**Running Notes:**

# New controller code for feign client

**AccountsController.java**

package com.bank.account.controller;

@RestController

public class AccountsController {

@Autowired

private AccountsRepository accountRepository;

@Autowired

LoansFeignClient loansFeignClient;

@Autowired

CardsFeignClient cardsFeignClient;

@Autowired

AccountsServiceConfig accountsConfig;

@PostMapping("/myaccount")

public Accounts getAccountDetails(@RequestBody Customer customer) {

Accounts account = accountRepository.findByCustomerId(customer.getCustomerId());

if (account != null) {

return account;

} else {

return null;

}

}

@GetMapping("/account/properties")

public String getPropertyDetails() throws JsonProcessingException {

ObjectWriter ow = new ObjectMapper().writer().withDefaultPrettyPrinter();

Properties properties = new Properties(accountsConfig.getMsg(), accountsConfig.getBuildVersion(),

accountsConfig.getMailDetails(), accountsConfig.getActiveBranches());

String jsonStr = ow.writeValueAsString(properties);

return jsonStr;

}

@PostMapping("/myCustomerDetails")

public CustomerDetails myCustomerDetails(@RequestBody Customer customer) {

Accounts accounts = accountRepository.findByCustomerId(customer.getCustomerId());

List<Loans> loans = loansFeignClient.getLoansDetails(customer);

List<Cards> cards = cardsFeignClient.getCardDetails(customer);

CustomerDetails customerDetails = new CustomerDetails();

customerDetails.setAccounts(accounts);

customerDetails.setLoans(loans);

customerDetails.setCards(cards);

return customerDetails;

}

**To build docker images:**

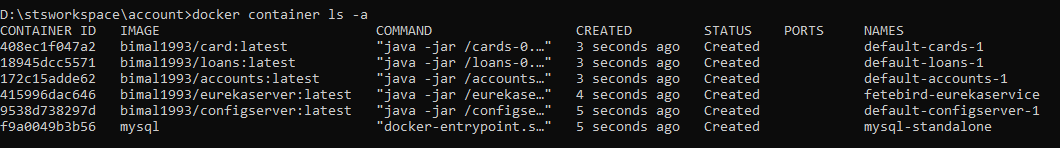
* docker build . -t bimal1993/accounts
* docker build . -t bimal1993/loans
* docker build . -t bimal1993/card
* docker build . -t bimal1993/configserver
* docker build . -t bimal1993/eurekaserver

docker run --network spring-net -p 8080:8081 bimal1993/accounts

docker run --network spring-net -p 8090:8090 bimal1993/loans

docker run --network spring-net -p 9000:9000 bimal1993/card

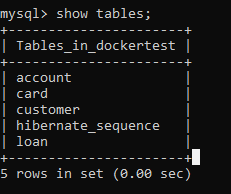
D:\stsworkspace\account\docker-compose\default>docker compose up



docker container exec -it be18f2a16232 bash

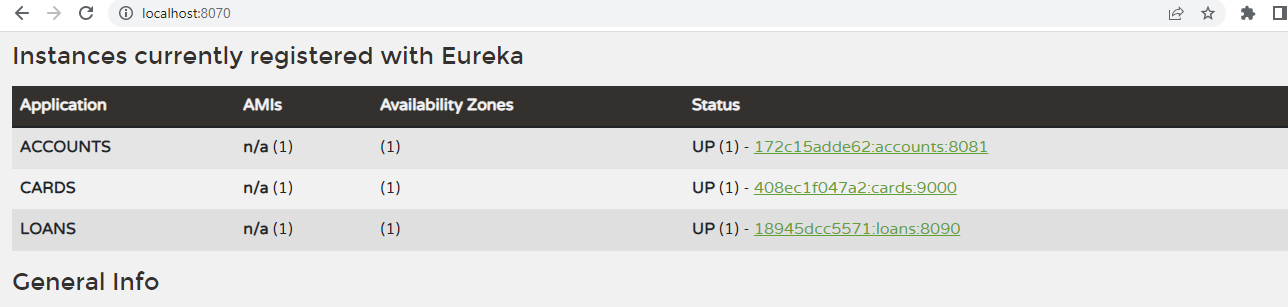
mysql -uroot -p

1234



**To check docker log**

D:\stsworkspace\loans>docker logs -f e0f509001613



docker-compose.yml

version: "3.8"

services:

mysql-standalone:

image: mysql

container\_name: mysql-standalone

networks:

- spring-net

environment:

- MYSQL\_ROOT\_PASSWORD=1234

- MYSQL\_DATABASE=dockertest

configserver:

image: bimal1993/configserver:latest

mem\_limit: 700m

ports:

- "8071:8071"

networks:

- spring-net

eurekaserver:

image: bimal1993/eurekaserver:latest

container\_name: fetebird-eurekaservice

mem\_limit: 700m

ports:

- "8070:8070"

networks:

- spring-net

depends\_on:

- configserver

deploy:

restart\_policy:

condition: on-failure

delay: 25s

max\_attempts: 10

window: 120s

environment:

