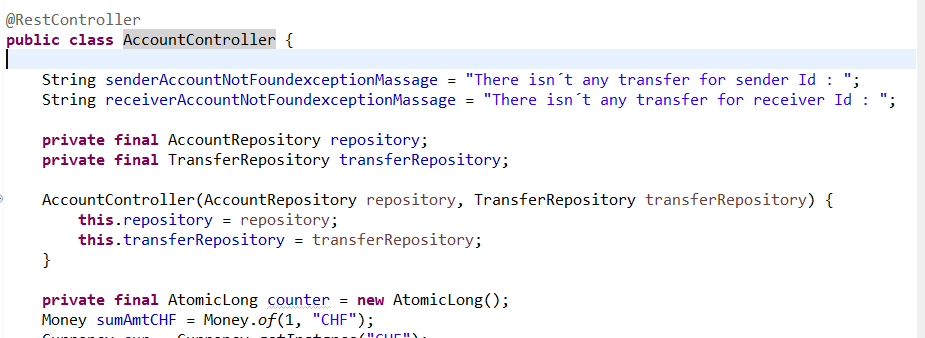
-Used a Maven Project due to automatic compile the dependencies i used.

Structure

HTTP requests are handled by a controller, class AccountController



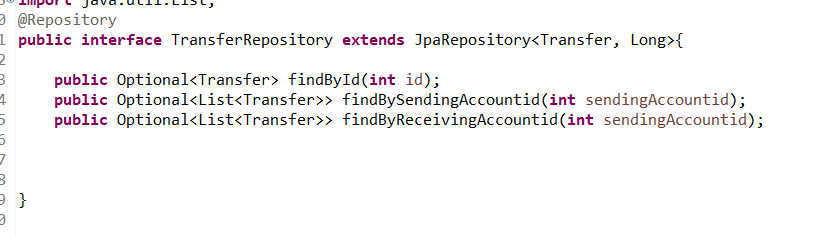
Models

I used a model for Account, a model for Transfer, and a a auxiliar model named AccountAx, to auxiliate Post (create new account) and Put(update account) methods, because the app wasn´t able to convert the Money information on json to Money Type Object. So the model AccountAx, has a Bigdecimal money instead of Money money)

Interfaces

Used interfaces TranferRepository and AccountRepository, to add on JpaRepositories the methods I need to querry the data.

Ex:



DB

On class LoadDatabase

I can add Accounts and Tranfers to be pre loaded we the service runs.

Services

<http://localhost:8080>

Get

@GetMapping("/account")

**public**

List<Account> all()

Return all accounts on repository

---------------------------------------------

@GetMapping("/account/{id}")

Account one(@PathVariable **int** id)

Return account by id

-------------------------------------------------

@GetMapping("/transfers")

List<Transfer> allTransfer() {

**return** transferRepository.findAll();

}

Return all transfers on repository

@GetMapping("/transfer/{id}")

Transfer transferByTransferId(@PathVariable **int** id)

Return transfer by transfer´s id

---------------------------------------

@GetMapping("/transfers/{id}/asSender")

Account id on Param

List<Transfer> transferAsSenderId(@PathVariable **int** id)

Return transfers by Account id as sender

-----------------------------------------

@GetMapping("/transfers/{id}/asReceiver")

Account id on Param

List<Transfer> transferAsReceiverId(@PathVariable **int** id)

Return transfers by Account id as receiver

----------------------------------------------------------------

Post

@PostMapping("/account")

**public**

Account newAccount(@RequestBody AccountAux newAccountAux)

Add new account

Json ex : {

        "name": "t3",

        "currency": "UD",

        "money":"60.95"

 ,

        "treasury": **true**

}

Return add account

-------------------------------------------------------------------

@PostMapping("/account/{id}/transfer")

Transfer accountTransfer(@RequestBody Transfer newTransfer, @PathVariable **int** id)

Makes a transfer from the sender account(on id param)(Sender id could be passed on json but i choosed do like that because looks more compreensive)

To receiver account(by json), and amount of money (by json)

Json ex: {

        "receivingAccountid": "3",

        "money":"100.00"

 }

Return the transfer, with complete information as status and sendingAccountid

-----------------------------------------------------------------------------

Put

@PutMapping("/account/{id}")

Account replaceAccount(@RequestBody AccountAux newAccountAux, @PathVariable **int** id)

Return updated account

-----------------------------------------------

Delete

@DeleteMapping("/account/{id}")

**void** deleteAccount(@PathVariable Long id)

Return deleted account

Testing the Rest Service

mvn spring-boot:run

or

mvn org.springframework.boot:spring-boot-maven-plugin:run