Creating MySQL DB containers for microservices

Remove these dependencies from configserver microservice  
D:\Experiments\Microservices\sb-bank-application\configserver\pom.xml  
<dependency>

            <groupId>org.springframework.cloud</groupId>

            <artifactId>spring-cloud-starter-bus-amqp</artifactId>

</dependency>

<dependency>

            <groupId>org.springframework.cloud</groupId>

            <artifactId>spring-cloud-config-monitor</artifactId>

            <version>4.1.2</version>

</dependency>

Remove these properties from configserver  
D:\Experiments\Microservices\sb-bank-application\configserver\src\main\resources\application.properties  
spring.cloud.rabbitmq.host=localhost

spring.cloud.rabbitmq.port=5672

spring.cloud.rabbitmq.username=guest

spring.cloud.rabbitmq.password=guest

Delete this dependency from accounts microservice

D:\Experiments\Microservices\sb-bank-application\accounts\pom.xml

<dependency>

            <groupId>org.springframework.cloud</groupId>

            <artifactId>spring-cloud-starter-bus-amqp</artifactId>

</dependency>

Remove these properties from the accounts  
D:\Experiments\Microservices\sb-bank-application\accounts\src\main\resources\application.properties

spring.rabbitmq.host=localhost

spring.rabbitmq.port=5672

spring.rabbitmq.username=guest

spring.rabbitmq.password=guest

Delete this dependency from loans microservice

D:\Experiments\Microservices\sb-bank-application\loans\pom.xml

<dependency>

            <groupId>org.springframework.cloud</groupId>

            <artifactId>spring-cloud-starter-bus-amqp</artifactId>

</dependency>

Remove these properties from the loans  
D:\Experiments\Microservices\sb-bank-application\loans\src\main\resources\application.properties

spring.rabbitmq.host=localhost

spring.rabbitmq.port=5672

spring.rabbitmq.username=guest

spring.rabbitmq.password=guest

Delete this dependency from cards microservice

D:\Experiments\Microservices\sb-bank-application\cards\pom.xml

<dependency>

            <groupId>org.springframework.cloud</groupId>

            <artifactId>spring-cloud-starter-bus-amqp</artifactId>

</dependency>

Remove these properties from the cards  
D:\Experiments\Microservices\sb-bank-application\cards\src\main\resources\application.properties

spring.rabbitmq.host=localhost

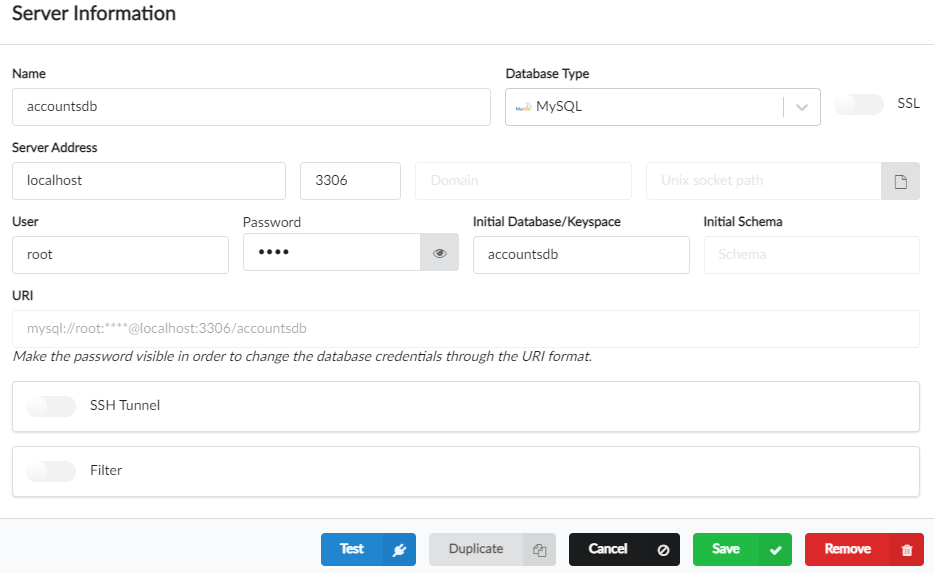
spring.rabbitmq.port=5672

spring.rabbitmq.username=guest

spring.rabbitmq.password=guest

Starting the MySQL docker container in local machine

PS D:\> docker run -p 3306:3306 --name accountsdb -e MYSQL\_ROOT\_PASSWORD=root -e MYSQL\_DATABASE=accountsdb -d mysql

To connect to the DB use a very super light client named SQLECTRON  
https://sqlectron.github.io/  
  


This database is empty does not have any tables.