SPRING\_PROFILES\_ACTIVE: default

SPRING\_CONFIG\_IMPORT: configserver:http://configserver:8071/

accounts:

image: bimal1993/accounts:latest

mem\_limit: 700m

ports:

- "8080:8081"

networks:

- spring-net

depends\_on:

- eurekaserver

- mysql-standalone

deploy:

restart\_policy:

condition: on-failure

delay: 60s

max\_attempts: 10

window: 120s

environment:

SPRING\_PROFILES\_ACTIVE: default

SPRING\_CONFIG\_IMPORT: configserver:http://configserver:8071/

loans:

image: bimal1993/loans:latest

mem\_limit: 700m

ports:

- "8090:8090"

networks:

- spring-net

depends\_on:

- eurekaserver

- mysql-standalone

deploy:

restart\_policy:

condition: on-failure

delay: 60s

max\_attempts: 10

window: 120s

environment:

SPRING\_PROFILES\_ACTIVE: default

SPRING\_CONFIG\_IMPORT: configserver:http://configserver:8071/

cards:

image: bimal1993/card:latest

mem\_limit: 700m

ports:

- "9000:9000"

networks:

- spring-net

depends\_on:

- eurekaserver

- mysql-standalone

deploy:

restart\_policy:

condition: on-failure

delay: 60s

max\_attempts: 10

window: 120s

environment:

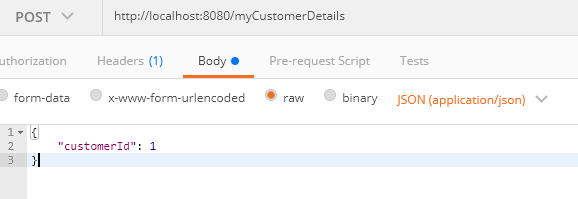
SPRING\_PROFILES\_ACTIVE: default

SPRING\_CONFIG\_IMPORT: configserver:http://configserver:8071/

networks:

spring-net:

Request:



{

"accounts": {

"customerId": 1,

"accountNumber": 186576453,

"accountType": "Savings",

"branchAddress": "123 Main Street, New York",

"createDt": "2022-07-09"

},

"loans": [

{

"loanNumber": 3,

"customerId": 1,

"startDt": "2021-02-14",

"loanType": "Home",

"totalLoan": 50000,

"amountPaid": 10000,

"outstandingAmount": 40000,

"createDt": "2018-02-14"

},

{

"loanNumber": 1,

"customerId": 1,

"startDt": "2020-10-13",

"loanType": "Home",

"totalLoan": 200000,

"amountPaid": 50000,

"outstandingAmount": 150000,

"createDt": "2020-10-13"

},

{

"loanNumber": 2,

"customerId": 1,

"startDt": "2020-06-06",

"loanType": "Vehicle",

"totalLoan": 40000,

"amountPaid": 10000,

"outstandingAmount": 30000,

"createDt": "2020-06-06"

},

{

"loanNumber": 4,

"customerId": 1,

"startDt": "2018-02-14",

"loanType": "Personal",

"totalLoan": 10000,

"amountPaid": 3500,

"outstandingAmount": 6500,

"createDt": "2018-02-14"

}

],

"cards": [

{

"cardId": 1,

"customerId": 1,

"cardNumber": "4565XXXX4656",

"cardType": "Credit",

"totalLimit": 10000,

"amountUsed": 500,

"availableAmount": 9500,

"createDt": "2022-07-09"

},

{

"cardId": 2,

"customerId": 1,

"cardNumber": "3455XXXX8673",

"cardType": "Credit",

"totalLimit": 7500,

"amountUsed": 600,

"availableAmount": 6900,

"createDt": "2022-07-09"

},

{

"cardId": 3,

"customerId": 1,

"cardNumber": "2359XXXX9346",

"cardType": "Credit",

"totalLimit": 20000,

"amountUsed": 4000,

"availableAmount": 16000,

"createDt": "2022-07-09"

}

]

}

<http://localhost:8071/accounts/default>

{

"name": "accounts",

"profiles": [

"default"

],

"label": null,

"version": "0484f35409d0c637fdf511424fb30646e6607417",

"state": null,

"propertySources": [

{

"name": "https://github.com/Bimal1993/microservice-config.git/accounts.properties",

"source": {

"accounts.msg": "Welcome to the EazyBank Accounts Default application",

"accounts.build-version": "3",

"accounts.mailDetails.hostName": "default-accounts@eazybytes.com",

"accounts.mailDetails.port": "9000",

"accounts.mailDetails.from": "default-accounts@eazybytes.com",

"accounts.mailDetails.subject": "Your Account Details from Eazy Bank Default Environment",

"accounts.activeBranches[0]": "Mumbai",

"accounts.activeBranches[1]": "London",

"accounts.activeBranches[2]": "Washington"

}

}

]

}

<http://localhost:9000/cards/properties>

{

"msg" : null,

"buildVersion" : "3",

"mailDetails" : {

"hostName" : "default-cards@eazybytes.com",

"port" : "9002",

"from" : "default-cards@eazybytes.com",

"subject" : "Your Card Details from Eazy Bank Default Environment"

},

"activeBranches" : [ "Mumbai", "London", "Washington" ]

}