Mobilab

Bank Account Management

REST API documentation

V 1.0

Entities

Account

The account is demonstrated as a virtual bank account. customers can have an account by their customer ID per currency, meaning that each customer can have one account for a specific currency.
The account is identified by its number. So for any operation, the number is the key metric that we going to use.
The account number will generate by the service itself and there is no way to change it.
The account number is a 4x4 random number.
There is two main currency considered, **EUR / USD.**
The account balance will be **0.0** at the time of creation.
Accounts can't be deleted from the database, a delete operation is considered a safe delete, which means on a delete operation, the **deleted** attribute will change to True.

Id	bigInt	Entity identifier
Number	varchar	account number
CustomerId	bigInt	customer identifier that come from another service
Name	varchar	customer name (can be modify)
Discription	varchar	Account description (can be modify)
Balance	Double	Balance of account
Currency	enum(USD/EUR)	Account currency that just can be USD or EUR
createdDate	Datetime	Account creation date time
updatedDate	datetime	Account update date time
Deleted	tinyint(1)	If account going to be delete -> True safe delete

Transaction

When a transfer happens between two accounts, two transaction records going to submit on the database. The first one is for the account that sent the money, the second one is for the account that received the money.
Transactions are the logs for all transfers that happen in the system.
Every transfer has a tracking code set on both submitted transactions. With the tracking code fetch both transactions, send and received.
The transaction can't be deleted, even when an account going to be safely deleted, all the transactions will remain on the database.

Id	bigInt	Entity identifier
accountNumber	varchar	the account number that transaction submitted for it
sourceAccountNumber	varChar	The account number that point to the account that sent the money
destinationAccountNumber	varChar	The account number that point to the account that received the money
sourceCurrency	enum(USD/EUR)	The main currency of the account that sent the money
destinationCurrency	enum(USD/EUR)	The main currency of the account that received the money
Amount	double	The amount on money that transferred from/to the account
exchangeRate	double	The exchange rate of the transfer, if both currency be the same, the default value will always be 1.0
Status	enum(SUCCESS,FAILURE)	The transaction status
Type	enum(SENT,RECEIVED)	The transaction type
trackingCode	varChar	The tracking code, on a transfer both transactions have same tracking code
createdDate	datetime	Transaction creation date time
updatedDate	datetime	Transaction update date time
Deleted	tinyInt(1)	Never use, comes from generic BaseEntity

Exception types and response structure

Response structure

The system exception response structure is :

```
{
    "timestamp": "time",
    "status": status_code,
    "error": “status_message”,
    "errors": [],
    "exception": “exception_detail",
    "message": “human readable message",
    "path": “the path, that exception accrue on it”
}
```

Inner System Types

NotFoundEntityException
Reason: when the record doesn’t exist on the database
Message: Entity record not found.
Status: 404

AccountCreationException
Reason: when an account exists for the customer by the currency, it's not possible for registering another account.
Message: customer already has an account with this currency, create new account is not possible.
Status: 406 (not_acceptable)

NotCustomerAccountException
Reason: when a customer wants to charge an account, but the account does not belong to him/her.
Message: This account does not belong to the customer.
Status: 406 (not_acceptable)

API’s Gateways

Account Webservice

Create Account

```
Method:                 POST
Path:                   /v1/accounts
Payload:                AccountCreateDTO
Response:               AccountResponseDTO
Success_status:         201
Failure_status:         406
```

Payload example

```
{
    "name" : "amir moradi",
    "description" : "Bank account description",
    "customerId" : 1,
    "currency" : "EUR"
}
```

Response example

```
{
    "id": 1,
    "name": "amir moradi",
    "number": "7454894119734220",
    "description": "Bank account description",
    "customerId": 1,
    "balance": 0,
    "currency": "EUR",
    "createdDate": "2022-12-03T14:08:02.867+00:00"
}
```

Update Account

```
Method:               PUT
Path:                 /v1/accounts/{account_number}
Payload:              AccountUpdateDTO
Response:             AccountResponseDTO
Success_status:       200
Failure_status:       400, 404
```

Payload example

```
{
    "name" : "amir moradi",
    "description" : "bank account description"
}
```

Response example

```
{
    "id": 1,
    "name": "amir moradi",
    "number": "7454894119734220",
    "description": "Bank account description",
    "customerId": 1,
    "balance": 0,
    "currency": "EUR",
    "createdDate": "2022-12-03T14:08:02.867+00:00"
}
```

Delete Account

```
Method:                 DELETE
Path:                   /v1/accounts/{account_number}
Payload:                -
Response:               -
Success_status:         200
Failure_status:         404
```

Charge Account

```
Method:                 POST
Path:                   /v1/accounts/charge/{account_number}
Query_Param:            - amount (required)
Payload:                AccountChargeDTO
Response:               AccountResponseDTO
Success_status:         200
Failure_status:         404
```