Similarly create database for loans and cards microservice

PS D:\> docker run -p 3307:3306 --name loansdb -e MYSQL\_ROOT\_PASSWORD=root -e MYSQL\_DATABASE=loansdb -d mysql

PS D:\> docker run -p 3308:3306 --name cardsdb -e MYSQL\_ROOT\_PASSWORD=root -e MYSQL\_DATABASE=cardsdb -d mysql

Note: Restart the Sqlectron after creating the database for successful connection

Update microservices code to replace H2 DB with MySQL DB

Delete H2 related dependency from accounts, loans, and cards microservice

<dependency>

        <groupId>com.h2database</groupId>

        <artifactId>h2</artifactId>

        <scope>runtime</scope>

</dependency>

Add MySQL related dependency from accounts, loans, and cards microservice

<dependency>

        <groupId>com.mysql</groupId>

        <artifactId>mysql-connector-j</artifactId>

<scope>runtime</scope>

</dependency>

Modify all the H2 configurations with MySQL configurations in accounts, loans, and cards microservice

D:\Experiments\Microservices\sb-bank-application\accounts\src\main\resources\application.properties

….

spring.datasource.url=jdbc:mysql://localhost:3306/accountsdb

~~# spring.datasource.driver-class-name=org.h2.Driver~~

spring.datasource.username=root

spring.datasource.password=root

~~# spring.jpa.database-platform=org.hibernate.dialect.H2Dialect~~

~~# spring.jpa.hibernate.ddl-auto=update~~

spring.jpa.show-sql=true

spring.sql.init.mode=always

~~# spring.h2.console.enabled=true~~

…

D:\Experiments\Microservices\sb-bank-application\loans\src\main\resources\application.properties

…

spring.datasource.url=jdbc:mysql://localhost:3307/loansdb

~~# spring.datasource.driver-class-name=org.h2.Driver~~

spring.datasource.username=root

spring.datasource.password=root

~~# spring.jpa.database-platform=org.hibernate.dialect.H2Dialect~~

~~# spring.jpa.hibernate.ddl-auto=update~~

spring.jpa.show-sql=true

spring.sql.init.mode=always

~~# spring.h2.console.enabled=true~~

…

D:\Experiments\Microservices\sb-bank-application\cards\src\main\resources\application.properties

