

## Assignment

Implement a Web service (SOAP/REST) that can fulfill the requirements as described below.

To build a Transfer Service, Using your service, users would be able to initiate a transfer from one account to another.

## Features

1. Ability to create monetary accounts with initial balance
2. Ability to transfer money from one account to another, with both the business rules taken in to account (Validate for negative balance , Simultaneous transfer)

## Data Definition

1. Account (  
    Name String not null, unique  
    Balance double not null  
)
2. TransferDetails {  
  
    FromAccount String not null,  
    ToAccount String not null,  
    TransferAmount Double not null  
  
}

## Description of Application

The application is developed as a Rest service. In addition, a presentation layer is also created where you can create account and transfer an amount between the accounts created. I would have preferred using a database, but for the assignment I have kept it as simple and have gone with a file based approach.

### *Rest Service details*

<i>HTTP</i>	<i>Function</i>	<i>URL</i>	<i>Valid Input</i>	<i>Success</i>
<i>POST</i>	<i>Create Account</i>	<i>/createAccount</i>	<i>{account_name: "Tom", "account_balance": "1000"}</i>	<i>200 (OK)</i>
<i>POST</i>	<i>Transfer Money</i>	<i>/transferAccount</i>	<i>{"fromAccount":"Tom", "toAccount":"Jerry", "transferAmount":"1000"}</i>	<i>200(OK)</i>
<i>GET</i>	<i>Get all Accounts</i>	<i>/getAllAccounts</i>		<i>200(OK)</i>

### **Unit and Integration test**

There are 8 unit test present in the controller for checking create and transfer logic and 2 integration test in the service layer ( Checking business logic like reading and writing file)

### **Technology used**

Java, Spring Boot, AngularJS, CSV

Testing: Postman (Rest client) , JMeter (Simultaneous request)

### **Limitations**

Current Implementation is using file for data storage. So transactions are not atomic.

Recommend to use database to cover transaction rollbacks

### **Deployment Instructions**

- 1.Clone the git repo "<https://github.com/merlinchacko/GENICO.git>" locally.
- 2.Go to Transfer-Service folder, run command "mvn clean install" (download apache maven if not installed).
- 3.Run command "mvn package" for creating jar in the target folder.
- 4.Run "java -jar target/transfer-service-0.0.1-SNAPSHOT.jar", for running the application.
- 5.Take "localhost:8080" in browser