

---

# Оглавление

|  |           |
|--|-----------|
| <b>Предисловие .....</b>   | <b>11</b> |
| Добро пожаловать .....   | 11        |
| Условные обозначения .....   | 12        |
| Использование примеров кода .....                                  | 13        |
| Благодарности .....  | 13        |
| От издательства .....  | 14        |
| <b>Глава 1. Коротко о Spring Boot .....</b>                        | <b>15</b> |
| Три основополагающие возможности Spring Boot .....                 | 15        |
| Упрощение управления зависимостями с помощью стартовых пакетов ... | 15        |
| Упрощение развертывания с помощью исполняемых JAR-файлов .....     | 17        |
| Автоконфигурация .....   | 19        |
| Резюме .....   | 21        |
| <b>Глава 2. Выбираем инструменты и приступаем к работе .....</b>   | <b>22</b> |
| Maven или Gradle? .....  | 22        |
| Apache Maven .....   | 22        |
| Gradle .....   | 24        |
| Выбор между Maven и Gradle .....                                   | 25        |
| Java или Kotlin .....  | 26        |
| Java .....   | 26        |
| Kotlin .....   | 27        |
| Выбор между Java и Kotlin .....                                    | 28        |
| Выбираем версию Spring Boot .....                                  | 28        |
| Spring Initializr .....  | 29        |
| Прямоком из командной строки .....                                 | 33        |
| Работа в интегрированных средах разработки .....                   | 36        |
| Прогулка по функции main() .....                                   | 38        |
| Резюме .....   | 39        |

---

|  |           |
|--|-----------|
| <b>Глава 3. Создаем первый Spring Boot REST API.....</b>         | <b>41</b> |
| «Как» и «почему» API .....                                       | 41        |
| Что такое REST и почему это важно.....                           | 42        |
| API в стиле HTTP-глаголов.....                                   | 43        |
| Возвращаемся к Initializr.....                                   | 44        |
| Создание простого класса предметной области.....                 | 45        |
| GET.....   | 47        |
| Коротко об аннотации @RestController .....                       | 47        |
| POST.....  | 51        |
| PUT.....   | 52        |
| DELETE.....  | 52        |
| И не только .....  | 53        |
| Доверяй, но проверяй .....                                       | 55        |
| Резюме.....  | 58        |
| <br>   |           |
| <b>Глава 4. Добавление в приложение Spring Boot доступа</b>      |           |
| <b>к базе данных.....</b>  | <b>60</b> |
| Подготовка автоконфигурации для доступа к базе данных.....       | 60        |
| Чего мы надеемся добиться .....                                  | 61        |
| Добавление зависимости для базы данных.....                      | 61        |
| Добавление кода .....  | 63        |
| Сохранение и извлечение данных .....                             | 70        |
| Наводим лоск .....   | 74        |
| Резюме.....  | 75        |
| <br>   |           |
| <b>Глава 5. Настройка и контроль приложения Spring Boot.....</b> | <b>77</b> |
| Конфигурация приложения.....                                     | 78        |
| @Value .....   | 79        |
| @ConfigurationProperties.....                                    | 84        |
| Возможные сторонние решения.....                                 | 89        |
| Отчет об автоконфигурации.....                                   | 92        |
| Actuator (актуатор).....   | 94        |
| Открываем доступ к Actuator.....                                 | 101       |
| Лучше учитываем среду приложения с помощью Actuator.....         | 101       |
| Регулировка уровня журналирования с помощью Actuator.....        | 104       |
| Резюме.....  | 106       |

|   |            |
|---|------------|
| <b>Глава 6. Займемся данными по-настоящему .....</b>  | <b>107</b> |
| Описание сущностей .....  | 108        |
| Поддержка шаблонов .....  | 109        |
| Поддержка репозитория .....   | 109        |
| @Before .....   | 110        |
| Создание с помощью Redis сервиса на основе шаблона.....   | 110        |
| Инициализация проекта .....   | 111        |
| Разработка сервиса Redis .....  | 112        |
| Преобразование из шаблона в репозиторий .....   | 121        |
| Создание сервиса на основе репозитория с помощью<br>Java Persistence API.....                       | 125        |
| Инициализация проекта .....   | 125        |
| Разработка JPA-сервиса (MySQL).....   | 126        |
| Загрузка данных.....  | 131        |
| Создание сервиса на основе репозитория с помощью<br>документоориентированной базы данных NoSQL..... | 136        |
| Инициализация проекта .....   | 136        |
| Разработка сервиса MongoDB.....   | 138        |
| Создание сервиса на основе репозитория с помощью графовой<br>базы данных NoSQL .....                | 144        |
| Инициализация проекта .....   | 145        |
| Разрабатываем сервис Neo4j .....  | 145        |
| Резюме.....   | 156        |
| <b>Глава 7. Создание приложений с помощью Spring MVC .....</b>                                      | <b>158</b> |
| Что такое Spring MVC .....  | 158        |
| Взаимодействия конечного пользователя с помощью шаблонизаторов.....                                 | 159        |
| Инициализация проекта .....   | 160        |
| Разработка приложения Aircraft Positions.....   | 161        |
| Передача сообщений .....  | 167        |
| Подключаем PlaneFinder .....  | 168        |
| Расширяем приложение Aircraft Positions.....  | 173        |
| Формирование диалогов с помощью WebSocket .....   | 178        |
| Что такое WebSocket .....   | 179        |
| Рефакторинг приложения Aircraft Positions .....   | 179        |
| Резюме.....   | 187        |