…

spring.datasource.url=jdbc:mysql://localhost:3308/cardsdb

~~# spring.datasource.driver-class-name=org.h2.Driver~~

spring.datasource.username=root

spring.datasource.password=root

~~# spring.jpa.database-platform=org.hibernate.dialect.H2Dialect~~

~~# spring.jpa.hibernate.ddl-auto=update~~

spring.jpa.show-sql=true

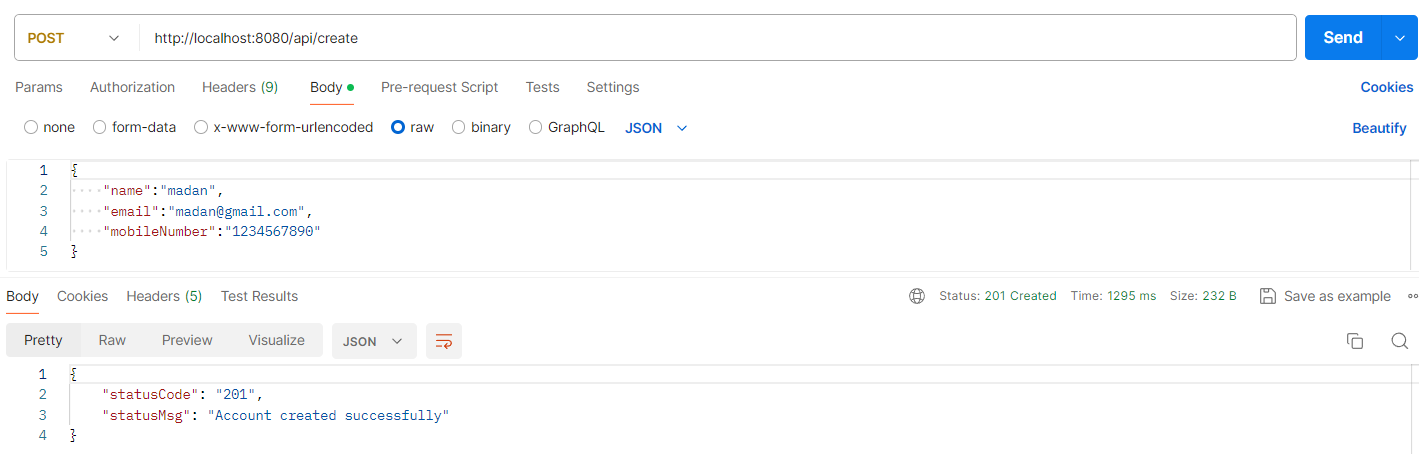
spring.sql.init.mode=always  
~~# spring.h2.console.enabled=true~~

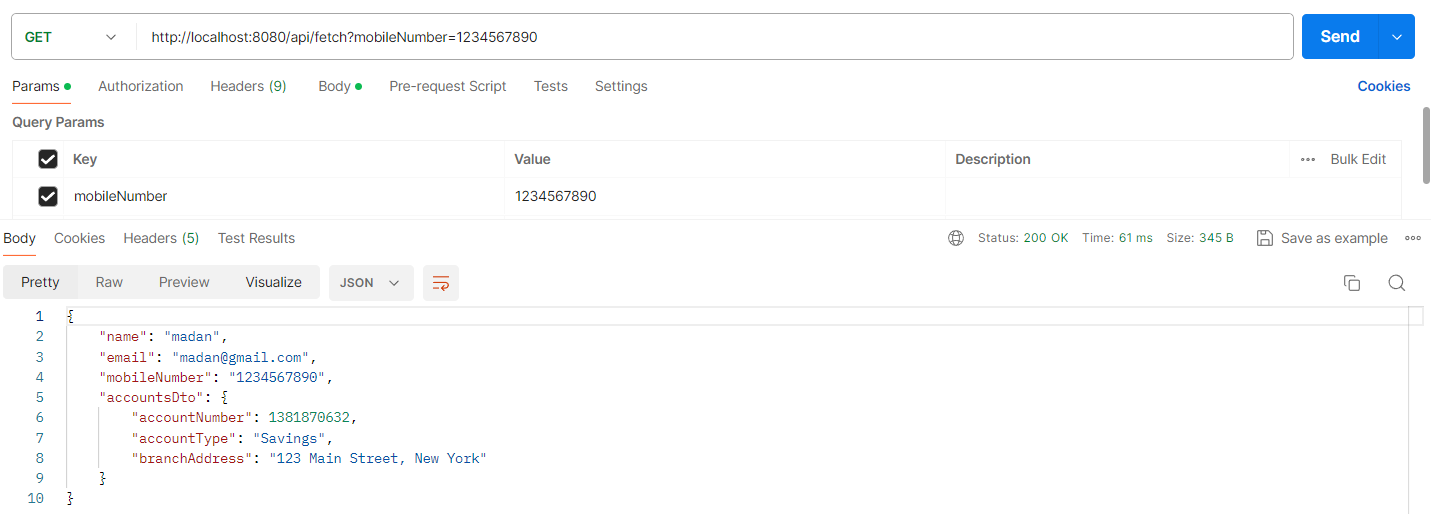
….

