

NEST HACKATHON

Submitted By

CARBON TEAM

ANOOP P M

JOHN CHRISTO

MANU FASIL M

Abstract

This Application is used for transfer money from one account to another accounts using mt103 parse method. When the user send a transfer request this application convert into MT103 Message Then Convert to MX message.

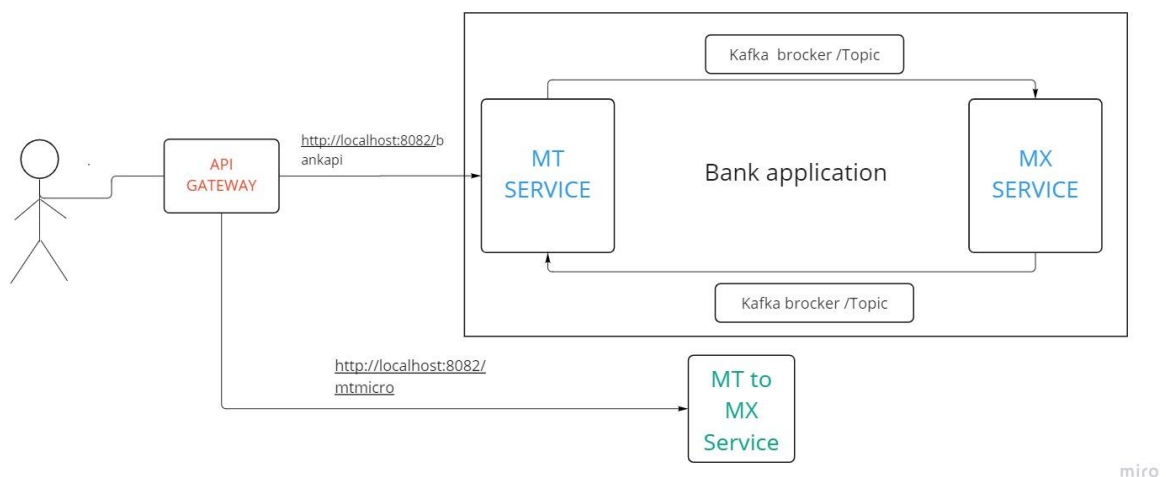
Used Technologies

- Java 8
- Spring boot
- Docker
- Kafka
- Mysql
- Swagger

Used Tools

- Docker Desktop
- Mysql Workbench
- Jmeter
- Spring tool
- Postman

ARCHITECTURE



Project Setup

- Install Java 8
- Set up Kafka Using Docker ,Docker compose file attached in project folder
- Create Kafka topic

```
kafka-topics.sh --create --topic mtmessage --bootstrap-server  
localhost:9092
```

```
kafka-topics.sh --create --topic mxmessage --bootstrap-server  
localhost:9092
```

Listen Kafka Messages

```
kafka-console-consumer.sh --topic mtmessage --from-beginning --  
bootstrap-server localhost:9092
```

```
kafka-console-consumer.sh --topic mxmessage --from-beginning --  
bootstrap-server localhost:9092
```

- Install mysql and give password as root (password = root) and create Database mtbank(DB = mtbank)
- Then Run the 4 Spring boot applications using spring or other tools

Hackathon API Request Examples

Make sure all four API are running successfully

1. Create User

URI = <http://localhost:8082/bankapi/usercreate> (post method)

Json Request body =

```
{  
  "accountnumber": "1720364789995555",  
  "username": "Ajay"  
}
```

2. Create Receivers bank details

URI = <http://localhost:8082/bankapi/addreceiveraccount> (post method)

Json Request body =

```
{
  "accountnumber": "11225546165620",
  "bankName": "SBI",
  "ifsc": "SBIN112445",
  "receivername": "Arshad"
}
```

3. User Deposit

URI = <http://localhost:8082/bankapi/deposit> (put method)

Json Request body =

```
{
  "accountnumber": "1720364789995555"
}
```

4. Transfer amount :this api take user data and create mt103 then send the mt103 message

URI = <http://localhost:8082/bankapi/transfmessage> (post method)

Json Request body =

```
{
  "accountnumber": "790773028412345",
  "address": "SBI",
  "amount": "100",
  "bankname": "SBI",
  "currency": "INR",
  "receiver": "ENF43332",
  "receiverAccountNo": "9539931867123",
  "reference": "CRED",
  "sender": "ENFEES123"
}
```

5. User Balance

URI = <http://localhost:8082/bankapi/userbalance> (Get Method)

Json request body:

```
{
  "accountnumber": "1720364789995555"
}
```

6. Receiver Balance

URI = <http://localhost:8082/bankapi/receiverbalance> (Get Method)

