

Cryptocurrency wallet

CryptoWallet is an application for managing and purchasing cryptocurrencies. Each Coin has an *id*, a *name* and a *value(in USD)*. A Wallet has an *id*, a *name* and a *list of coin amounts(foreign key)*. A CoinAmount has an *id*, a *coin(foreign key)* and an *amount*. See a few examples:

Coin:

id	name	value
1	Bitcoin	41657.58
2	Ethereum	2894.94
3	BNB	393.32

CoinAmount:

id	coin (id, name, value)	amount
11	(1, Bitcoin, 41657.58)	2
12	(2, Ethereum, 2894.94)	3.5

Wallet:

id	name	coinAmounts (list)
21	Andrei's Wallet	[(1, (1, Bitcoin, 41657.58), 2), (2, (2, Ethereum, 2894.94),3.5)]

In the practical session we have developed a spring boot application that implements the following requirements:

Requirement	Branch	Location of endpoint
List all coins	iteration/1	CoinController.java
Add a new coin	iteration/2	CoinController.java
Create a new wallet	iteration/3	WalletController.java

List all wallets	iteration/3	WalletController.java
Get wallet by id	iteration/3	WalletController.java
Buy coins	iteration/3	WalletController.java
Exchange coins	iteration/3	WalletController.java

As an assignment, please implement the following requirements:

- Create an API call that returns the total value of coins held in a wallet in USD
 - The HTTP method of the API endpoint should be GET
 - API call should have as the request parameter the *id* of the wallet (variable amount given as a query parameter, see `@QueryParam` annotation in Spring).
 - API call should have a response containing a float value consisting of the total value of the wallet in USD
- Add transaction history. A *Transaction* has an *id*, a *transaction date*, a *coin* (foreign key), an *amount* and a *totalValue*. Example:

Transaction:

id	transactionDate	coin (id, name, value)	amount	totalValue
1	30/03/2022 10:00:00	(1, Bitcoin, 41657.58)	2	83315.16
2	29/03/2022 11:37:24	(2, Ethereum, 2894.94)	3	8684.82

Every time a user buys a coin(only using the buy coin endpoint, not the exchange coins), a *Transaction* object has to be created and stored in the database.

Create an API call which lists all the transactions from the database.

- The HTTP method of the API endpoint should be GET.
- API call should have a response containing a list of transactions.

Create an API call which lists all the transactions with the *totalPrice* amount greater than a certain amount of USD (variable amount given as a query parameter, see `@QueryParam` annotation in Spring).

- The HTTP method of the API endpoint should be GET
- API call should have as the request parameter the *amountOfUsd* over which to show the transactions.
- API call should have a response containing a list of transactions.

Please submit your assignments by April 7th at 10:00.