Quick Start Guide

Project :
Creating an FDX-Compliant API for Deposit Account Management Microservice

Source Code	**https://github.com/opendesk-ca/SpringBootGraphQL.git**
Branch	**SpringBoot-GraphQL-Project-Setup**
How to run the application?	**Follow the steps below.**

Steps :

1 To clone the Repository from your command line

```
git clone https://github.com/opendesk-ca/SpringBootGraphQL.git
```

2 Switch to the target branch :

```
cd SpringBootGraphQL

git checkout SpringBoot-GraphQL-Project-Setup
```

3 Open your project in your IDE

Select "Open" or "Import Project" and navigate to the cloned repository's directory.

4 Run the application

Find the main class with @SpringBootApplication, right-click on it, and select "Run".

5 Access the application

At **http://localhost:8080/graphiql** in a browser or through a GraphQL client tool like Postman.

High Level Functional Requirements:

1. Build Financial Data Exchange (FDX) Compliant API: Deposit Account Management MicroService.

2. A "Deposit Account" extends Accounts and its definition can be found in the FDX Specification. (https://docs.akoya.com/docs/data-elements-fdx-v4#depositaccount)
    a. Refer to "Deposit Accounts Data Elements"

3. A "Deposit Transaction" extends Transaction. Its definition can be found in the FDX Specification. (https://docs.akoya.com/docs/data-elements-fdx-v4#deposittransaction).
    a. Refer to "Deposit Transaction Data Elements"

4. Design GraphQL Schema for the model.

5. Implement the following APIs using SpringBoot 3, GraphQL and JPA

API	Description
addAccount (account: Account) : Boolean	Takes an Account Object as input. Adds an Account to the Database.
editAccount (account: Account) : Account	Facilitate Editing of Accounts
deleteAccount (accountID: ID) : Boolean	Facilitate Deleting of Accounts

getAccountsById (accountID: ID) : [Account]	Query Accounts for a given account Id
getAccounts (): [Account]	Get All Accounts
addTransaction (transaction: Transaction) : Boolean	Facilitate adding a Transaction to the Database.
getTransactionById (accountId: ID) : [Transaction]	Query Transactions for a given Account Id.

6. Add JUint Test Case for all the APIs using GraphQL IT Tester Framework.

7. Implement GraphQL Specific Error Handling.

8. Modularise GraphQL Schema into multiple files to facilitate ease of maintenance such as:
    a. main.graphqls
    b. account.graphqls
    c. Transaction.graphqls

9. Use GraphQL type enums for to restrict values for InterestRateType, accountType, AccountStatus, transactionType

Accounts Data Elements

Data Element	Type	Description
accountId	Identifier	Long-term persistent identity of the account. Not an account number. This identity must be unique to the data provider
accountType	AccountType	Enum: CD, CHECKING, CREDITCARD, LOAN, MORTGAGE
accountNumber	string	Full account number.
accountNumberDisplay	string	This is to be displayed by the data recipient.
nickName	string	Name given by the user. Used in UIs to assist in account selection

status	AccountStatus	Enum: OPEN, CLOSED, PENDINGOPEN, PENDINGCLOSE, DELINQUENT
description	string	Description of account
lineOfBusiness		The line of business, such as consumer, consumer joint, small business, corporate, etc.
routingTransitNumber		Routing transit number (RTN) associated with account number at data provider.
interestRate	Number	Interest Rate of Account
interestRateType	InterestRateType	Enum: FIXED or VARIABLE
interestRateAsOf	Timestamp	Date of account's interest rate. ISO 8601 date time with milliseconds in UTC time zone.
currency	Currency	
lastActivityDate	Timestamp	Date that last transaction occurred on account. ISO 8601 date time with milliseconds in UTC time zone.

Deposit Accounts Data Elements *(Inherit Accounts)*

Data Element	Type	Description
balanceAsOf	Timestamp	As-of date of balances
currentBalance	Number	Balance of funds in account
openingDayBalance	Number	Day's opening fund balance
availableBalance	Number	Balance of funds available for use
annualPercentageYield	Number	Annual Percentage Yield
interestYtd	Number	YTD Interest
term	int	Term of CD in months
maturityDate	Timestamp	Maturity date for CDs. ISO 8601 date time with milliseconds in UTC time zone.

