283
11.3.1. Пример с Postman и OpenAPI	283
11.3.2. Пример с Dredd и OpenAPI	285
11.3.3. Публикация API-спецификации провайдера и результатов тестов	287
11.3.4. Проверка контрактов потребителя	288
Итоги	290
Глава 12. Замена интеграционных тестов контрактными	291
12.1. Когда интеграционные тесты работают лучше контрактных	293
12.2. Какие интеграционные тесты стоит перевести в контрактные	296
12.3. Переводим интеграционные тесты в контрактные	297
12.4. Переводим сквозные тесты в контрактные	303
Итоги	305
Приложения	
Приложение А. Сравнение инструментов	308
А.1. Инструменты, основанные на контрактах	308
А.1.1. Pact	308
А.1.2. Pact Broker	309
А.1.3. PactFlow	309
А.1.4. Pact CLI	309

A.1.5. Плагины Pact	310
A.1.6. Swagger Mock Validator	310
A.1.7. Spring Cloud Contract	310
A.1.8. Postman	311
A.2. Инструмент, основанный на схеме.....	312
A.2.1. JSON Schema.....	312
A.3. Варианты размещения Pact Broker	313
Приложение Б. Подготовка окружения для контрактного тестирования потребителя (веб-приложения)	315
Б.1. Устанавливаем Node	315
Б.2. Устанавливаем pact-js	316
Б.3. Копируем проект	318
Б.4. Загружаем проект.....	320
Б.5. Устанавливаем Visual Studio Code	321
Приложение В. Ответы на упражнения	323
В.1. Ответы на упражнения главы 5.....	323
В.2. Ответы на упражнения главы 6.....	324
В.3. Ответы на упражнения главы 7.....	328
В.4. Ответы на упражнения главы 10	331