CS310 – Fall 2002 – Assignment 1

Instructions

- Create a Java Project NAMED AS YOUR NAME_LASTNAME pair in Spring Tool Suite (Eclipse).
- It is strictly RESTRICTED TO SHARE OR RECEIVE and code to or from other attendees.

Task

You are going to code a Web API that manages money transactions between accounts for a bank. Accounts can be created and transactions can be made from one account to the other. All communications must be through HTTP protocol, and data must be exchanged in JSON format. There are 5 endpoints in the API with the following rules and constraints:

(10 pts.)
- Accounts must have and id (String), owner (String) and createDate (Date String).
- Transactions must have from account id (String), to account id (String) and the amount transferred (Double)

(10 pts.) When Spring Boot application starts, manage the MongoDB database initialization and insertion of test data (if no data exists):

- You must create two accounts with id's "1111" and "2222".
- One transaction of 1.500$ from account "1111" to "2222".
- One transaction of 2.500 from account "2222" to "1111".

You must follow the URL's, parameters fields of JSON object exactly as described:

API

All JSON Responses must be in the following format:

```
{
    "message": "{SUCCESS/ERROR}",
    "data": {Object of the endpoint, null if ERROR}
}
```

MongoDB Connection String must be as follows:

```
mongodb://admin:pass@localhost:27017
database: cs310hw1
```

1. Create Account (16 pts.)

Creates new accounts.

Method : POST
URL   : http://localhost:8080/account/save
Accepts: JSON
Sample JSON Data for Request Body:

```
{
    "id":"3333",
    "owner":"Mary Perkinson"
}
```

Returns: JSON
Sample JSON Return Data for successful account creation ("createDate" must be the date of account creation):

```json
{
    "message": "SUCCESS",
    "data": {
        "id": "3333",
        "owner": "Mary Perkinson",
        "createDate": "2022-11-15T11:17:12.251"
    }
}
```

If "id" or "owner" fields are not provided returns:

```json
{
    "message": "ERROR:missing fields",
    "data": null
}
```

2. Create Transaction (16 pts.)

Creates new transactions from one account to another.

Method : POST
URL   : http://localhost:8080/transaction/save
Accepts: JSON
Sample JSON Data for Request Body:
```json
{
    "fromAccountId":2222,
    "toAccountId":1111,
    "amount":1500
}
```

Returns: JSON
Sample JSON Return Data for successful transaction creation ("createDate" must be the date of transaction creation, "from" and "to" refer to accounts):

```json
{
    "message": "SUCCESS",
    "data": {
        "id": "637353269aef9a0dbd61b1ef",
        "from": {
            "id": "2222",
            "owner": "Henry Williams",
            "createDate": "2022-11-15T11:16:58.784"
        },
        "to": {
            "id": "1111",
            "owner": "Jack Johns",
            "createDate": "2022-11-15T11:16:43.911"
```

```
        },
        "createDate": "2022-11-15T11:51:50.775",
        "amount": 1500.0
    }
}
```

If one of "fromAccountId" or "toAccountId" or "amount" are not provided returns:

```
{
    "message": "ERROR:missing fields",
    "data": null
}
```

If one of the "fromAccountId" or "toAccountId" are not existing then returns:
```
{
    "message": "ERROR: account id",
    "data": null
}
```

3. Account Summary (16 pts.)

Prints all information about an account including transactions from this account, transactions to this account and the current balance of the account. Balance is the sum of amounts of all incoming transactions minus amounts of all outgoing transactions. Balance of an account can be negative. List incoming and outgoing transactions separately.

