

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.

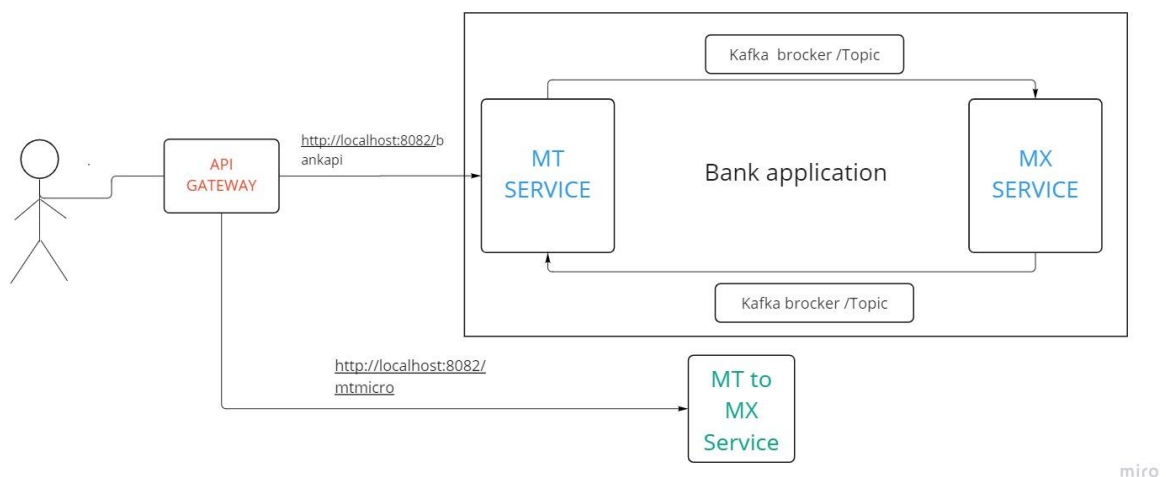
Used Technologies

- 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
```

```

{"SenderAccountnumber":"790773028412345","MTMessage":{"1:F01ENFEESS1A230000000000}{2:I103ENF43332XXXXN}{3:{121
:724209c7-6d6e-4a68-becd-
6c51bb0997bf}}{4:\r\n:20:REFERENCE\r\n:23B:CRED\r\n:32A:2208291NR1\r\n:50A:/790773028412345\r\nSBI\r\n:59:/95399318
67123\r\nSBI\r\n:71A:OUR\r\n-"},"time":"2022-08-29T09:50:12.369307200Z"}

```

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

```
{ "MXmessage": "<message>\r\n<block1>\r\n\t<applicationId>F<\ApplicationId>\r\n\t<serviceld>01</serviceld>\r\n\t<logicalTer  
minal>0000000000</logicalTerminal>\r\n\t</block1>\r\n\t<block2  
type=\"input\">\r\n\t\t<messageType>103</messageType>\r\n\t\t<receiverAddress>N</receiverAddress>\r\n\t\t</block2>\r\n\t\t<blo  
ck3>\r\n\t\t\t<tag>\r\n\t\t\t\t<name>121</name>\r\n\t\t\t\t<value>6e7e8273-8467-4b0b-8b3a-  
0bc9a1d356a5</value>\r\n\t\t\t\t</tag>\r\n\t\t\t\t</block3>\r\n\t\t\t\t<block4>\r\n\t\t\t\t\t<tag>\r\n\t\t\t\t\t\t<name>20</name>\r\n\t\t\t\t\t\t<value>REFEREN  
CE</value>\r\n\t\t\t\t\t\t</tag>\r\n\t\t\t\t\t\t<tag>\r\n\t\t\t\t\t\t\t<name>23B</name>\r\n\t\t\t\t\t\t\t<value>CRED</value>\r\n\t\t\t\t\t\t\t</tag>\r\n\t\t\t\t\t\t\t<tag>\r\n\t\t\t\t\t\t\t\t<name>32A</name>\r\n\t\t\t\t\t\t\t\t<value>220829100</value>\r\n\t\t\t\t\t\t\t\t</tag>\r\n\t\t\t\t\t\t\t\t<tag>\r\n\t\t\t\t\t\t\t\t\t<name>50A</name>\r\n\t\t\t\t\t\t\t\t\t<value>/1  
720364789995554\r\n\t\t\t\t\t\t\t\t\t</value>\r\n\t\t\t\t\t\t\t\t\t</tag>\r\n\t\t\t\t\t\t\t\t\t<tag>\r\n\t\t\t\t\t\t\t\t\t\t<name>59</name>\r\n\t\t\t\t\t\t\t\t\t\t<value>/11225546165621\r\n\t\t\t\t\t\t\t\t\t\t</value>\r\n\t\t\t\t\t\t\t\t\t\t</tag>\r\n\t\t\t\t\t\t\t\t\t\t<tag>\r\n\t\t\t\t\t\t\t\t\t\t\t<name>71A</name>\r\n\t\t\t\t\t\t\t\t\t\t\t<value>OUR</value>\r\n\t\t\t\t\t\t\t\t\t\t\t</tag>\r\n\t\t\t\t\t\t\t\t\t\t</block4>\r\n\t\t\t\t\t\t\t\t\t</v  
alue>"}"
```

- 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/transfermessage> (post method)

Json Request body =

