PostgreSQL container rba-postgres

docker-compose up -d

Baza podataka se zove ‘**rba**’ iako tako inače nebih nazvao, ali da se mogu lakše snaći zbgo drugih otvorenih containera na poslu ovo mi je bilo najpraktičnije

Koristim sequence Long id umjesto npr. Auto incrementa strategije. Sequence objekti nisu podržani u svim bazama no omogućuju veću kontrola generiranih id-eva. Također sequence strategia omogućava JPA batch execution.

Stavljam anotacije tipa koje nisu obavezne na zadnjim verzijama springa tipa @Autowired na konstruktor (kontruktor Dependency injection je preferirani način od strane ekipe koja je radila spring, DI putem fielda I anotacije se koristi prilikom testiranja) ili @Repository na spring data jpa interface radi preglednosti koda I jasnog definiranja čemu što služi

Status kreditne kartice je napravljen kao entity, a ne kao enum radi lakšeg održavanja, refactoriranja, a opet ima safety ako i enum.

Delete je napravljen tako da kada se jednom obriše, zaista se obriše iz baze podataka. Možda bismo mogli dodati opciju tipa soft delete-a ili neki archieve, da se u tablicu doda DeleteStatus pa kada se obriše se stavi pod “deleted” ili “arhieved” te ako bi se htjelo povratiti usera ili vidjeti obrisane(arhivirane, neakitvne) usere/klijente bi ih mogli pretraziti putem tog atributa.

S obzirom da validacija OIB-a nije kompleksna korištena spring buit in validation framework umjesto AOP-a.

Create new user Success

A screenshot of a computer

Description automatically generated

Create new user fail because wrong card status name

A screenshot of a computer

Description automatically generated

Create new user Fail because not valid OIB too short

A screenshot of a computer

Description automatically generated

Create new user fail because OIB already exists

A screenshot of a computer

Description automatically generated

Get User by OIB exception because user with oib not found

A screenshot of a computer

Description automatically generated

Delete user by OIB

A screenshot of a computer

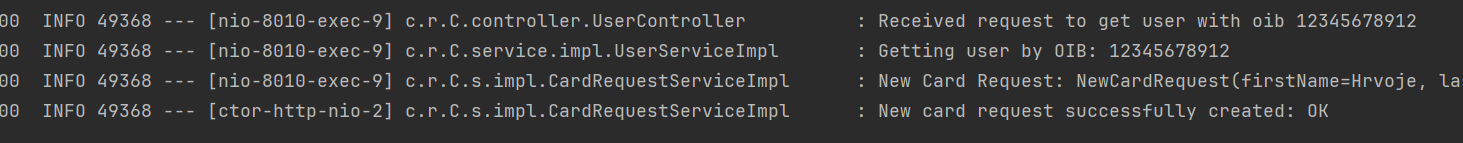
Description automatically generated

Get user by OIB and forward to Card Creation API – I used some small app with test endpoint to test this locally  
A screen shot of a computer code

Description automatically generated

A screen shot of a computer

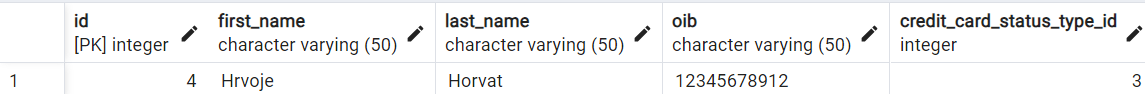
Description automatically generated



A screenshot of a computer

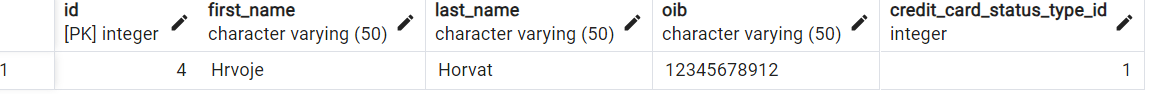
Description automatically generated

Having user with OIB 12345678912 and credit card in status Pending, app receives new Kafka message – update card status for user with OIB 12345678912



A screenshot of a computer

Description automatically generated



A screen shot of a computer program

Description automatically generated

Kafka message card status but OIB does not exists

A screenshot of a computer

Description automatically generated

But in app there will be logged exceptions