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**: Luka Mladićević 69/2022

Sadržaj

[Obavezni deo 3](#_Toc211808709)

[Arhitektura 3](#_Toc211808710)

[Implementacija baze 4](#_Toc211808711)

[Dijagram komponenti 5](#_Toc211808712)

[Tabela servisa 6](#_Toc211808713)

[Users-Service 6](#_Toc211808714)

[Booking-Service 6](#_Toc211808715)

# Obavezni deo

# Arhitektura

Projekat je rasporedjen na četiri modula, oni čine zatvoreni mrežni sistem koji je zadužen za kako medjusobnu komunikaciju tako i za komunikaciju sa klijentom. Ova četiri modula su:

* API-Gateway
* Discovery-Service (Eureka server)
* Users-Service
* Booking-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.

**Booking-Service** predstavlja mikroservis, koji sadrži sve funkcionalnosti Users-Service-a ali i dodatno ima mogućnost slanja zahteva Users-Service-u putem OpenFeign-a, pored toga realizuje i agregacioni endpoint putem OpenFeign-a.

**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.

Users-Service i Booking-Service su implementirani prateći MVC arhitekturu.

## Implementacija baze

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

Svaki atribut ima odgovarajući prohtev za validaciju.

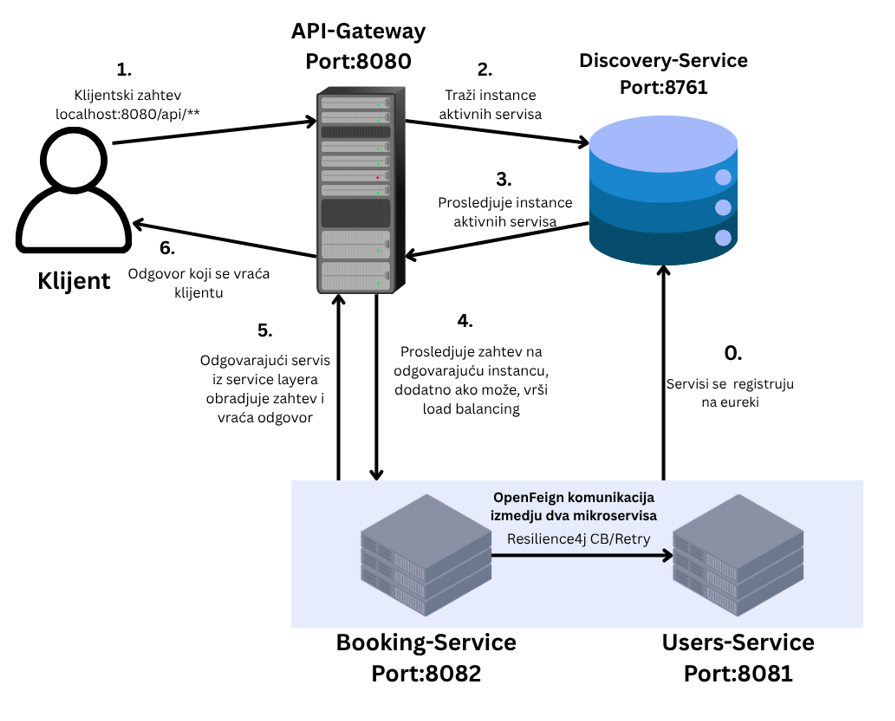
* bookings(*id*,accomodation,accomodationType,startAt,endAt)

Svaki atribut ima odgovarajući prohtev za validaciju, **accomodationType** je ENUM (HOTEL,AIRBNB,APARTMENT).

* reservations(*booking\_id*,booking,user\_id,createdAt)

Atribut booking sluzi kao referenca na BookingAccomodation za dati booking\_id, a booking\_id je ujedno i PK i FK za ovu tabelu. Ovo nam obezbedjuje da može postojati tačno jedna rezervacija za svaki booking, a da pritom jedan user može imati više distinktnih rezervisanih bookinga.

# Dijagram komponenti



# Tabela servisa

## Users-Service

Realizovan na port-u **8081.**

Rute servisa:

|  |  |  |
| --- | --- | --- |
| **user-controller** | | |
| **method** | **endpoint** | **odgovornost** |
| **GET** | **/users** | **Fetchovanje svih UserDTO objekata.** |
| **POST** | **/users** | **Dodavanje novog korisnika.** |
| **PATCH** | **/users/email** | **Promena email-a korisnika.** |
| **GET** | **/users/{id}** | **Fetchovanje UserDTO objekta za prosledjeni id.** |
| **DELETE** | **/users/{id}** | **Brisanje korisnika iz baze.** |
| **GET** | **/users/names** | **Fetchovanje imena svih korisnika u bazi.** |

## Booking-Service

Realizovan na port-u **8082.**

Rute servisa:

|  |  |  |
| --- | --- | --- |
| **booking-controller** | | |
| **method** | **endpoint** | **odgovornost** |
| **GET** | **/booking** | **Fetchovanje imena svih smeštaja iz baze.** |
| **POST** | **/booking** | **Dodavanje novog booking-a.** |
| **PATCH** | **/users/changeDate/{id}** | **Promena pocetnog i zavrsnog datuma za prosledjeni id bookinga.** |
| **GET** | **/booking/{id}** | **Fetchovanje**  **BookingAccomodationDTO za prosledjeni id.** |
| **DELETE** | **/users/{id}** | **Brisanje bookinga iz baze.** |
| **reservation-controller** | | | |
| **method** | | **endpoint** | **odgovornost** |
| **POST** | | **/reservations/reserve** | **Rezervisanje bookinga za proslednjeni bookingID i prosledjen userID.** |
| **GET** | | **/reservations** | **Fetchovanje svih rezervacija za sve korisnike.** |
| **GET** | | **/reservations/{id}/user** | **Fetchovanje svih rezervacija za prosledjen userID.** |

Napomena:

Service layout(booking i users) imaju postavljen dependency za Swagger-UI, tako da je moguće sve ove pozive lako i pregledno uraditi koristeći Swagger-UI. (Portovi 8081 i 8082)