Transfer of money between accounts Challenge

Here ,added functionality for a transfer of money between accounts.

In code Used below fields as Input JSON request :

com.db.awmd.challenge.domain.TransferMoneyReqDetails.java

accountFrom id
accountTo id
amount

Added below functionality & validation
- Account ID should not be blank
- Fund transferAmount should not be negative.
- Cheked added to validate sufficient balance is available or not.
- For notification called Notification services by considering other colleague will implement it.
- If all validation are correct then amount gets transferred between accounts and balance will reduced fromAccountID and added into toAccountId.
- In case of any error,it resets the state of each account.
- Taken care account should not be currupt or inconsitence in data.
- Added test cases

Java Controller as below
Java API :

Java Controller :
com.db.awmd.challenge.TransferMoneyController.java

Java Service :
com.db.awmd.challenge.service.TransferMoneyService.java

Test case API
Java Test Controller :
com.db.awmd.challenge.TransferMoneyControllerTest.java
Java Test Service :
com.db.awmd.challenge.TransferMoneyServiceTest.java

Service Endpoint : http://localhost:18080

Resource Path : /v1/accounts/Id-123

1) **To create Account use POST http method**

Resource Endpoint : POST : http://localhost:18080/v1/accounts/

Request 1 :

Media Type : application/json

```
{

  "accountId": "Id-123",

  "balance": 10000

}
```

Request 2 :

Media Type : application/json

```
{

  "accountId": "Id-456",

  "balance": 10000

}
```

2) **To retrive Account information use GET http method**

Resource Endpoint : GET : http://localhost:18080/v1/accounts/Id-123

Response 1:

```
{

  "accountId": "Id-123",

  "balance": 1000

}
```

Resource Endpoint : GET : http://localhost:18080/v1/accounts/Id-456

Response 1:

```
{

  "accountId": "Id-456",
```

"balance": 2000

}

    3) To transfer money between accounts

    Java Controller API : com.db.awmd.challenge.TransferMoneyController.java

Resource Endpoint : POST : http://localhost:18080/v1/moneytransfer

Request 1 :

Media Type : application/json

{

  "fromAccountId": "Id-123",

  "toAccountId": "Id-456"

   "amount" : 700

}

Note : T o check Amount for both account use resource

Resource Endpoint : GET : http://localhost:18080/v1/accounts/Id-123

Response 1:

{

  "accountId": "Id-123",

  "balance": 300

}

Resource Endpoint : GET : http://localhost:18080/v1/accounts/Id-456

Response 1:

{

  "accountId": "Id-456",

  "balance": 2700

}

Resource Endpoint : POST : http://localhost:18080/v1/moneytransfer

Request 2 : With same request

Media Type : application/json

{

  "fromAccountId": "Id-123",

  "toAccountId": "Id-456"

   "amount" : 700

}

Response : Will show no enough balance having Id-123