**Analysis document (Story 19):**

1. **Create an api to query all information for a set of accounts provided in the payload.**
2. **Request/Response format :**

**Authorization**: Bearer D1VjrVnl9L2PK69aY-KDqWyKqKPJugtgGbhoyyli-EKIcmpdx-4I1ZSPUneKNOHmQ9MZ9uYZyz0Ue0rsXeqvqQ

**Request Formats (Content-Type):** application/json

**Response Formats (Accept):** application/json

**POST:**  /api/v1/accounts/details

**Request format:**

[

{

"accountId": "1357902468",

"startTime": "2015-04-01T00:00:00.000Z"

},

{

"accountId": "1357902469",

"startTime": "2015-05-01T00:00:00.000Z"

}

]

**Response (Data in json format):**

HTTP/1.1 200 OK

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

{

"total": 16,

"totalPages": 4,

"page": 9,

"accounts":

[

{

"accountId": "1357902468",

"displayName": "Savings Account",

"description": "Savings Account",

"accountType": "SAVINGS",

"status": "OPEN",

"parentAccountId": "2468135790",

"interestRate": 3.00,

"nickname": "My Savings Account A",

"currency": "AFN",

"accountNumber": "478978673",

"interestRateType": "FIXED",

"micrNumber": "9753108642"

},

{

"accountId": "1357902469",

"displayName": "Checking Account",

"description": "Checking Account",

"accountType": "CHECKING",

"status": "OPEN",

"parentAccountId": "2468135790",

"interestRate": 10.00,

"nickname": "My Savings Account A",

"currency": "AZN",

"accountNumber": "478978673",

"interestRateType": "FIXED",

"micrNumber": "579018642"

}

]

}

1. **Possible Error Codes:**

* If Account is found with Id and response is success: set code 200.
* Invalid start date with time for particular account set http status code is 400 with message “Invalid start or end date”

{

“Code”: “400”,

“Message”: “Invalid account id”

}

* If access token is not valid set http status code is 401 with message “access token not valid”

{

“Code”: “401”,

“Message”: “access token not valid”

}

* If Account not found with id set http status code is 404 with message “Account not found”

{

“Code”: “404”,

“Message”: “Account not found”

}

* If server doesn’t respond set http status code 500 Internal server Error

{

Code : “500”,

Message : “Internal server Error”

}

* HTTP Version Not Supported set http status code 505

{

“Code”: “505”,

“Message”: “http version not supported”

}

1. **Table references:**

**Account, accountdescriptor.**

1. **Entity classes involves in the particular requirement:**

* **SingleAccountDetailsRequest**

|  |  |
| --- | --- |
| **Type** | **Field** |
| string | accountId |
| Timestamp | startTime |
| Timestamp | endTime |
| Number | page |

* **AccountDescriptor**

|  |  |
| --- | --- |
| **Type** | **Field** |
| Identifier | accountId |
| Accounttype | accountType |
| AccountStatus (ENUM) | status |
| string | displayName |
| string | description |
| string | customerId |

* Enum class for AccountStatus and Accountype.

Enum AccountStatus

{

CLOSED, DELINQUENT, NEGATIVECURRENTBALANCE, OPEN,

PAID, PENDINGCLOSE, PENDINGOPEN

}

Enum AccountType

{

401A,401K,403B,529,AUTOLOAN,CD,CHARGE,CHECKING,COMMERCIALLINEOFCREDIT,COMMERCIALLOAN,COVERDELL,CREDITCARD,ESCROW,ESOP,GUARDIAN,HOMEEQUITYLOAN,HOMELINEOFCREDIT,INSITUTIONALTRUST,INSTALLMENT,IRA, KEOGH,LINEOFCREDIT,LOAN,MILATARYLOAN,MONEYMARKET,MORTGAGE,PERSONALLOAN,ROLLOVER,ROTH,SARSEP,SAVINGS,SMBLOAN,STUDENTLOAN,TAXABLE,TDA,TRUST,UGMA,UTMA

}

* **Account Entity (Extends AccountDescriptor)**

|  |  |
| --- | --- |
| Type | Field |
| Identifier | parentAccountId |
| string | accountNumber |
| number | interestRate |
| string | nickname |
| InterestRateType | interestRateType |
| iso4217Code | Currency |
| integer | transferIn |
| integer | transferOut |
| string | lineOfBusiness |

Enum InterestRateType

{

FIXED, VARIABLE

}

Enum Iso4217Code

{

AED,AFN,ALL,AMD,ANG,AOA,ARS,AUD,AWG,AZN,BAM,BBD,BDT,BGN,BHD,BIF,BMD,BND,BOB,BOV,BRL,BSD,BTN,BWP,BYR,BZD,CAD,CDF,CHE,CHF,CHW,CLF,CLP,CNY,COP,COU,CRC,CUC,CUP,CVE,CZK,DJF,DKK,DOP,DZD,EGP,ERN,ETB,EUR,FJD,FKP,GBP,GEL,GHS,GIP,GMD,GNF,GTQ,GYD,HKD,HNL,HRK,HTG,HUF,IDR,ILS,INR,IQD,IRR,ISK,JMD,JOD,JPY,KES,KGS,KHR,KF,KPW,KRW,KWD,KYD,KZT,LAK,LBP,LKR,LRD,LSL,LYD,MAD,MDL,MGA,MKD,MMK,MNT,MOP,MRO,MUR,MVR,MWK,MXN,MXV,MYR,MZN,NAD,NGN,NIO,NOK,NPR,NZD,OMR,PAB,PEN,PGK,PHP,PKR,PLN,PYG,QAR,RON,RSD,RUB,RWF,SAR,SBD,SCR,SDG,SEK,SGD,SHP,SLL,SOS,SRD,SP,STD,SVC,SYP,SZL,THB,TJS,TMT,TND,TOP,TRY,TTD,TWD,TZS,UAH,UGX,USD,USN,UYI,UYU,UZS,VEF,VND,VUV,WST,XAF,XAG,XAU,XBA,XBB,XBC,XBD,XCD,XDR,XOF,XPD,XPF,XPT,XSU,XTS,XUA,XXX,YER,ZAR,ZMW,ZWL }

* **Accounts Entity**

It will be used to get all accounts.

|  |  |
| --- | --- |
| **Type** | **Field** |
| int | total |
| int | totalpages |
| int | page |
| List<Account>  Account can be DepositAccount,  InvestmentAccount,  LoanAccount, or  LocAccount | accounts |