JBDL-65

Digital Wallet

4th September 2024

OVERVIEW

Create a project with different microservices performing the different parts of the transaction involved in the wallet.

Modules

1. User Service
2. Wallet Service
3. Transaction Service
4. Notification Service
5. Report Service [Add on later]

SPECIFICATIONS

Each of the services should have their own servers and should have individual databases.

1. User Service: 3000
2. Wallet Service: 9000
3. Transaction Service: 8000
4. Notification Service: 4000

Database Schema

User

1. Id
2. Name
3. Username
4. Email
5. Password
6. Mobile

7. Authority
8. Time Stamp

Wallet

1. walletId
2. Balance
3. Mobile
4. Time Stamp

Transaction

1. Id
2. externalTxnId
3. Sender // Sender Wallet Number or Sender Username
4. Receiver/ / Sender Wallet Number or Sender Username
5. Amount
6. Purpose
7. Status
8. Transaction Time // Creation TS
9. UpdatedOn // Updation TS

Notification [Add On]

1. Id
2. Sender
3. ExternalId
4. Receiver
5. Message
6. Status
7. SentTime

Controllers/API

User Module

Request Mapping: /user/<endpoint>

1. Create User
    a. Send a message to the user queue to create the wallet for the user
    b. Addon/Optional:

        i. Run the validation to ensure that the number starts with 6,7,8,9 and should be 10 digits
        ii. The Phone number should not be either a user or a wallet
        iii. OTP over the email to confirm the user's wallet creation.
2. Get User
    a. Use the Cache to store the user details in case not already cached
3. Load User details on the basis of username
4. Delete User [Addon later]
5. Update User Details [Addon later]

Wallet Module

1. Kafka Listener to listen the User Create and Create a Wallet with promotional Balance
2. API for wallet transaction on sender and Receiver

Transaction Module

Request Mapping: /transaction/<endpoint>

1. Transact:
    a. Check if Wallet Receiver Exists
    b. Check for user's balance and if it more than the transaction amount then process the transactions
    c. Send a OTP to validate the transaction's authenticity [Optional]
    d. Send the notification to wallet to deduct the amount

Notification Module

1. Kafka Listener to Notify User by sending an email.

Report Module [Add On]

Request Mapping: /report/<endpoint>

1. All transactions done by the user in the following ranges:
    a. Today
    b. Yesterday
    c. This Week
    d. Last Week
    e. This Month

    f. Last Month
    g. This Year
    h. Last Year
2. List all the active wallets with their username and balances
3. List new users created today

All the reports 1,2,3 should run on CRON every day at 12:00.