Emin AKSOY

eminaksoy35@gmail.com
+90 (543) 890 2860
emin-aksoy-b70548203
github.com/Eukolos
eminksy.me

SUMMARY

After completing my International Relations degree, I planned to make the most of the Covid-19 period by joining the military as a reserve officer. I attended Cadet classes and coordinated distance education at the Land Military Academy. After my mission, I received the 2020-2021 service memory. Upon returning, I pursued software development from January 2022 and later transitioned to my current position in accounting at AYZ Spare Parts Ltd. Co., while still continuing my software development journey since January 2022.

EDUCATION

09.2015 – 07.2019
International Relations
(B.A. – 3.20)
Faculty of Economics
University of Ataturk – Erzurum

09.2022 - Current
Computer Programming
(Distance Education)
University of Anatolia – Eskisehir

LANGUAGES

- Turkish (Native)
- English (B2)

ACHIEVEMENTS

- First degree in International Relations Department.
- NDU Head of International Relations Department – Memory of Services 2020-2021

OTHER INFORMATIONS

- Birth of Day: 22.07.1997
- Military Status: Reserve Officer (2020-2021)
- Driving License: B1

KEY SKILLS

- Java
- Spring
  - Spring Boot
  - Spring JPA
  - Spring Security
  - Spring Cloud
- RDBMS
- Docker
- Git

SKILLS

- HTML, CSS, JS
- Node.js
- Vue.js
- Flutter

REFERENCE

Name: İsmail Sarı
Title: Principal Software Engineer
Company: Scalefocus Co.
Relationship: Mentor
Phone: +90 (506) 119 9201
Contact: rootsaak@gmail.com / Linkedin

CAREER OBJECTIVE

As someone striving to become an experienced Java Developer with a solid knowledge base, I am excited to continue my career with a motivating team that values creative problem-solving.

WORK EXPERIENCE

Accounting & ERP
Rantech – AYZ Spare Parts LTD. CO.
July 2022 – Current / Bağcılar, IST

PROJECT EXPERIENCE

Arcade Blog Backend & Frontend

Our project uses Express.js in Node.js Runtime Engine, with middleware like Joi for validation and JWT for authentication. The user model handles project creation (blogs/posts).

Forgotten passwords are managed using NodeMailer Package with Crypto.js's HmacSHA256 hasher. Logs are handled through Winston Package with Mongoose's hook for model layer. MongoDB is our database, Dockerized for easy deployment.

For the frontend, we use Flutter with Cubit for state management and Dio for backend communication. Planned features include post comments, likes, and Dockerizing the service. Overall, we have a solid backend and a Flutter-based frontend.

Node.js · Express.js · Mongodb · Mongoose · JWT · Winston · Joi
Flutter · Cubit · Dio · Figma

Proud of Dadash - More

After observing the Node.js community's preference for TypeScript due to its type safety, I opted to use Java for my projects. One of the projects I designed was a restaurant management system. During the development process, I utilized H2DB for easy configuration, Lombok to reduce boilerplate code and provide a builder design pattern, and JPA to handle database requests. I effectively managed exceptions, customizing them to follow Google Error format. For authentication and authorization, I successfully implemented Spring Security and JWT after thorough research. The project was tested using Postman for endpoints. Future plans include adding JUnit for testing purposes.

Spring Boot · Spring JPA · Spring Security · JWT · H2 Database
Lombok · Postman

Spring Boot 3 Example - More

After Spring Boot 3.0.0 release, I saw new feature about Graalvm and Spring Boot 3. Now we can build spring boot projects as a native image by graalvm. After native build, we can run projects from docker-compose.

Spring Boot 3.0.0. · GraalVM · Maven · Docker

Friendly Library - More

In my library system project, I embraced Spring Cloud technologies to implement an efficient microservice architecture. To cater to different data requirements, I utilized MongoDB, a NoSQL database, for the library service, and PostgreSQL, an SQL database, for another service I developed. Communication between these services was facilitated through the use of Feign Client. I also ensured that any exceptions that might arise during these interactions were properly handled and customized to adhere to the Google Error Format.

Comprehensive unit tests were written for both the service layers and controller layers to ensure the reliability and correctness of the system. Integration tests of the databases running on Docker images were made possible thanks to Testcontainer.

To manage service discovery and registration, I employed Eureka Service as the Registry Service, streamlining communication between the microservices within the architecture.

Spring Boot · Spring JPA · Spring Cloud · TestContainers · JUnit · @Schedule
Feign Client · Eureka · Mongodb · Postgresql · Docker · JavaMailer