

Защита проекта

Тема: Онлайн-библиотека BookVerse

Евстигнеев Никита

Java 17 • Spring Boot 3.5.5 • Microservices
JWT Security • RabbitMQ • Docker Compose



План защиты

Цель и задачи проекта

Какие технологии использовались

Что получилось

Выводы

Вопросы и рекомендации



Цель и задачи проекта

Цель проекта: создать микросервисную онлайн-библиотеку с авторизацией через JWT и коммуникацией сервисов через REST/очереди.

1. Реализовать отдельный сервис аутентификации (JWT + JWK).
2. Реализовать каталог книг и выдачу/возврат (JPA + Flyway).
3. Добавить коммуникацию через RabbitMQ (события выдачи/возврата).
4. Подключить отказоустойчивость: CircuitBreaker/Retry + Cache.
5. Упаковать систему в Docker Compose и добавить тесты.



Какие технологии использовались

1. Spring Boot 3.5.5, Java 17 (единый стек и быстрый запуск).
2. Spring MVC (REST API) для сервисов.
3. Spring Security + JWT (Resource Server) для защиты API.
4. Auth-service: выпуск JWT, публикация JWKSet (ключи в одном месте).
5. Spring Data JPA + Flyway (PostgreSQL/H2) для хранения и миграций.
6. RabbitMQ (AMQP, TopicExchange) для событий выдачи/возврата.
7. Resilience4j: CircuitBreaker/Retry для устойчивости внешних вызовов.
8. Cache: Caffeine для ускорения чтения каталога.
9. API Gateway: Spring Cloud Gateway для единой точки входа.
10. Docker Compose для локального запуска всего стенда.



Что получилось

- docker-compose.yml и запуск стенда одной командой
- REST эндпоинты через API Gateway
- миграции Flyway и модель БД
- обмен событиями RabbitMQ (borrow/return)



Выводы

1. Цель достигнута: реализован полный сценарий (каталог -> выдача -> возврат).
2. Легко: CRUD и Flyway. Сложнее: согласовать JWT/JWK между сервисами.
3. Время: объём ~3-4 ДЗ (архитектура + безопасность + Docker).
4. Полезность: 9/10 — собрал в одном месте ключевые практики курса.