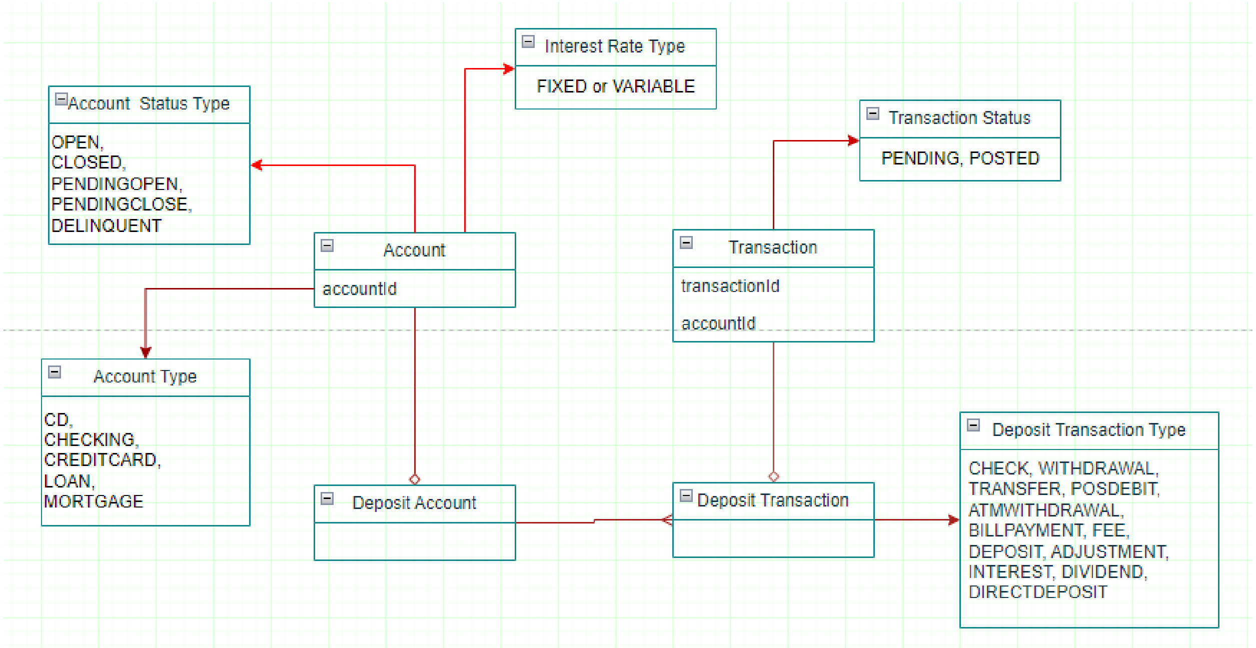
Transaction Data Elements

Data Element	Type	Description
accountId	Identifier	Corresponds to AccountId in Account
transactionId	Identifier	Long term persistent identity of the transaction.
postedTimestamp	Timestamp	The date and time that the transaction was posted to the account.
transactionTimestamp	Timestamp	The date and time that the transaction was added to the server backend systems.
description	String	The description of the transaction
status	TransactionStatus	PENDING, POSTED
amount	Number	The amount of money in the account currency

Deposit Transaction Data Elements *(Inherit Transaction)*

Data Element	Type	Description
transactionType	DepositTransactionType	CHECK, WITHDRAWAL, TRANSFER, POSDEBIT, ATMWITHDRAWAL, BILLPAYMENT, FEE, DEPOSIT, ADJUSTMENT, INTEREST, DIVIDEND, DIRECTDEPOSIT
payee	String	Payee name
checkNumber	Int	Check number

Proposed Domain Model


Interest Rate Type
FIXED or VARIABLE
Account Status Type
OPEN,
CLOSED,
PENDINGOPEN,
PENDINGCLOSE,
DELINQUENT
Transaction Status
PENDING, POSTED
Account
accountId
Transaction
transactionId
accountId
Account Type
CD,
CHECKING,
CREDITCARD,
LOAN,
MORTGAGE
Deposit Transaction Type
CHECK, WITHDRAWAL,
TRANSFER, POSDEBIT,
ATMWITHDRAWAL,
BILLPAYMENT, FEE,
DEPOSIT, ADJUSTMENT,
INTEREST, DIVIDEND,
DIRECTDEPOSIT
Deposit Account
Deposit Transaction