

# HALIL IBRAHIM KOCAOZ

## SOFTWARE DEVELOPER 4 YEARS EXPERIENCE BACK-END EXPERIENCES

### SOFTWARE DEVELOPER

#### Rasyonet & EquityRT

Services, analysis and research about Financial Markets

January 2024 to present

- Deeply involved in improving Rasyonet's **infrastructure** about **real time system design**.

### SOFTWARE DEVELOPER

#### Shopi

Omnichannel retailing platform

September 2021 to 2024 January

- I played a crucial role at Shopi, **communication** between **customer systems** and **Shopi services** via **Integration** layers, also applying **microservices** with **event/domain driven design** for **scalable** and **adaptable** solutions.
- Migrated almost **one million order data** to Shopi.
- Integration projects with high-profile clients like **Vakko Group, Etoile Group (UAE), IKEA (TR), Ebebek, and Greyder**.
- Software flows and architecture documentation.

### SOFTWARE DEVELOPMENT INTERN

#### VBT Yazılım A.Ş

Summer Intern - 4 week - 2021

General education about modern software development.

- Containerization/Docker, CI/CD - Azure DevOps
- Logging and Monitoring - Serilog & Azure Application Insights
- Agile Software Development
- Programming approaches

### SOFTWARE DEVELOPMENT INTERN

#### Unibim Bilişim

Summer Intern - 4 week - 2018

Hands-on experience in software development, focused on:

- Database Design: Designed a relational database for an enterprise website and blog pages.
- Backend Development: Contributed to the development of the CMS using ASP.NET MVC.
- Frontend Development: Created user interfaces with Razor Pages, HTML, Bootstrap, and JavaScript.

[halilkocaoz.com/linkedin](https://halilkocaoz.com/linkedin)

[halilkocaoz.com/github](https://halilkocaoz.com/github)

[halilkocaoz.com/blogs](https://halilkocaoz.com/blogs)

**+905558780668**

**[halil.i.kocaoz@gmail.com](mailto:halil.i.kocaoz@gmail.com)**

## IN SHORT, I CAN DEVELOP/DO

- Server-side software.
- **Distributed & scalable** software systems.
- **End-to-end** system and software integration.
- **RESTful, gRPC** services, **SignalR** hubs for mobile and web applications or any type of client.

## SKILLS

### PROGRAMMING LANGUAGES

- C#, Go

### SERVER-SIDE SOFTWARES & TOOLS

- Kafka, RabbitMQ, Azure Service Bus
- Hangfire

### DATABASE, CACHING Systems & ORMS

- MSSQL, MongoDB, Redis
- Entity Framework, Dapper

### GENERAL SOFTWARE DEVELOPMENT KNOWLEDGE

- OOP, SOLID
- Software Documentation
- Microservice & Event/Domain Driven Design basics
- Message streaming & queue mechanisms
- Real time system
- REST, gRPC, SignalR
- Mediator pattern, CQRS
- Docker

## EDUCATIONS

**Çukurova Üniversitesi**

**Computer Engineering**

**Düzce Üniversitesi**

**Computer Programming**

You can reach the latest version of the resume [here](#)

# OPEN SOURCE CONTRIBUTIONS & REPOSITORIES

## [kafka-retry-job - Trendyol \(contribution\)](#)

<https://github.com/Trendyol/kafka-retry-job>

Kafka Retry Job is an open-source project for moving Kafka messages from error topics to retry topics

## [eShop - dotnet \(contribution\)](#)

<https://github.com/dotnet/eshop>

A reference application implementing an eCommerce web site using a services-based architecture with .NET 8 Aspire

## [Go-dcp-kafka - Trendyol \(contribution\)](#)

<https://github.com/Trendyol/go-dcp-kafka>

The Go implementation of the Couchbase to Kafka with DCP.

## [e-cinema](#)

<https://github.com/halilkocaoz/e-cinema>

For practicing microservice basics. The solution helps movie directors to distribute the movies. Use case:

- Movie service POST /movies > inserts new movie to MongoDB, and publishing the MovieCreatedEvent. MediatR handles the event and sends the new movie to RabbitMQ Queue.
- Movie service PUT /movies/{id} > updates exist movie, and publishes the MovieUpdatedEvent. MediatR handles the event and sends the updated movie to RabbitMQ Queue.
- Movie House service consumes the two queue and informs the movie houses.

(.NET 8, RabbitMQ, MongoDB, Docker, Mediator pattern with MediatR, RESTful web services, Domain Driven Design, SOLID)

## [kafka-response-log](#)

<https://github.com/halilkocaoz/kafka-response-log>

For tracking the request processing time.

- Web API, /api/products endpoint's for creating dummy data to log. This endpoint has Delayer and TimeTracker. It's mean that every request will wait for a random time before processing at the endpoint and elapsed time to response will be logged by TimeTracker.
- Go Project consumes the kafka messages and writes those messages to the database.

(.NET 5, Go, Postgres, Docker, Kafka)

## [upsmo-server](#)

<https://github.com/halilkocaoz/upsmo-server>

Cloning an uptime monitoring SaaS.

- Web API for uptime monitoring.
- **checker** solution with Go. It runs in different locations on the Azure and checks uptime monitors by region. Also, it takes actions according to the HTTP responses, such as sending messages to the Azure Service Bus.
- **inserter** solution with Go. It listens to service bus messages which are sent to insert into the database by the upsmo-**checker**.

(.NET 5, Go, Postgres, Docker, RESTful web service, Azure Service Bus)

## [cu-yemekhane](#)

<https://github.com/halilkocaoz/cu-yemekhane>

Web API and bots for accessing information about Çukurova Üniversitesi dining hall.

- Web API scraps the Çukurova Üniversitesi dining hall data, and cache it, presents data via API.
- Telegram fetches the data from Web API, it also caches too and presents the data via text messaging by commands.

(.NET 6, Docker, RESTful web service, Heroku)

Find more look at: [halilkocaoz.com/github](https://halilkocaoz.com/github)