JBDL-60

Payment Wallet

17th December 2023

OVERVIEW

GOALS

1. Create an ER diagram based on the database schema
2. Create API/endpoints as per the specified controllers
3. You have to create an Architecture Diagram
4. Draw out the MSOA for the project in terms of data flow
5. Adding Spring Security
6. Adding Redis as database query cache
7. Adding database seed file [Hold]

SPECIFICATIONS

Entities:

1. Wallet
2. User
3. Transaction
4. Notification

Database Schema

User

1. Id
2. Name
3. Username
4. Address
5. Email
6. Age
7. Password

8. Authority
9. Phone
10. TimeStamps (Both the timestamps....which is Created and updated)

Wallet

1. Id
2. Username
3. walletId [Phone Number for the user]
4. Balance
5. Currency [Optional]
6. TimeStamps (Both the timestamps....which is Created and updated)

Transaction

1. Id
2. SenderId
3. Receiver Id
4. Amount
5. CreatedOn
6. UpdatedOn

Notification

1. Id
2. Purpose
3. Recipient
4. Message
5. Status
6. CreatedOn
7. UpdatedOn

Entities Relation

Source	Destination	Relation
User	Wallet	1-1

Controllers/API

User Controller

RequestMapping: /user/<endpoint>

1. CRUD API for User

Wallet Controller

RequestMapping: /wallet/<endpoint>

1. CRUD API for Wallet [To be Secure and authorized only for admin and self user]

Transaction Controller

RequestMapping: /transact/<endpoint>

1. User Creation/Onboarding Flow (Parameter: Phone Number, User Details)
    a. Run validation for the phone Number (It should be a 10 digit number and start from only 6,7,8,9)
    b. Minimum amount to create a wallet is 100
    c. Only INR currency based wallets are supported
    d. The Phone number should not have an active wallet pre registered
    e. If all of the above is ok, then create the wallet
        i. Create a new record for the wallet and set it to active with default currency as INR.
        ii. Once the above process is ok, send a kafka message to the user onboard topic to send an email to the user.
    f. If any of the above steps return an error, insert the transaction with status as failure
2. Payment (Parameter: receiverId, senderId, amount)
    a. Validation:
        i. ReceiverId and senderId should be valid 10 digits number
        ii. Amount should be greater than 0.
    b. ReceiverId and senderId both should be active
    c. The senderId should not have a negative balance after payment.
    d. The sender should not have sent more than 1000 in the last 24 hours. [Optional]
    e. If all the above are ok, then before processing the transaction [Optional]
        i. Send a kafka message to the OTP topic to the sender.
    f. Once step e is completed and a to d are ok, then
        i. Deduct money from Sender’s Wallet
        ii. Credit money to the Receiver's Wallet

3. If the above transactions are ok, then send a confirmation message to the Transaction confirmation topic.

Report Controller [Optional]

RequestMapping: /report/<endpoint>

1. All transaction done by user in between a date range
2. List of all active wallets and their usernames
3. List of new users signed up today
4. Total amount sent across the users.

All the kafka messages sent in the app should be saved in the database.