Aliaksey Kalinckovich.

STUDENT SOFTWARE ENGINEER

Education

A third-year student at the Belarusian State University of Informatics and Radioelectronics (BSUIR).

My faculty: Computer Systems and Networks

My area of specialisation is: Programme Engineering (POIT)

Language proficiency

English - B2 (HR opinion)

Projects:

Github: AlexKalinckovich (Aliaksei)

Location:

Belarus, Minsk

Contacts:

Email: aleckseykalickovich@gmail.com
TG: @alexKalinck

Student Software engineer with 6 month of internship experience.

Main stack programming languages:

Java, Scala, SQL, JavaScript.

Backend

Spring (Boot, Security, Data, MVC, Cloud), JPA (Hibernate), MapStruct, Lombok,

Build Tools

Worked with: Maven.
Familiar with: Gradle.

Protocols

Worked with: REST, WebSocket
Familiar with: SOAP

Spring-Web-Communication

Worked with: WebClient.
Familiar with: FeignClient, RestTemplate.

Formats

Worked with: JSON, XML, YAML, ENV.
Familiar with: BSON.

Testing

Worked with: JUnit, Mockito, Testcontainers, WireMock.

Databases

Worked with: PostgreSQL, MySQL, Redis, MongoDB, Liquibase, Debezium.

Event Streaming Platforms

Worked with: Apache Kafka.
Familiar with: RabbitMQ.

DevOps

Worked with: Docker, Docker-Compose, Kubernetes, CI/CD (Github Actions).
Familiar with: Helm

Frontend

Worked with: React, TypeScript, HTML, CSS

Source control systems

Worked with: Git, GitHub, GitLab.

Pet-Project

E-COMMERCE PLATFORM

This is a backend system created for an online retail store, designed to manage the entire sales process from product to final delivery. It contains the following microservices:

- ApiGateway (where all responses come from and where the JWT token is checked).
- Auth-service (where all user credentials are checked and stored)

What I learned here:

- Implemented REST API endpoints for Product Management, implemented CRUD operations using Spring Boot and MySQL;
- Contributed to the event-driven architecture by developing Kafka consumers to process order status updates from payment and shipping services;
- Developed and executed database migrations to support new application features and schema updates using Liquibase;
- Improved CI/CD pipelines in Github Actions, troubleshooting build failures and adding steps to automate code quality checks;
- Reduced the database query response time by adding indexes and materialized views;

- User-service (where all user actions are cached using Redis)
- OrderService (order creation logic, also my first Kafka consumer and producer).
- Payment service (also a Kafka consumer and producer, and my first experience with Mongo!)
- React frontend (where I learnt about different hooks and had my first experience with frontend)!

- Worked with MongoDB to optimise the storage of complex, in-depth payment data.
- Send some API calls and test them using the Spring WebClient and WebFlux tools.
- Studied types of security attacks (CSRF, SQL injection, XSS) and Cors features using Spring Boot
- Redirect the following request to the necessary microservice via my ApiGateway using Spring Cloud.
- Containerise my application using different tools, ranging from the simple, such as Docker Compose, to the complex, such as Kubernetes.