```
{
  "accountnumber": "790773028412345",
  "address": "SBI",
  "amount": "100",
  "bankname": "SBI",
  "currency": "INR",
  "receiver": "ENF43332",
  "receiverAccountNo": "9539931867123",
  "refernce": "CRED",
  "sender": "ENFEESS123 }
```

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

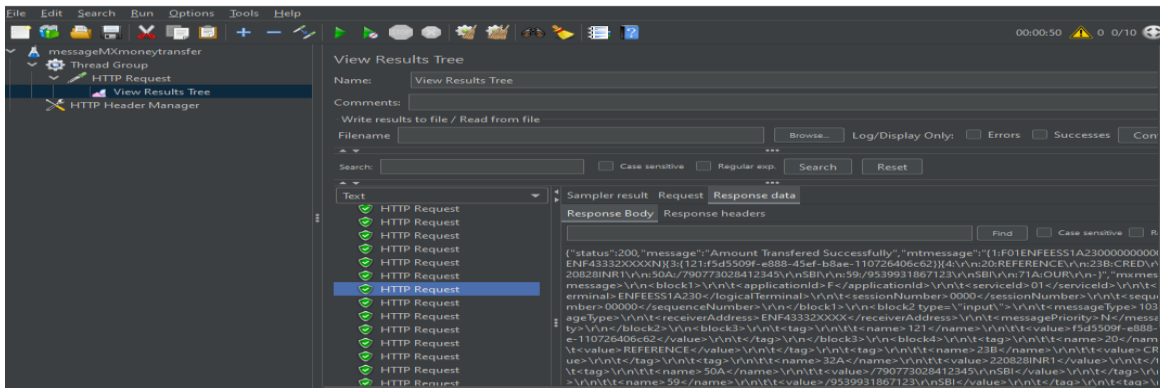
URI = http://localhost:8082/mtmicro/mttovalue (Get Method)

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

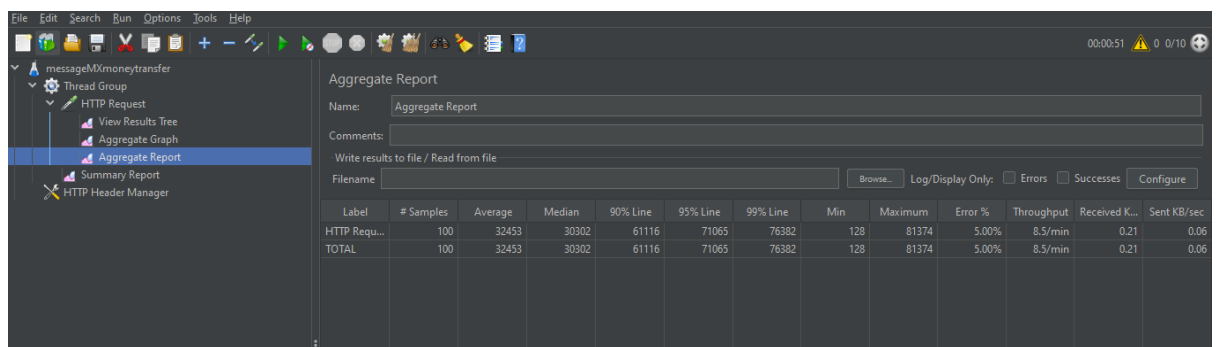
```
{
  "message": "{1:F01BICFOOYYAXXX8683497519}{2:01031535051028ESPBESMMAXX54237522470510281535N}{3:{113:ROMF}{108:0510280182794665}{119:STP}}{4:\r\n:20:0061350113089908\r\n:13C:/RNCTIME/1534+0000\r\n:23B:CRED\r\n:23E:SDVA\r\n:32A:061028EUR100000,\r\n:33A:081029EUR120000,\r\n:33B:EUR100000,\r\n:50K:/12345678\r\nAGENTES DE BOLSA FOO AGENCIA\r\nAV XXXXX 123 BIS 9 PL\r\n12345 BARCELONA\r\n:52A:/2337\r\nFOOAESMMXX\r\n:53A:FOOAESMMXX\r\n:57A:BICFOOYYXXX\r\n:59:/ES0123456789012345671234\r\nFOO AGENT ES DE BOLSA ASOC\r\n:71A:OUR\r\n:72:/BNF/TRANSF. BCO. FOO\r\n-}{5:{MAC:88B4F929}{CHK:22EF370A4073}}"
```

Project Performance Test

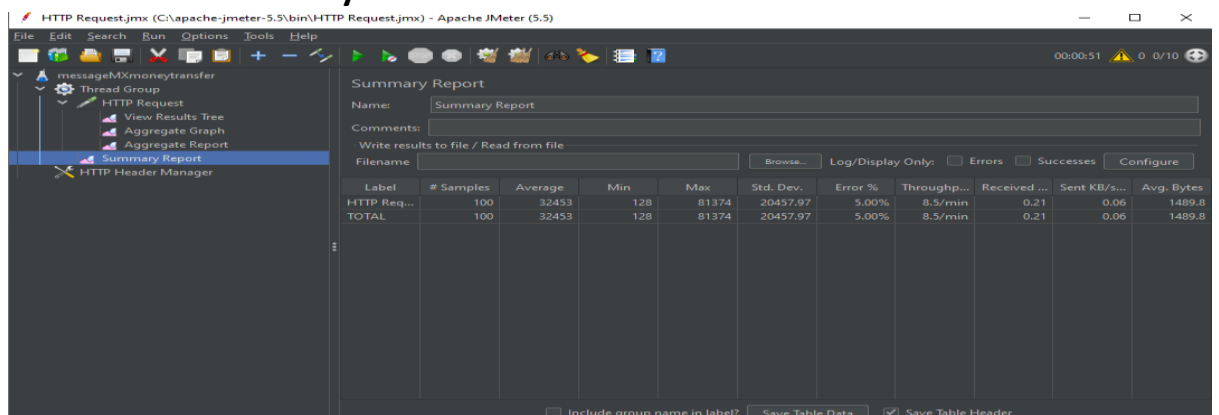
Using JMeter



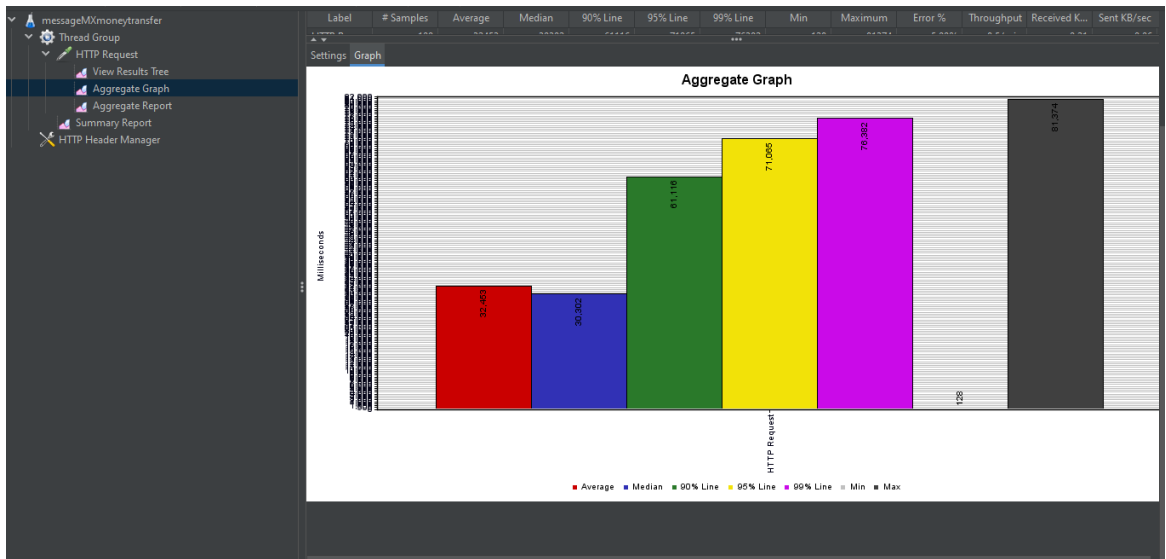
AGGREGATE REPORT

[illegible]

AGGREGATE Summary



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)