Json request body:

```
{
  "accountnumber": "1244"
}
```

7. MT to MX Converter

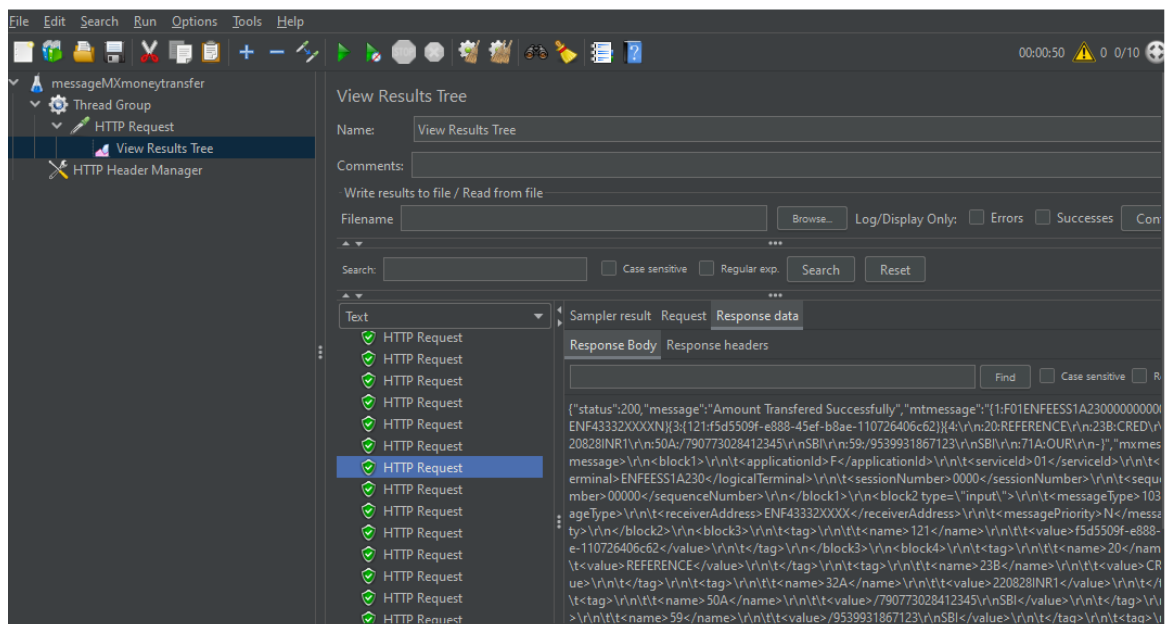
URI = <http://localhost:8082/mtmicro/mttovalue> (Get Method)

Json request body: This message takes from given 103.txt file

```
{
  "message": "{1:F01BICFOOYYAXXX8683497519}{2:01031535051028ESPBSMMAXX54237522470510281535N}{3:{113:ROMF}{108:0510280182794665}{119:STP}}{4:\r\n:20:0061350113089908\r\n:13C:/RNCTIME/1534+0000\r\n:23B:CRED\r\n:23E:SDVA\r\n:32A:061028EUR100000,\r\n:33A:081029EUR120000,\r\n:33B:EUR100000,\r\n:50K:/12345678\r\nAGENTES DE BOLSA FOO AGENCIA\r\nAV XXXXX 123 BIS 9 PL\r\n12345 BARCELONA\r\n:52A:/2337\r\nFOOAESMMXXX\r\n:53A:FOOAESMMXXX\r\n:57A:BICFOOYYXXX\r\n:59:/ES0123456789012345671234\r\nFOO AGENTES DE BOLSA ASOC\r\n:71A:OUR\r\n:72:/BNF/TRANSF. BCO. FOO\r\n-}{5:{MAC:88B4F929}{CHK:22EF370A4073}}"
```