---

|   |            |
|---|------------|
| <b>Глава 8. Реактивное программирование: Project Reactor и Spring WebFlux.....</b>                | <b>189</b> |
| Введение в реактивное программирование.....   | 189        |
| Манифест реактивных систем.....   | 190        |
| Project Reactor.....  | 193        |
| Tomcat и Netty.....   | 195        |
| Реактивный доступ к данным.....   | 196        |
| R2DBC и база данных H2.....   | 196        |
| Реактивный Thymeleaf.....   | 209        |
| RSocket и полностью реактивное взаимодействие между процессами.....                               | 210        |
| Что такое RSocket.....  | 210        |
| Применяем RSocket на практике.....  | 211        |
| Резюме.....   | 216        |
| <br>  |            |
| <b>Глава 9. Тестирование приложений Spring Boot для повышения их готовности к продакшену.....</b> | <b>218</b> |
| Модульное тестирование.....   | 218        |
| Знакомимся с аннотацией @SpringBootTest.....  | 219        |
| Важнейшие модульные тесты для приложения Aircraft Positions.....                                  | 220        |
| Рефакторинг кода для лучшего тестирования.....  | 226        |
| Тестовые срезы.....   | 234        |
| Резюме.....   | 240        |
| <br>  |            |
| <b>Глава 10. Безопасность приложений Spring Boot.....</b>   | <b>241</b> |
| Аутентификация и авторизация.....   | 241        |
| Аутентификация.....   | 242        |
| Авторизация.....  | 243        |
| Коротко о Spring Security.....  | 244        |
| HTTP-брандмауэр.....  | 244        |
| Цепочки фильтров безопасности.....  | 244        |
| Заголовки запросов и ответов.....   | 245        |
| Реализация аутентификации и авторизации на основе форм с помощью Spring Security.....             | 245        |
| Добавление зависимостей Spring Security.....  | 246        |
| Добавляем аутентификацию.....   | 252        |
| Авторизация.....  | 259        |

|  |            |
|--|------------|
| Реализация OpenID Connect и OAuth2 для аутентификации<br>и авторизации ..... | 268        |
| Клиентское приложение Aircraft Positions .....                               | 270        |
| Сервер ресурсов PlaneFinder .....  | 277        |
| Резюме .....   | 285        |
| <b>Глава 11. Развертывание приложений Spring Boot .....</b>                  | <b>287</b> |
| Возвращаемся к исполняемым JAR-файлам Spring Boot .....                      | 288        |
| Сборка «полностью исполняемого» JAR-файла Spring Boot.....                   | 289        |
| Что это нам дает .....   | 295        |
| Разобранные JAR-файлы .....  | 295        |
| Развертывание приложений Spring Boot в контейнерах.....                      | 302        |
| Создание образа контейнера из IDE.....                                       | 304        |
| Создание образа контейнера из командной строки .....                         | 306        |
| Проверяем наличие образа .....   | 307        |
| Запуск контейнеризованного приложения .....                                  | 308        |
| Утилиты для исследования образов контейнеров приложений<br>Spring Boot.....  | 309        |
| Pack.....  | 309        |
| Dive.....  | 311        |
| Резюме .....   | 312        |
| <b>Глава 12. Углубляемся в реактивное программирование.....</b>              | <b>313</b> |
| Когда следует использовать реактивное программирование .....                 | 314        |
| Тестирование реактивных приложений.....                                      | 315        |
| Но сначала — рефакторинг .....   | 316        |
| А теперь — тестирование .....  | 323        |
| Диагностика и отладка реактивных приложений.....                             | 332        |
| Hooks.onOperatorDebug() .....  | 333        |
| Контрольные точки .....  | 343        |
| ReactorDebugAgent.init() .....   | 346        |
| Резюме .....   | 348        |
| <b>Об авторе .....</b>   | <b>350</b> |
| <b>Об иллюстрации на обложке .....</b>                                       | <b>351</b> |