Kratak opis dela sistema

U okviru ovog dela projekta se nalaze ukupno 3 mikroservisa : gateway, authorization i logger.

1. Gateway

Gateway koristimo kao način za pristup korisnika našem sistemu i ovde korisničke zahteve preusmeravamo na odgovarajuće servise. Takođe ovde se vrši i autentifikacija korisničkih jwt tokena i validacija tipova naloga koje imaju kako bi ograničili pristup određenim delovima sistema. Autentifikacija se vrši samo jednom ovde ostali servisi ne vrše autentifikaciju jwt tokena. U gateway-u koristimo AddCommunicationKeyMiddleware kako bi pridodali tokene za komunikaciju na korisničke zahteve.

* 1. Autorizacija korisnika – <https://localhost:5005/api/login/user> POST

Preusmeravanje zahteva na <https://localhost:5001/api/authorize/user>

* 1. Autorizacija admina – <https://localhost:5005/api/login/admin> POST

Preusmeravanje na <https://localhost:5001/api/authorize/user>

* 1. Vracanje logova – <https://localhost:5005/api/logger> GET

Preusmerava na <https://localhost:5003/api/log/> ako je prosleđeni jwt token validan i ako u svojem telu ima admin vrednost atributa Role(samo admin može da vidi logove).

1. Authorization – [Swagger UI](https://localhost:5001/swagger/index.html) (potreban ključ za komunikaciju za poziv endpoint-a)

Authorization koristimo kako bi dodelili korisnicima jwt tokene i role za pristup ostalim delovima sistema. Ovde se vrši provera kredencijala i komunikacija sa user mikroservisom kao i dodela jwt tokena. Koristimo CommunicationKeyAuthMiddleware koji proverava token za komunikaciju u dolazećim zahtevima kako bi znali da nam zahtev šalje naš servis.

* 1. Autorizacija korisnika – <https://localhost:5001/api/authorize/user> POST

Korisnik šalje kredencijale(korisničko ime i lozinka) a vraća se jwt token ovog korisnika ako su validni kredencijali

* 1. Autorizacija admina – <https://localhost:5001/api/authorize/admin> POST

Admin šalje kredencijale(korisničko ime i lozinka) a vraća se jwt token ovog admina ako su validni kredencijali

1. Logger – [Swagger UI](https://localhost:5003/swagger/index.html) (potreban ključ za komunikaciju za poziv endpoint-a)

Logger mikroservis služi za logovanje aktivnosti na našim servisima. Koristimo MongoDB za čuvanje naših logova.

* 1. Logovanje – <https://localhost:5003/api/log/> POST

Kreiranje logova

* 1. Vracanje logova - <https://localhost:5003/api/log/> GET

Vraća logove koji zadovoljavaju prosleđene parametre. Ako se nijedan parametar ne pošalje vraćaju se svi logovi iz baze.

Parametri koje možemo proslediti su sledeći:

1.Microservice – Naziv mikroservisa koji je izvršio logovanje

2.LogLevel – Nivo logovanja moguće vrednosti su Information, Error i Warning

3.from – Datum i vreme od kada želimo logove

4.to – Datum i vreme do kada želimo logove

# Pomoćne biblioteke

Takođe u okviru ovog projekta su napravljene 2 biblioteke radi konzistentnog logovanja i autentifikacije komunikacije izmedju servisa :

1. CommunicationKeyAuthClassLibrary.dll

U ovoj biblioteci se nalaze dva middleware-a (uzima token iz konfiguracionog file-a Configuraion[“CommunicationKey:Key”] ):

* + - 1. AddCommunicationKeyMiddleware – Middleware koji dodaje token za komunikaciju
      2. CommunicationKeyAuthMiddleware – Middleware koji proverava token u header-u i vraća status 401 ako nije validan

1. LoggingClassLibrary.dll

U ovoj biblioteci se nalaze logger provider i sam logger putem kojeg logujemo:

* + - 1. LoggerProvider – Nasleđuje ILoggerProvider
      2. Logger – Nasleđuje ILogger i ima dodatnu funkciju Log koja prima LogLevel,requestID,previousRequestID,message i exception koju koristimo prilikom „ručnog“ logovanja