Payload example

```
{
    "customerId" : 1,
    "amount" : 100
}
```

Get Account

```
Method:          GET
Path:            /v1/accounts
Query_Params:
                 - id (not required)
                 - startDate (not required | format_example: 2022-12-01 )
                 - endDate (not required | format_example: 2022-12-01 )
                 - number (not required)
                 - name (not required)
                 - customerId (not required)
                 - currency (not required | format_example: EUR )
                 - deleted (not required | format_example: false )
                 - pageable structure
Response:        Page<AccountResponseDTO>
Success_status:  200
```

Response example

```
{
    "content": [
        {
            "id": 14,
            "name": "amir moradi",
            "number": "4619750026358980",
            "description": "test bank account",
            "customerId": 1,
            "balance": 100,
            "currency": "USD",
            "createdDate": "2022-12-03T14:05:47.819+00:00"
        },
        {
            "id": 13,
            "name": "amir moradi",
            "number": "8137359468255659",
            "description": "test bank account",
            "customerId": 1,
            "balance": 100,
            "currency": "EUR",
            "createdDate": "2022-12-03T14:05:41.710+00:00"
        }
    ],
    "pageable": {
        "sort": {
            "empty": true,
            "sorted": false,
            "unsorted": true
        },
        "offset": 0,
        "pageNumber": 0,
        "pageSize": 20,
        "paged": true,
        "unpaged": false
    },
    "totalPages": 1,
    "totalElements": 4,
    "last": true,
    "size": 20,
    "number": 0,
    "sort": {
        "empty": true,
        "sorted": false,
        "unsorted": true
    },
    "numberOfElements": 4,
    "first": true,
    "empty": false
}
```

Transaction Webservice

Search

```
Method:          GET
Path:            /v1/transactions
Query_Params:
                 - id (not required)
                 - startDate (not required | format_example: 2022-12-01 )
                 - endDate (not required | format_example: 2022-12-01 )
                 - accountNumber (not required)
                 - sourceAccountNumber (not required)
                 - destinationAccountNumber (not required)
                 - status (not required | fromat_example: SUCCESS)
                 - trackingCode (not required)
                 - type (not required | format_example: SENT)
                 - pageable
Response:        Page<TransactionResponseDTO>
Success_status:  200
```

Response example

```
{
    "content": [
        {
            "id": 8,
            "accountNumber": "251812293643563",
            "sourceAccountNumber": "4619750026358980",
            "destinationAccountNumber": "251812293643563",
            "amount": 9.48992,
            "exchangeRate": 0.948992,
            "sourceCurrency": "USD",
            "destinationCurrency": "EUR",
            "status": "SUCCESS",
            "type": "RECEIVED",
            "trackingCode": "3c9ac6e0-d811-4c06-97ea-64dc5bf455ba",
            "createdDate": "2022-12-03T15:07:39.584+00:00"
        },
        {
            "id": 7,
            "accountNumber": "4619750026358980",
            "sourceAccountNumber": "4619750026358980",
            "destinationAccountNumber": "251812293643563",
            "amount": 10,
            "exchangeRate": 0.948992,
            "sourceCurrency": "USD",
            "destinationCurrency": "EUR",
            "status": "SUCCESS",
            "type": "SENT",
            "trackingCode": "3c9ac6e0-d811-4c06-97ea-64dc5bf455ba",
            "createdDate": "2022-12-03T15:07:39.576+00:00"
        }
    ],
    "pageable": {
        "sort": {
            "empty": true,
            "sorted": false,
            "unsorted": true
        },
        "offset": 0,
        "pageNumber": 0,
        "pageSize": 20,
        "paged": true,
        "unpaged": false
    },
    "totalPages": 1,
    "totalElements": 4,
    "last": true,
    "size": 20,
    "number": 0,
    "sort": {
        "empty": true,
        "sorted": false,
        "unsorted": true
    },
    "numberOfElements": 4,
    "first": true,
    "empty": false
}
```

Transfer Webservice

Transfer Request

```
Method:                 POST
Path:                   /v1/transfer
Payload:                TransferRequestDTO
Response:               TransferResponseDTO
Success_status:         code: 1
Failure_status:         code: 0
```

Payload example

```
{
    "sourceAccountNumber" : "4619750026358980",
    "destinationAccountNumber" : "5309196774O7434",
    "amount" : 10
}
```

Successful transfer Response

```
{
    "trackingCode": "c925a8ce-6a2a-4211-a828-11f886fb63a7",
    "message": "10.000000 USD successfully transferred from (4619750026358980)
to (5309196774O7434).",
    "code": 1
}
```

Unsuccessful transfer Response (Balance not enough)

```
{
    "trackingCode": null,
    "message": "Unsuccessful transfer : Balance is not enough.",
    "code": 0
}
```

Unsuccessful transfer Response (Account not found)

```
{
    "trackingCode": null,
    "message": "Unsuccessful transfer : Account not found.",
    "code": 0
}
```

Unsuccessful transfer Response (Exchange not respond)

```
{
    "trackingCode": null,
    "message": "Unsuccessful transfer : Exchange server did not respond
.",
    "code": 0
}
```