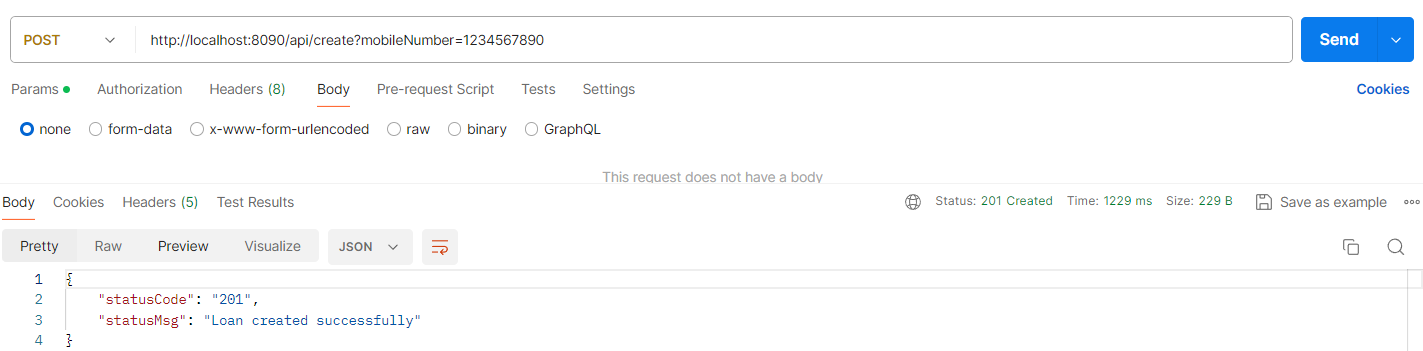
Start the config server  
Start accounts, loans, and cards microservice

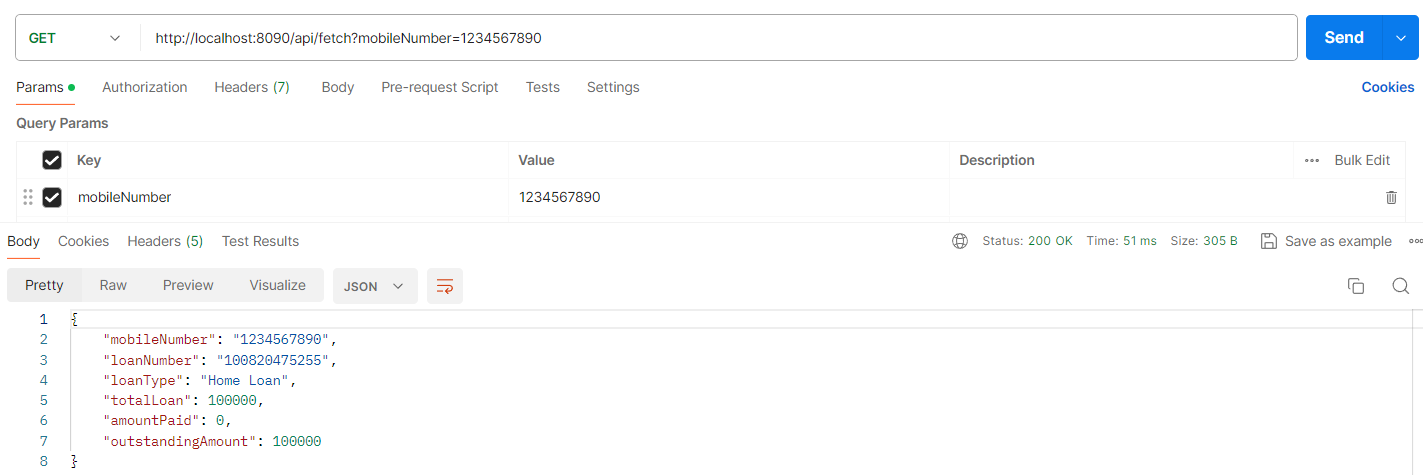
There will be tables created for each of the microservices on starting the service.