Method : GET
URL   : http://localhost:8080/account/{accountId}

Returns: JSON

```
{
    "message": "SUCCESS",
    "data": {
        "id": "1111",
        "owner": "Jack Johns",
        "createDate": "2022-11-15T11:16:43.911",
        "transactionsOut": [
            {
                "id": "637350cdb1d4c17cc2205355",
                "from": {
                    "id": "1111",
                    "owner": "Jack Johns",
                    "createDate": "2022-11-15T11:16:43.911"
                },
                "to": {
                    "id": "2222",
                    "owner": "Henry Williams",
                    "createDate": "2022-11-15T11:16:58.784"
                },
                "createDate": "2022-11-15T11:41:49.47",
                "amount": 2000.0
            },
            {
```

```json
        "id": "637350d0b1d4c17cc2205356",
        "from": {
            "id": "1111",
            "owner": "Jack Johns",
            "createDate": "2022-11-15T11:16:43.911"
        },
        "to": {
            "id": "2222",
            "owner": "Henry Williams",
            "createDate": "2022-11-15T11:16:58.784"
        },
        "createDate": "2022-11-15T11:41:52.351",
        "amount": 2000.0
    },
    {
        "id": "637350d6b1d4c17cc2205357",
        "from": {
            "id": "1111",
            "owner": "Jack Johns",
            "createDate": "2022-11-15T11:16:43.911"
        },
        "to": {
            "id": "2222",
            "owner": "Henry Williams",
            "createDate": "2022-11-15T11:16:58.784"
        },
        "createDate": "2022-11-15T11:41:58.244",
        "amount": 3000.0
    }
],
"transactionsIn": [
    {
        "id": "637353259aef9a0dbd61b1ee",
        "from": {
            "id": "2222",
            "owner": "Henry Williams",
            "createDate": "2022-11-15T11:16:58.784"
        },
        "to": {
            "id": "1111",
            "owner": "Jack Johns",
            "createDate": "2022-11-15T11:16:43.911"
        },
        "createDate": "2022-11-15T11:51:49.115",
        "amount": 1500.0
    },
    {
        "id": "637353269aef9a0dbd61b1ef",
        "from": {
            "id": "2222",
            "owner": "Henry Williams",
            "createDate": "2022-11-15T11:16:58.784"
        },
```

```
            "to": {
                "id": "1111",
                "owner": "Jack Johns",
                "createDate": "2022-11-15T11:16:43.911"
            },
            "createDate": "2022-11-15T11:51:50.775",
            "amount": 1500.0
        }
    ],
    "balance": -4000.0
    }
}
```

If the account id is invalid returns:

```
{
    "message": "ERROR:account doesnt exist!",
    "data": null
}
```

4. List Incoming Transactions (16 pts.)

Prints the list of incoming transactions of an account.

Method : GET
URL   : http://localhost:8080/transaction/to/{accountId}

Returns: JSON

```
{
    "message": "SUCCESS",
    "data": [
        {
            "id": "637353259aef9a0dbd61b1ee",
            "from": {
                "id": "2222",
                "owner": "Henry Williams",
                "createDate": "2022-11-15T11:16:58.784"
            },
            "to": {
                "id": "1111",
                "owner": "Jack Johns",
                "createDate": "2022-11-15T11:16:43.911"
            },
            "createDate": "2022-11-15T11:51:49.115",
            "amount": 1500.0
        },
        {
            "id": "637353269aef9a0dbd61b1ef",
            "from": {
                "id": "2222",
                "owner": "Henry Williams",
                "createDate": "2022-11-15T11:16:58.784"
```

```
        },
        "to": {
            "id": "1111",
            "owner": "Jack Johns",
            "createDate": "2022-11-15T11:16:43.911"
        },
        "createDate": "2022-11-15T11:51:50.775",
        "amount": 1500.0
    }
  ]
}
```

If the account id is invalid returns:

```
{
    "message": "ERROR:account doesn't exist",
    "data": null
}
```

5. List Outgoing Transactions (16 pts.)

Prints the list of outgoing transactions of an account.

Method : GET
URL   : http://localhost:8080/transaction/from/{accountId}

Returns: JSON

```
{
    "message": "SUCCESS",
    "data": [
        {
            "id": "637350cdb1d4c17cc2205355",
            "from": {
                "id": "1111",
                "owner": "Jack Johns",
                "createDate": "2022-11-15T11:16:43.911"
            },
            "to": {
                "id": "2222",
                "owner": "Henry Williams",
                "createDate": "2022-11-15T11:16:58.784"
            },
            "createDate": "2022-11-15T11:41:49.47",
            "amount": 2000.0
        },
        {
            "id": "637350d0b1d4c17cc2205356",
            "from": {
                "id": "1111",
                "owner": "Jack Johns",
                "createDate": "2022-11-15T11:16:43.911"
            },
```

```json
            "to": {
                "id": "2222",
                "owner": "Henry Williams",
                "createDate": "2022-11-15T11:16:58.784"
            },
            "createDate": "2022-11-15T11:41:52.351",
            "amount": 2000.0
        },
        {
            "id": "637350d6b1d4c17cc2205357",
            "from": {
                "id": "1111",
                "owner": "Jack Johns",
                "createDate": "2022-11-15T11:16:43.911"
            },
            "to": {
                "id": "2222",
                "owner": "Henry Williams",
                "createDate": "2022-11-15T11:16:58.784"
            },
            "createDate": "2022-11-15T11:41:58.244",
            "amount": 3000.0
        }
    ]
}
```

If the account id is invalid returns:

```json
{
    "message": "ERROR:account doesn't exist",
    "data": null
}
```

Have Fun!