

REQUIREMENT ANALYSIS DOCUMENT

1.User Story : Create an api to get an account

Request

POST /account

Accept: application/json

Authorization: Bearer w0mcJylzCn-AfvuGdqkty2-KP48=

Content-Type: application/x-www-form-urlencoded

accountId=1357902468

Response

HTTP/1.1 200 OK

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

```
{  
  "DepositAccount" : { }  
}
```

Response type: DepositAccount, LoanAccount, LocAccount, or InvestmentAccount

Task:

1. Requirement Analysis – Identify the Request and Response data
2. Create Raml to get Account information
3. Create Spring boot App to get Account
 - 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. Create Mulesoft App to get Account

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

2.Entities and related Entities

-AccountDescriptorEntity

-AccountEntity(PK-AccountId) --> Uses **CurrencyEntity**

-DepositAccountEntity --> Uses **DepositTransactionEntity**

-LoanAccountEntity --> Uses **LoanTransactionEntity**-->Uses
PaymentDetailsEntity

-LocAccountEntity --> Uses **LocTransactionEntity**

-InvestmentAccountEntity --> Uses **HoldingEntity** , **OpenOrderEntity**,
ContributionEntity , **VestingEntity** ,**InvestmentBalance**,
InvestmentTransaction ,**InvestmentLoan**

-ErrorEntity

3.Entity feilds and their datatypes taken in consideration

-AccountDescriptorEntity

-AccountId: *Identifier (String with length 128)*

-AccountType: *String[LOAN,DEPOSIT,INVESTMENT,LOC]*

-DisplayName: *String*

-Status: *String[CLOSED,DELINQUENT,NEGATIVECURRENTBALANCE,OPEN,PAID,PENDINGCLOSE,PENDINGOPEN]*

-AccountEntity

-ParentAccountId: *Identifier(String with length 128)*

-Nickname: *String*

-Currency:**CurrencyEntity**

-AccountNumber: *String*

-InterestRate: *Number (decimal)*

-BalanceType: *String[ASSET,LIABILITY]*

-CurrencyEntity

-CurrencyRate: *Number (decimal)*

-CurrencyCode: *ISO4217Code*

-OriginalCurrencyCode: *ISO4217Code*

-DepositAccountEntity

-BalanceAsOf: *Timestamp*

-CurrentBalance: *Number*

-OpeningDayBalance: *Number*

-AvailableBalance: *Number*

-AnnualPercentageYield: *Number*

-InterestYtd: *Number*

-Term: *Int*

-MaturityDate: *Timestamp*

-Transactions: *Array of DepositTransactionEntity*

-DepositTransactionEntity

-TransactionType: *DepositTransactionType* [CHECK, WITHDRAWAL, TRANSFER, POSDEBIT, ATMWITHDRAWAL, BILLPAYMENT, FEE, DEPOSIT, ADJUSTMENT, INTEREST, DIVIDEND, DIRECTDEPOSIT, ATMDEPOSIT, POSCREDIT]

-Payee: *String255*

-CheckNumber: *Int*

-LoanAccountEntity

-BalanceAsOf: *Timestamp*

- PrincipalBalance: *Number*
- EscrowBalance: *Number*
- OriginalPrincipal: *Number*
- OriginatingDate: *Timestamp*
- LoanTerm: *Int*
- TotalNumberOfPayments: *Int*
- NextPaymentAmount: *Number*
- NextPaymentDate: *Timestamp*
- PaymentFrequency: *PaymentFrequency*[DAILY, WEEKLY, BIWEEKLY, SEMIMONTHLY, MONTHLY, SEMIANNUALLY, ANNUALLY]
- CompoundingPeriod: *CompoundingPeriod* [DAILY, WEEKLY, BIWEEKLY, SEMIMONTHLY, MONTHLY, SEMIANNUALLY, ANNUALLY]
- PayoffAmount: *Number*
- LastPaymentAmount: *Number*
- LastPaymentDate: *Timestamp*
- MaturityDate: *Timestamp*
- InterestPaidYearToDate: *Number*
- Transactions: Array of ***LoanTransactionEntity***

-LoanTransactionEntity

- TransactionType: *LoanTransactionType*[PAYMENT, FEE, ADJUSTMENT, INTEREST]
- PaymentDetails: ***PaymentDetails***

-LocAccountEntity

- BalanceAsOf: *Timestamp*
- CreditLine: *Number*

- AvailableCredit: *Number*
- NextPaymentAmount: *Number*
- NextPaymentDate: *Timestamp*
- PrincipalBalance: *Number*
- CurrentBalance: *Number*
- MinimumPaymentAmount: *Number*
- LastPaymentAmount: *Number*
- LastPaymentDate: *Timestamp*
- PointsAccrued: *Number*
- CurrentRewardsBalance: *Number*
- PointsRedeemed: *Number*
- PurchasesApr: *Number*
- AdvancesApr: *Number*
- CashAdvanceLimit: *Number*
- AvailableCash: *Number*
- FinanceCharges: *Number*
- Transactions: *Array of **LocTransactionEntity***

-InvestmentAccountEntity

- BalanceAsOf: *Timestamp*
- AllowedCheckWriting: *Boolean*
- AllowedOptionTrade: *Boolean*
- CurrentValue: *Number*
- Holdings: *Array of **HoldingEntity***
- OpenOrders: *Array of **OpenOrderEntity***

- Contribution: Array of **ContributionEntity**
- Vesting: Array of **VestingEntity**
- InvestmentLoans: Array of **InvestmentLoan**
- AvailableCashBalance: *Number*
- Margin: *Boolean*
- MarginBalance: *Number*
- ShortBalance: *Number*
- RolloverAmount: *Number*
- EmployerName: *String*
- BrokerId: *String*
- PlanId: *String*
- CalendarYearFor401k: *Timestamp*
- BalanceList: Array of **InvestmentBalance**
- DailyChange: *Number*
- PercentageChange: *Number*
- Transactions: Array of **InvestmentTransaction**
- ErrorEntity**
 - Code: *String*
 - Message: *String*

4.Sample Request/Response

Application Endpoint: /api/v1/account

HTTP method: POST

Accept: application/json

Content-Type: application/json

Request body example:

```
{  
  "AccountId" : "0001001DEPO"  
}
```

Response body example:

HTTP/1.1 200 OK

Content-Type: application/json

```
{  
  "DepositAccount": {  
    "AccountId" : "0001001DEPO",  
    "ParentAccountId" : "0001001",  
    "BalanceAsOf" : 2018-05-01T15:16:03.387Z,  
    "CurrentBalance": 90000.00,  
    "OpeningDayBalance" : 100000.00,  
    "AvailableBalance" : 90000.00,  
    "AnnualPercentageYield": 10.05,  
    "InterestYtd" : 20.00,  
    "Term" : 10.00,  
    "MaturityDate" : 2018-05-01T15:16:03.387Z,  
    "Transactions" : [{"TRANSFER','CGSalary',101"}, {"ADJUSTMENT','Hsbc',102"},...]  
  }  
}
```

Response type: DepositAccount

5.Security

Use Basic Spring boot security feature of authentication using the security dependency that requires to enter username and password before the user accesses the Endpoint.

6.Exception Handling/HTTP response codes

Error Code	HTTP Status Code	Description
	200	OK ,successful response from the endpoint
	201	Created,The request has been fulfilled, resulting in the creation of a new resource
	400	Bad Request
	401	Unauthorized,User is not authenticated
701	404	Account not found