Exercise - Transfer money between two accounts

Using Spring Boot (java) we would like to ask you to design and build a simple microservice which handles financial transactions. Your service will need to expose a RESTful API (Application Programming Interface) that allows a user to make financial transactions between two bank accounts and meet the following acceptance criteria:

AC 1 - *Happy path for money transfer between two accounts*

Given source account exists

and target account exists

and source account has a balance greater or equal to the transaction amount

When a transaction request is received

Then the balance of source account should be debited

and the balance of target account should be credited

AC 2 - *Insufficient balance to process money transfer*

Given source account exists

and target account exists

and source account has a balance less than the transaction amount

When a transaction request is received

Then the balance of source account should remain the same

and the balance of target account should remain the same

and the client of the API should receive an error

AC 3 - *Transfer between same account*

Given source account exists

and both source and target accounts are the same

When a transaction request is received

Then the balance of source account should remain the same

and the client of the API should receive an error

AC 4 - *One or more of the accounts does not exist*

Given source or target account do not exist

When a transaction request is received

Then the balance of the existing account should remain the same

and the client of the API should receive an error

Data Model

You can consider the following structure for an account:

Field	Notes
`id`	Unique identifier of this account
`balance`	Decimal number, must always be positive
`currency`	For example, "GBP", the currency being transferred
`createdAt`	When the account was created

And this structure for a transaction:

Field	Notes
`id`	Unique identifier of this transaction
`sourceAccountId`	Id of the account sending the funds
`targetAccountId`	Id of the account receiving the funds
`amount`	For example, 10.50, the amount being transferred.
`currency`	For example, "GBP" the currency being transferred

Deliverables

Please return a zip file containing the *repo.bundle* resulting from the following command:

```
git bundle create repo.bundle --branches --tags
```

Make sure to include:

- All source code you produced.
- A gradle or maven project.
- Readme file containing your notes and suggestions.

General notes

- You can use https://start.spring.io to generate your project.
- You can mock any external service including the database in any way you see fit.
- You can use any libraries you would normally use.
- You can use either maven or gradle for dependency management.
- You should write as many tests you feel is necessary, but your code needs to compile, and the tests must pass.
- For this assignment, you can assume your API will be public, there is no need to handle security.
- If any part of the assignment is not clear you can make assumptions, we would appreciate if you made note of them in the README.md file.
- Feel free to note any future improvements you could make to your service that were out of the scope of this assignment.