Demo application shown by Chandra at Spring-microservices meetup.

Make the following changes for the application to

i) run in your local environment

ii) use MySQL database.

i) Run in your local environment

Unzip the source file to a convenient location.

Initiate git and commit in config folder. Commands to run in config folder are:

git init .

git add \*

git commit -m “First commit” .

In config-server/application.yml, change the config folder path to *your* filesystem path, for example:

uri: file:///home/mahboob/Code/java/spring-msa/config

ii) Use MySQL database

Change database in config/application.yml

spring:

database:

driverClassName: com.mysql.jdbc.Driver

datasource:

password: *<<your password>>*

platform: mysql

username: root

url: jdbc:mysql://localhost:3306/msa

jpa:

database: MYSQL

show-sql: true

generate-ddl: true

hibernate:

ddl-auto: none

Change postgres dependency to mysql in three pom.xml files

product-service, rating-service and authorization-server

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<scope>runtime</scope>

</dependency>

Start mysql server

Log into mysql and create the database. I used the name msa

mysql> create database msa;

Start the microservices separately in six terminal windows

cd into config-server, edge-service, discovery-service, authorization-server, product-service, rating-service folders and run the following command

mvn clean spring-boot:run

Calling the services

Open a new terminal window:

Create two products  
curl -XPOST -H "Content-Type:application/json" -d '{"name":"Keyboard", "description":"Wired Keyboard"}' http://localhost:8080/api/products

curl -XPOST -H "Content-Type:application/json" -d '{"name":"Mouse", "description":"Optic Mouse"}' http://localhost:8080/api/products

Get details of all products

curl -XGET http://localhost:8080/api/products

curl -XGET http://localhost:8080/api/products/{id}

Replace {id} with an actual id (int) get from the previous call.

Take the id, say 2

Get details

curl -XGET http://localhost:8080/api/products/2

Post two ratings

curl -XPOST -H "Content-Type:application/json" -d '{"rating":"4.5", "comments":"Excellent product"}' http://localhost:8080/api/ratings/products/2

curl -XPOST -H "Content-Type:application/json" -d '{"rating":"2.5", "comments":"Average product"}' http://localhost:8080/api/ratings/products/2

Postman

These calls can be made via Postman client too. For example, to call rating service for product id 7:

Select POST from left dropdown list.

URL: <http://localhost:8080/api/ratings/products/7>

Click on Body. Click raw. Select JSON(application/json) from the drop-down list on the right.

Enter the following data

{

“rating” : “5.0”,

“Comments” : “Awesome product”

}

Click Send

