**User Story :- 09**

Create an api to get account information which are not eligible for transferIn and transferOut

1. Requirement Analysis – Identify the Request and Response data

2. Create Raml to get account information which are not eligible for transferIn and transferOut

3. Create Spring boot App to get account information which are not eligible for transferIn and transferOut

a. Retrieve Data from DB table

b. Push Data to the Queue before exposing Data through API

c. Retrieve data from soap endpoint and expose as Rest API

4.Handle exceptions,logging and other Non Functional Requirements(Security) as requested.

**Request fields**

GET /account

Accept: application/json

transferIn=false&tranferOut=false&page=1

**Response fields**

Response

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

{

  "Accounts" : {

AccountMasterId:” 200”

ParentAccountId:” 2468135790”

Nickname:” My Savings Account”

LineOfBusiness:” consumer”

AccountNumber:” 4561237890”

InterestRate:” 3”

Currency:” INR”

AccDescriptorId:” 111”

TransferIn:”0”

TransferOut:”0”

InterestRateType:”FIXED”

}

}

**Tables:-**

**AccountDescriptor**

|  |  |
| --- | --- |
| **Field** | **Type** |
| AccountId | String(unique key) |
| AccountType | Accounttype(ESOP,GUARDIAN,HOMEQUITYLOAN) |
| DisplayName | string |
| Description | string |
| Status | AccountStatus (String,CLOSED,OPEN,PAID) |
| AccountDescriptorId | int(primary Key) |
| CustomerId | string |

**Account**

|  |  |
| --- | --- |
| **Field** | **Type** |
| AccountMasterId | String(unique key) |
| ParentAccountId | Accounttype(ESOP,GUARDIAN,HOMEQUITYLOAN) |
| Nickname | string |
| LineOfBusiness | string |
| AccountNumber | string(unique key) |
| InterestRate | int |
| Currency | varchar |
| AccDescriptorId | Foreign Key |
| TransferIn | Boolean |
| TransferOut | Boolean |
| InterestRateType | String(FIXED,VARIABLE) |

**Success and Error scenarios**

200 : Success

Response

HTTP/1.1 200 OK

Content-Type: application/json; charset=utf-8

{

  code : ”200”

message : “Account information found successful”

}

404 : Account Not Found

Response

HTTP/1.1 404 Account not Found

Content-Type: application/json; charset=utf-8

{

  code : ”404”

message : “Account Not Found”

}

405 : Method Not Found

Response

HTTP/1.1 405 Method not Found

Content-Type: application/json; charset=utf-8

{

  code : ”405”

message : “Method Not Found”

}