Spring Advanced

Код: JVA-075

Длительность: 40 ч.

Описание:

Курс посвящен темам, необходимым для профессионального применения Spring Framework на практике, таким как расширенные вопросы применения Spring REST, технологии HATEOAS и CORS, документирование REST-сервисов, сериализация и ограничение данных, передаваемых клиенту, тонкости применения аннотации @Transactional, применение оптимистических транзакций, JWT-авторизация и сервер авторизации OAuth2, практическое применение реактивного программирования с использованием WebFlux, тестирование Spring-сервисов, планировщик задач и Actuator, развертывание в Docker.

Цели:

Познакомить слушателей с расширенными темами применения Spring Framework, предоставить примеры применения.

Разбираемые темы:

1. Spring Data REST - расширенные темы (5h)

1) Spring REST (3h)

- Введение. Обзор Spring REST (1h)
- Условные операции с применением заголовков (ETag, If-Match, If-None-Match, If-modified-since) (0.5h)
- REST события (0.5h)
- Модель зрелости REST-сервисов (0.5h)
- Spring HATEOAS (0.5h)

2) API RestTemplate и WebClient для написания клиентов REST-сервисов (1h)

3) Документирование REST-сервисов с использованием Swagger и SpringFox (1h)

Домашняя работа: ~2 ч

2. Object mapping и валидация (5h)

- Ограничение данных, передаваемых клиенту: @JsonIgnore, nullifying fields, DTO (1h)
- Проекции и ограничения в Spring Data REST (1h)

- Использование MapStruct для автоматического мэппинга в DTO (1h)
- Пользовательские сериализаторы и десериализаторы (1h)
- Валидация данных модели с помощью аннотаций javax.validation и кастомных аннотаций (1h)

Домашняя работа: ~2 ч

3. Транзакции (6h)

- Анотация @Transactional (0.5h)
- Уровни изоляции транзакций (1h)
- Распространение транзакций (2h)
- Обработка исключений в транзакциях (0.5h)
- Проблема вызова транзакции из того же бина (0.5h)
- Оптимистические блокировки (0.5h)
- Подключение нескольких БД к проекту (0.5h)
- Распределенные транзакции и паттерн SAGA (0.5h)

4. Безопасность для REST-сервисов (JWT авторизация и управление ролями) (6h)

- Подходы к использованию безопасности в Spring (1h)
- Разработка сервера UAA с использованием OAuth2 (1h)
- JWT токены (2h)
- Использование Spring Security для доступа на основе ролей, демонстрация примеров (2h)

Домашняя работа: ~2 ч

5. Реактивное программирование на Spring WebFlux (8h)

- Обзор реактивного подхода (0.5h)
- Спецификация реактивных потоков в Java 9 (0.5h)
- Классы Mono и Flux (0.5h)
- Операторы (2h)
- Реактивные драйверы БД R2DBC (0.5h)
- Реактивный Spring Data (0.5h)
- Построение REST API с использованием WebFlux (1h)
- WebClient для получения реактивных данных (0.5h)
- Протокол RSocket (1h)
- Реактивный доступ к RabbitMQ (0.5h)
- Реактивные паттерны и бенчмарки (0.5h)

Домашняя работа: ~2 ч

6. Тестирование сервисов Spring (4h)

- Юнит-тесты с применением Mockito (0.5h)
- Интеграционное тестирование (1h)
- Аннотации тестирования Spring (0.5h)
- Фреймворк TestContext (0.5h)
- Серверные тесты REST (0.5h)
- Тестирование REST сервисов в IntelliJ IDEA (0.5h)
- Тестирование WebFlux (0.5h)

Домашняя работа: ~2 ч

7. Spring Планировщики (0.5h)

- Аннотация @Scheduled
- fixedRate, fixedDelay, initialDelay
- использование выражений CRON

Домашняя работа: ~30 мин

8. Spring кэширование (2h)

- Использование кеширования
- Использование и параметры настройки EhCache
- @Cacheable
- @CacheEvict, @CachePut
- @CacheConfig
- Conditional кэширование
- Java-based кэширование
- Кэширование в Spring Data
- Кэширование в Spring на основе EhCache
- Конфигурирование EhCache

Домашняя работа: ~1 ч

9. Spring Boot Actuator (1h)

- Actuator endpoints
- Метрики в Spring Boot
- Мониторинг и управление через HTTP
- Инструмент мониторинга Spring Boot Admin

10. Развертывание приложения Spring (2,5h)

- Развертывание в Docker (0.5h)
- Развертывание в nginx as a front serverв качестве фронт-сервера (включая балансировку нагрузки) (0.5h)
- HTTP кэширование при помощи NGINX (0.5h)

- Развертывание приложения Spring с помощью nginx - контейнерная установка, docker-compose (0.5h)

Домашняя работа: ~30 min

Целевая аудитория:

Разработчики Java, архитекторы Java.

Предварительная подготовка – общее:

Уверенное владение Java, знание Spring.