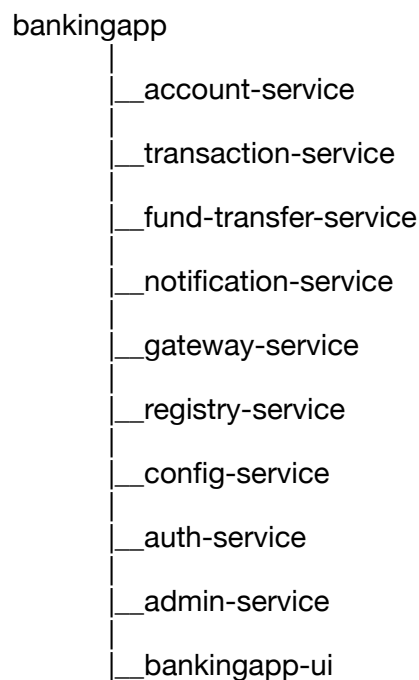


BankingApp Capstone Project:

Design and Develop Banking Application to manage account and transaction details with micro services based architecture with below services and technology stack with automated containerisation deployments into on promise and cloud infrastructure.

Technology Stack:

Spring Boot + Spring Cloud + REST API + Validation + Spring Data JPA / H2 + MySQL/MongoDB + Kafka +
Unit Testing and Integration/API Testing with 80% coverage + Docker + K8s + Azure Cloud + Agile Scrum



REST API

Account Service [account-service]

GET /accounts

GET /accounts/{id}

POST /accounts

PUT /accounts/{id}

DELETE /accounts/{id}

GET /accounts/search

POST /accounts/{id}/deposits [SYNC - REST API]

POST /accounts/{id}/withdrawal [SYNC - REST API]

Transaction Service [transaction-service]

GET /transactions

GET /transactions/{id}

POST /transactions

Fund Transfer Service [fund-transfer-service]

POST /fundTransfers [ASYNC - MESSAGING]

Notification Service [notification-service] [ASYNC - MESSAGING] Event Driven

Account

int id
String name
AccountType type
double balance
double roi
boolean active
LocalDateTime createTimestamp
LocalDateTime updateTimestamp

```
{  
  "name": "xyz",  
  "type": "Savings|Current|Loan",  
  "balance": 50000.0,  
  "roi": 7.5  
  "active": true|false  
  "createTimestamp": "2023-02-16T16:00:00",  
  "updateTimestamp": "2023-02-16T16:00:00"  
}
```

Transaction

int id
TransactionType type
String description
double amount
Account account
LocalDateTime transactionTimestamp

```
{  
  "type": "Credit|Debit",  
  "description": "School Fee"  
  "amount": 50000,  
  "accountId": 100,  
  "transactionTimestamp": "2023-02-16T16:00:00"  
}
```

Fund Transfers

```
{  
  "sourceAccountId": 100,  
  "targetAccountld": 200  
  "amount": 50000,  
  "description": "School Fee"  
}
```

Package
=====

