

Java Backend Developer Case (Digital Wallet Challenge)

You are expected to write a backend Wallet API for digital payment company so that their customers (or employees) can create wallet, deposit and withdraw-payment money.

Employees can make operations for all customers. Customers can make operations only for themselves

Your backend service should have those endpoints:

- **Create Wallet:** Create a new wallet with given details below
 - **WalletName:** Name of wallet to be created, **Currency:** Currency of wallet, **ActiveForShopping:** Wallet can be used for shopping, **ActiveForWithdraw:** Wallet can be used for withdraw
 - Acceptable currencies are TRY, USD, EUR
- **List Wallets:** List wallets for a given customer
 - If you want you can add more filters like currency, amount in wallet etc.
- **Deposit:** Make deposit with given details below
 - **Amount:** Amount to to deposit, **WalletId:** Wallet id to make deposit into, **Source:** Source of deposit. Can be an iban or payment id.
 - All deposits should be saved in transactions.
 - Amount more than 1000 should be saved with status PENDING, less than this amount should be saved with status APPROVED.
 - Approved deposits should be reflected to balance and usable balance of the wallet. Pending deposits should only be reflected to balance of the wallet.
- **List Transactions:** List transactions for a given wallet
- **Withdraw:** Make a withdraw with given details below
 - **Amount:** amount to withdraw-pay, **WalletId:** wallet id to make withdraw-pay from, **Destination:** Can be an iban or payment id.
 - Wallet settings for shopping and withdraw should be taken into account. If the setting is not ok, endpoint should return an appropriate error.
 - All withdraws should be saved in transactions.
 - Amount more than 1000 should be saved with status PENDING, less than this amount should be saved with status APPROVED.
 - Approved withdraws should be reflected to balance and usable balance of the wallet. Pending withdraws should only be reflected to usable balance of the wallet.
- **Approve:** Approve or deny a transaction with given details below
 - **TransactionId:** Transaction id to approve or deny
 - **Status:** approved or denied.

- After approving or denying, necessary reflections to wallet balance and usable balance should be done.

Requirements:

- Documentation is important. Prepare a document, readme, about how to use the application.
- All endpoints should be authorized appropriately.
- All info should be stored in database as below:
 - Customer: name, surname, TCKN
 - Wallet: customer id, wallet name, currency, active for shopping, active for withdraw, balance, usable balance.
 - Transaction: wallet id, amount, type, opposite party type, opposite party, status.
 - Type is either DEPOSIT or WITHDRAW
 - Opposite Party Type is either IBAN or PAYMENT
 - Status is PENDING, APPROVED or DENIED

Implementation:

We would like you to build a java application using Spring Boot Framework. You can use h2 db as database. You are expected to write some unit tests. Try to build and design your code as it will be deployed to prod (or at least test env 😊) Make sure to add information how to build and run the project.