Project Performance

Using JMeter



AGGREGATE REPORT

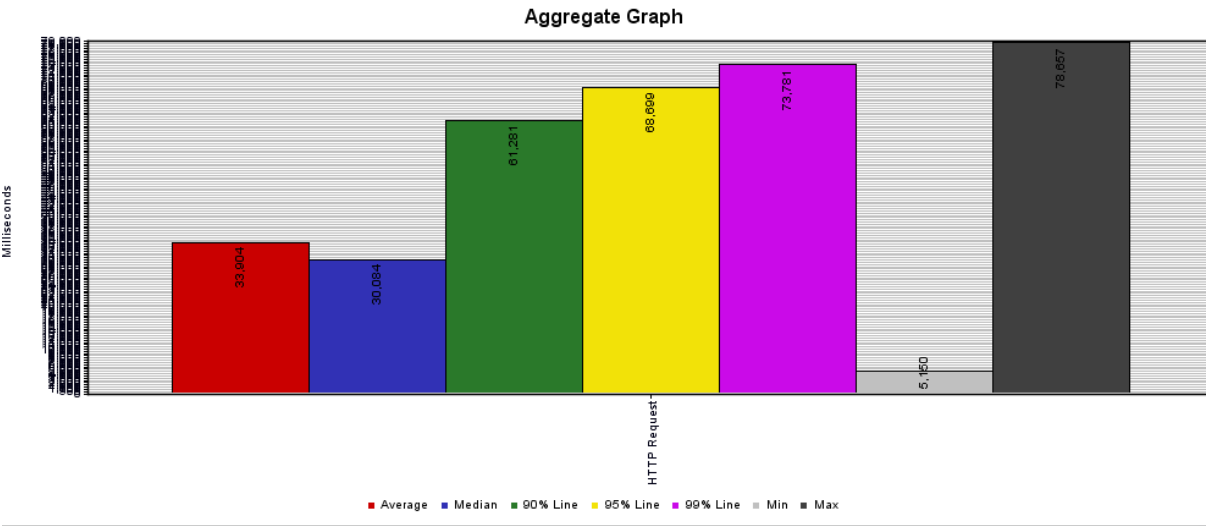
| Filename | | | | | | | | Browse... | Log/Display Only: | | <input type="checkbox"/> Errors | <input type="checkbox"/> Successes | Configure |
|--------------|-----------|---------|--------|----------|----------|----------|------|-----------|-------------------|------------|---------------------------------|------------------------------------|-----------|
| Label | # Samples | Average | Median | 90% Line | 95% Line | 99% Line | Min | Maximum | Error % | Throughput | Received K... | Sent KB/sec | |
| HTTP Requ... | 55 | 33904 | 30084 | 61281 | 68699 | 73781 | 5150 | 78657 | 7.27% | 8.8/min | 0.21 | 0.06 | |
| TOTAL | 55 | 33904 | 30084 | 61281 | 68699 | 73781 | 5150 | 78657 | 7.27% | 8.8/min | 0.21 | 0.06 | |
| | | | | | | | | | | | | | |

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | Received KB/... | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|------|-------|-----------|---------|------------|-----------------|-------------|------------|
| HTTP Request | 10 | 25117 | 5150 | 45104 | 12747.95 | 0.00% | 11.7/min | 0.30 | 0.08 | 1557.0 |
| TOTAL | 10 | 25117 | 5150 | 45104 | 12747.95 | 0.00% | 11.7/min | 0.30 | 0.08 | 1557.0 |
| | | | | | | | | | | |

AGGREGATE Summary

| Label | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | Received KB/... | Sent KB/sec | Avg. Bytes |
|--------------|-----------|---------|------|-------|-----------|---------|------------|-----------------|-------------|------------|
| HTTP Request | 55 | 33904 | 5150 | 78657 | 19239.62 | 7.27% | 8.8/min | 0.21 | 0.06 | 1461.9 |
| TOTAL | 55 | 33904 | 5150 | 78657 | 19239.62 | 7.27% | 8.8/min | 0.21 | 0.06 | 1461.9 |
| | | | | | | | | | | |

Graph



JUNIT TESTING

Junit test for 7 modules

Finished after 13.022 seconds

Runs: 8/8 Errors: 0 Failures: 0

TestRest [Runner: JUnit 5] (6.013 s)

testCreate() (0.345 s)

getbalance() (0.170 s)

getreceiverbalance() (0.010 s)

receiverAccount() (0.009 s)

serviceone() (0.019 s)

servicetwo() (0.009 s)

transfer() (5.399 s)

receiverbankService() (0.042 s)

Failure Trace

```
100 //
109 //
110 //
111 //
112 //
113 //
114 //
115 //
116 //
117 //
118 //
119
120 Mtmmessage mtm =new Mtmmessage();
121 mtm.setAccountnumber("1720364789995554");
122 mtm.setAddress("SBI");
123 mtm.setAmount("1000");
124 mtm.setBankname("SBI");
125 mtm.setReceiver("ENF43332");
126 mtm.setReceiverAccountNo("11225546165621");
127 mtm.setRefernce("CRED");
128 mtm.setSender("ENFEES123");
129
130 Object abcd =service.transfermessage(mtm);
131 // then - verify the output
132 assertThat(abcd).isNotNull();
133
134
135 @Test
136 @Order(8)
137 public void receiverbankService() {
138     ReceiverBank p = new ReceiverBank();
139     p.setId(1L);
140     p.setAccountbalance(0);
141     p.setAccountnumber("11111");
142     p.setBankName("sbi");
143     p.setIfscCode("sbinss");
```

SWAGGER IMPLIMENTATION

Bank Solution

[Base URL: 'localhost:8080/']

<http://localhost:8080/v2/api-docs>

"Transfer money from one Account to Another Account"

[Anoop P.M - Website](#)

[Send email to Anoop P.M](#)

[Apache License Version 2.0](#)

message-controller

Message Controller

POST

/bankapi/addreceiveraccount

Register A Receiver User

PUT

/bankapi/deposit

Deposit to user account

GET

/bankapi/receiverbalance

Register A User

POST

/bankapi/transfermessage

Transfer amount

GET

/bankapi/userbalance

Check user account balance

POST

/bankapi/usercreate

Register A User