```
100 // 1
109 //     "accountnumber": "1720364789995554",
110 //     "address": "SBI",
111 //     "amount": "1000",
112 //     "bankname": "SBI",
113 //     "currency": "INR",
114 //     "receiver": "ENF43332",
115 //     "receiverAccountNo": "11225546165621",
116 //     "reference": "CRED",
117 //     "sender": "ENFEESS123"
118 // }
119 Mtmmessage mtm =new Mtmmessage();
120 mtm.setAccountnumber("1720364789995554");
121 mtm.setAddress("SBI");
122 mtm.setAmount("1000");
123 mtm.setBankname("SBI");
124 mtm.setReceiver("ENFSSS");
125 mtm.setReceiverAccountNo("11225546165621");
126 mtm.setReference("CRED");
127 mtm.setSender("ENFFSS");
128
129 Object abcd =service.transfermessage(mtm);
130 // then - verify the output
131 assertThat(abcd).isNotNull();
132
133
134 @Test
135 @Order(0)
136 public void receiverbankService() {
137     ReceiverBank p = new ReceiverBank();
138     p.setId(1L);
139     p.setAccountbalance(0);
140     p.setAccountnumber("11111");
141     p.setBankName("sbi");
142     p.setIfscCode("sbinss");
```

SWAGGER IMPLIMENTATION

Bank Solution^{1.0.0}
[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