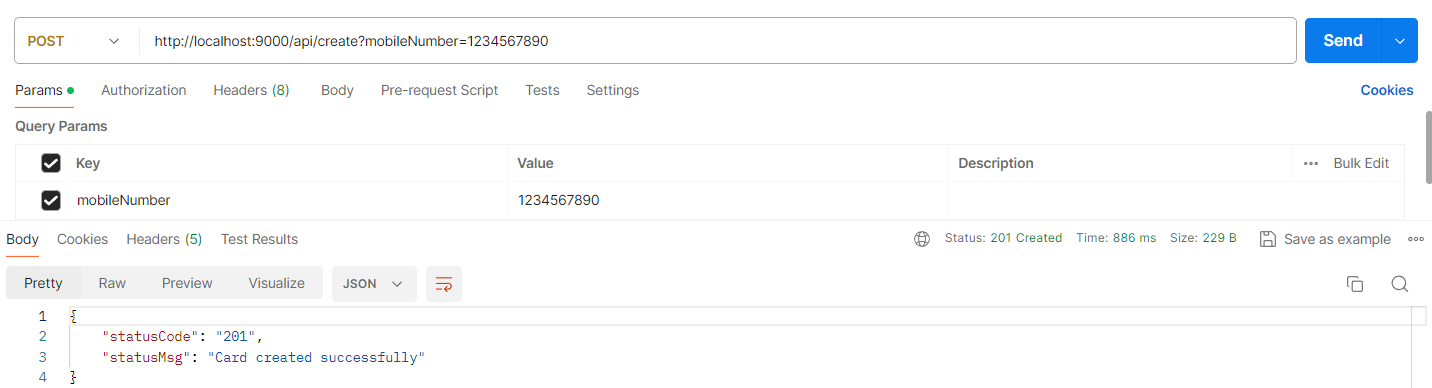
Test the databases using postman

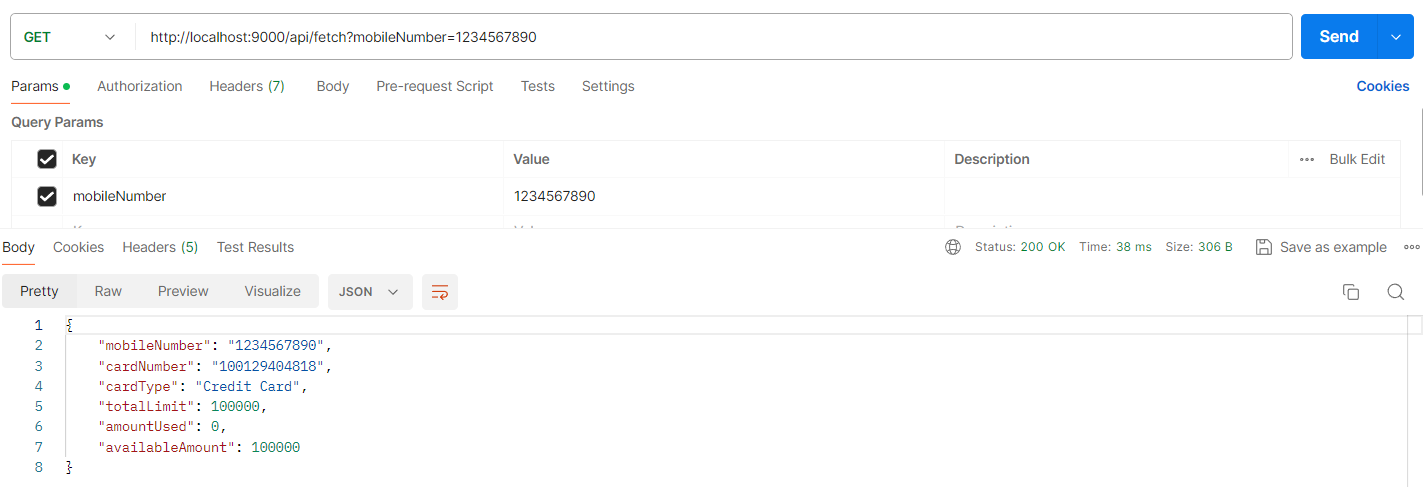


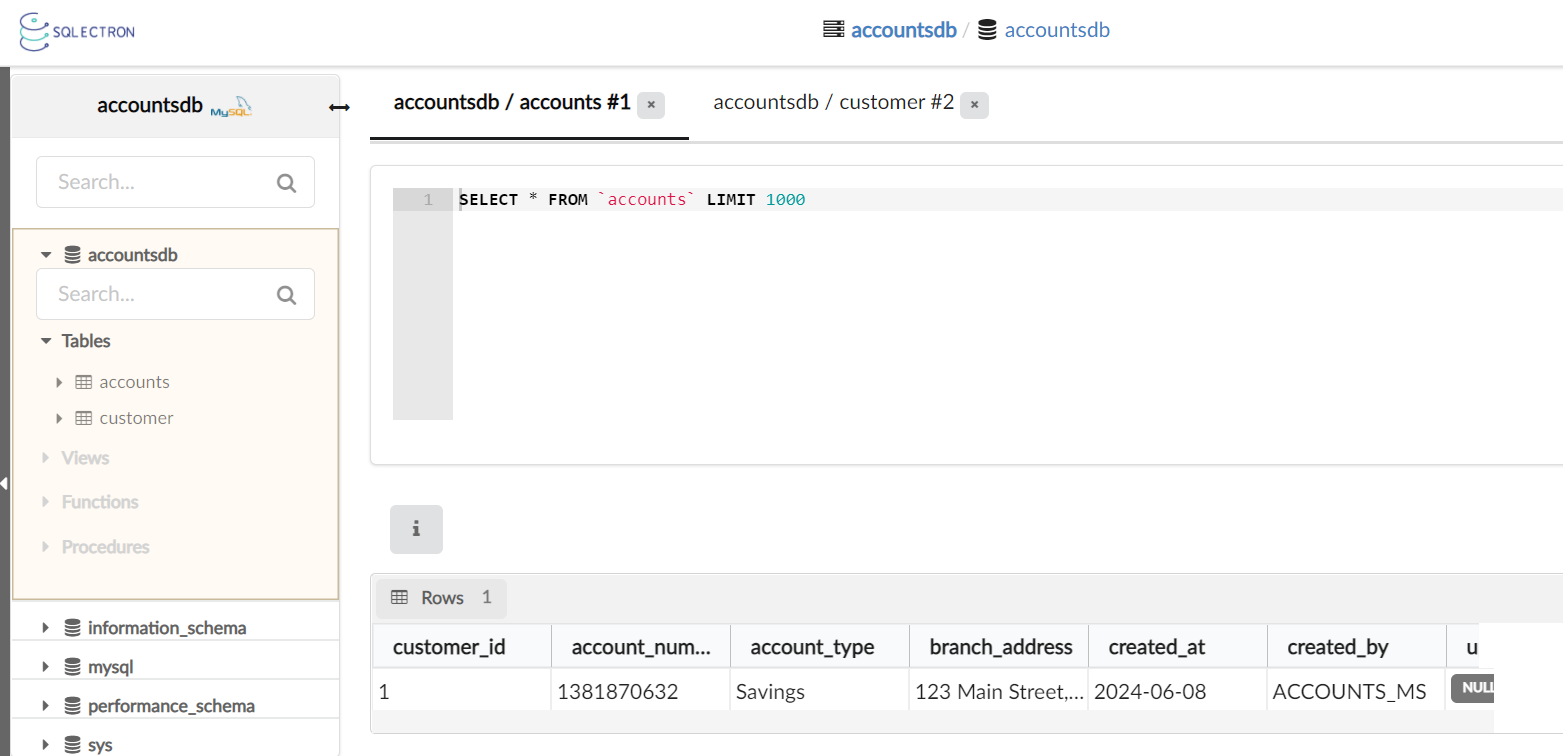












If we stop the container, data will not be deleted, but if we delete the container, all the data will be lost.

Update docker compose file to create and use MySQL DB

Running microservices & MySQL DB containers using docker compose file

Demo of Docker network concept