Univerzitet u Kragujevcu

Prirodno-Matematički fakultet

Institut za matematiku i informatiku

A blue and silver logo

AI-generated content may be incorrect.

**Rezervacije**

Seminarski rad iz predmeta Programiranje distribuiranih Sistema

**Profesor** Miloš Pavković

**Student**: Aleksandar Golubović 43-2022

Sadrzaj

[Arhitektura projekta 3](#_Toc212126684)

[Implementacija baze 3](#_Toc212126685)

[Dijagram komunikacije 4](#_Toc212126686)

[Tabela servisa 4](#_Toc212126687)

[User-service 4](#_Toc212126688)

[Order-service 5](#_Toc212126689)

[Product controller 6](#_Toc212126690)

# Arhitektura projekta

Projekat je podeljen na četiri modula koji zajedno čine zatvoren sistem mikroservisa, odgovoran za internu komunikaciju između servisa, kao i za interakciju sa klijentom. Moduli su:

* **API-Gateway**
* **Discovery-Service (Eureka Server)**
* **Users-Service**
* Order-Service

**Users-Service** predstavlja mikroservis koji sadrži kompletan CRUD, i vrši validaciju podataka koje dobija. Podaci kojima manipuliše se čuvaju u H2 (In-Memory) bazi.

Order-Service prestavlja mikroservis koji sadrži CRUD i vrši validaciju podataka koje dobije. Svoje podatke cuva u H2 bazi, i u njemu se nalazi deo za ordere i odvojen deo za producte.

**Discovery-Service** predstavlja Eureka server koji sadrži “imenik” svih servisa koji su prijavljenji na taj server, s toga poznaje na kojim adresama-portovima su instance ovih servisa, i može da prosledi tu listu klijentima koji žele da pristupe nekom servisu.

**API-Gateway** služi kao centralna tačka pristupa, on predstavlja “konobara” ovog sistema, čije je glavno zaduženje da opsluži klijenta tako što će se za klijentski zahtev obratiti Discovery-Service modulu, dobiti URL svih mikroservisa i po principu round robin-a rutirati taj zahtev na izabrani servis (load-balancing). U našem slučaju, nemamo višestruke instance mikroservisa, pa će uvek prosledjeni zahtev otići na jedini mikro servis koji postoji za taj zahtev.

# Implementacija baze

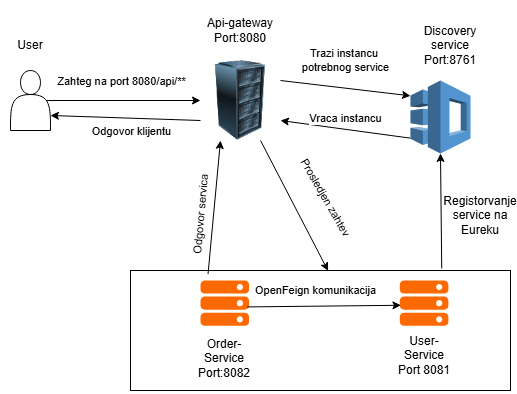
**Users** (id,firstName,lastName,email,phone)

**Products**(id,name,description,price)

**Orders** (id,UserId,ProductId,quantity)

U tabeli orders UserId i ProductId su eksterni kljucevi na tabele users i products.

# Dijagram komunikacije



# Tabela servisa

User-service port 8081 User-Controller bazna ruta „/users“

| **Metoda** | **Endpoint** | **Odgovornost** |
| --- | --- | --- |
| **GET** | /users | Fetchovanje liste svih UserDto objekata. |
| **POST** | /users/add | Dodavanje novog korisnika u sistem. Prima UserDto u telu zahteva. |
| **DELETE** | /users/delete/{email} | Brisanje korisnika iz baze na osnovu prosleđenog **email**-a. |
| **PATCH** | /users/change-phone | Promena broja telefona korisnika. Prima UserChangePhone DTO u telu zahteva. |
| **GET** | /users/{email} | Fetchovanje podataka korisnika na osnovu prosleđenog **email**-a. |
| **GET** | /users/id/{id} | Fetchovanje UserDto objekta za prosleđeni **ID** korisnika. |

Order-service port8082 Order-controller bazna ruta „/orders“

| **Metoda** | **Endpoint** | **Odgovornost** |
| --- | --- | --- |
| **POST** | /orders/add | Kreiranje i dodavanje nove porudžbine (Order) u sistem. Prima **OrderDto** u telu zahteva. |
| **GET** | /orders | Fetchovanje liste svih porudžbina (Order) iz baze. |
| **GET** | /orders/{id}/details | Fetchovanje detalja porudžbine na osnovu prosleđenog **ID**-a. |
| **DELETE** | /orders/{id} | Brisanje porudžbine iz baze na osnovu prosleđenog **ID**-a. |
| **PATCH** | /orders/{id}/quantity | Ažuriranje količine (quantity) stavke u porudžbini. Koristi **ID** iz putanje i **quantity** kao parametar upita (@RequestParam). |

Product controller u sklopu order-service bazna ruta „/products“

| **Metoda** | **Endpoint** | **Odgovornost** |
| --- | --- | --- |
| **POST** | /products/add | Kreiranje i dodavanje novog **proizvoda** u sistem. Prima **ProductDto** u telu zahteva. |
| **GET** | /products | Fetchovanje liste svih **ProductDto** objekata. |
| **GET** | /products/{id} | Fetchovanje podataka o proizvodu na osnovu prosleđenog **ID**-a. |
| **DELETE** | /products/delete/{id} | Brisanje proizvoda iz baze na osnovu prosleđenog **ID**-a. |
| **PATCH** | /products/{id}/price | Ažuriranje **cene** proizvoda. Koristi **ID** iz putanje i prima **ProductPriceDto** u telu zahteva. |
| **GET** | /products/name/{name} | Fetchovanje podataka o proizvodu na osnovu prosleđenog **imena**. |

Swagger je podignut na na portovima 8082 i 8081.