# URIS - Kratak Opis Sistema

## User mikroservis

U ovom delu projekta odrađen je User mikroservis koji je opisan u sledecem tekstu

## User mikroservis

User mikroservis predstavlja servis koji rukuje korisnicima, ulogama, takodje dodata je i tabela zemlja u kojoj se nalaze naši korisnici (ova tabela je dodata zato što sam smatrao da je za moj servis premalo odraditi samo za korisnike i uloge). Korisnici su podeljeni na personalne i korporativne.

**Personalni nalog:**

Personalni korisnici sadrže: identifikaciono obeležje userId, ime, prezime, email, da li je nalog aktivan, username, telephone, identifikaciono obeležje uloge, i identifikaciono obeležje zemlje u kojoj se nalaze.

Ovaj mikroservis omogućava kreiranje personalnih naloga, izmenu, brisanje i pregled svih personalnih naloga.

* Get metoda:   
  - /api/personalUsers?[country, username] - Vraća sve personalne naloge, parametri su opcioni  
  - /api/personalUsers/{userId} - Vraća personalni nalog sa odredjenim id-om
* Post metoda:   
  - /api/personalUsers - Kreiramo personalni nalog. U body delu metode se specificira objekat bez identifikacionog obeležja.   
  - /api/personalUsers/admins - Kreiranje personalnog naloga sa ulogom admina.
* Put metoda:   
  - api/personalUsers/{userId} - Menjamo personalni nalog sa odredjenim id-om.
* Delete metoda:   
  - api/personalUsers/{userId} - Brisemo personalni nalog sa odredjenim id-om.

**Korporativni nalog:**

Korporativni korisnici sadrže: identifikaciono obeležje userId, ime korporacije, email, da li je nalog aktivan, username, telephone, grad u kome se korporacija nalazi, adresa korporacije, identifikaciono obeležje uloge, i identifikaciono obeležje zemlje u kojoj se nalazi korporacija.

Ovaj mikroservis omogućava kreiranje korporativnih naloga, izmenu, brisanje i pregled svih korporativnih naloga.

* Get metoda:   
  - /api/corporationlUsers?[country, username] - Vraća sve korporativne naloge, parametri su opcioni  
  - /api/corporationlUsers/{userId} - Vraća korporativni nalog sa odredjenim id-om
* Post metoda:   
  - /api/corporationUsers - Kreiramo korporativni nalog. U body delu metode se specificira objekat bez identifikacionog obeležja.
* Put metoda:   
  - api/corporationUsers/{userId} - Menjamo korporativni nalog sa odredjenim id-om.
* Delete metoda:   
  - api/corporationUsers/{userId} - Brisemo korporativni nalog sa odredjenim id-om.

**Država:**

Država sadrži identifikaciono obeležje zemlje i naziv države.

Ovaj mikroservis omogućava kreiranje država, izmenu, brisanje i pregled svih država.

* Get metoda:   
  -/api/countries?[countryName] - Vraća sve države, parametri su opcioni  
  - /api/countries/{countryId} - Vraća državu sa odredjenim id-om
* Post metoda:   
  - /api/countries - Kreiramo državu. U body delu metode se specificira objekat bez identifikacionog obeležja. Vraća nam model države koju kreiramo.
* Put metoda:   
  - api/countries/{countryId} - Menjamo državu sa odredjenim id-om.
* Delete metoda:   
  - api/countries/{userId} - Brisemo državu sa odredjenim id-om.

**Uloga:**

Uloga sadrži identifikaciono obeležje uloge i naziv uloge.

Ovaj mikroservis omogućava kreiranje država, izmenu, brisanje i pregled svih država.

* Get metoda:   
  -/api/roles?[roleName] - Vraća sve uloge, parametri su opcioni  
  - /api/roles/{roleId} - Vraća ulogu sa odredjenim id-om
* Post metoda:   
  - /api/roles- Kreiramo ulogu. U body delu metode se specificira objekat bez identifikacionog obeležja. Vraća nam model uloge koju kreiramo.
* Put metoda:   
  - api/roles/{roleId} - Menjamo ulogu sa odredjenim id-om.
* Delete metoda:   
  - api/roles/{roleId} - Brisemo državu sa odredjenim id-om.

# Pomoćne biblioteke

Takođe, u okviru ovog projekta korišćene su dve biblioteke radi konzistentnog logovanja i autentifikacije komunikacije izmedju servisa:

* CommunicationKeyAuthClassLibrary.dll

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

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

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

* LoggerProvider – Nasleđuje IloggerProvider.
* Logger – Nasleđuje ILogger i ima dodatnu funkciju Log koja prima LogLevel, requestID, previousRequestID, message i exception koju koristimo prilikom logovanja.