Blue Harvest Code Assessment

# oVER vIEW

This is short document for explaining how to build and start the project and what already covered from technical requirements and bounce.

# Covered features

**Requirements**

The application should expose:

* API to be used for opening a “current account” of already existent customers: **Done**
* Endpoint which accepts the user information: **Done**
* New account will be opened connected to the user whose ID is customerID: **Done**
* if initialCredit is not 0, a transaction will be sent to the new account: **Done**
* Endpoint will output the user information showing Name, Surname, balance, and

transactions of the accounts: **Done**

* the data can be saved in memory and not actually persisted: **Done**
* source code published on github: **Done**

​**Bonus features**

* Accounts and Transactions are different services: **Done**
* Frontend: **Done**
* Testability will be also assessed. **Done**

# how to build, start AND RUN the project

The project implement using Java with Spring Boot, MVC, Core, JPA, javascript and maven.

To build and start the project please follow below:

* Clone repo from GitHub: <https://github.com/EhabKashef/BlueHarvestCodeAssessment>
* Repo: <https://github.com/EhabKashef/BlueHarvestCodeAssessment.git>
* Go to folder path
* Make sure you are using JDK 8 (open power shell and set java home and write $env:JAVA\_HOME="your\_path\jdk1.8.0\_74")
* Run the command: “mvn clean install” to build the project, make sure you
* Run the command: “mvn test” to run test cases and make sure project run correctly
* Run generated Jar BlueHarvestCodeAssessment-1.0.0.jar
* java -jar BlueHarvestCodeAssessment-1.0.0.jar
* download **BlueHarvestAssessment.postman\_collection.json** from repo and insert into postman to test endpoints

# how to test Endpoints AND FRONTEND

There are three services to retrieve, create users, accounts and transactions, after start the server and run the jar you can test the endpoints using postman project:

* import [**BlueHarvestAssessment.postman\_collection.json**](https://github.com/EhabKashef/BlueHarvestCodeAssessment/blob/master/BlueHarvestAssessment.postman_collection.json) file (on GitHub) into postman OR using below:

**User Service:** avail endpoints to create and retrieve users.

1. **User**: retrieve all users.

Method: GET

URL: <http://localhost:8080/BlueHarvestAssessment/api/user/>

Header:

Accept:application/json

Content-Type:application/json;charset=utf-8

Sample Response:

[{"id":1,"name":"Ehab","surname":"Ehab Surname","balance":0.0,"accounts":[]},{"id":2,"name":"Luran","surname":"Luran Surname","balance":0.0,"accounts":[]},{"id":3,"name":"Blue","surname":"Blue Surname","balance":0.0,"accounts":[]},{"id":4,"name":"Harvest","surname":"Harvest Surname","balance":0.0,"accounts":[]},{"id":5,"name":"qwew11","surname":"12","balance":123.0,"accounts":[]}]

1. **Create User:** create user

Method: POST

URL: <http://localhost:8080/BlueHarvestAssessment/api/user/>

Header:

Accept:application/json

Content-Type:application/json;charset=utf-8

Body:

{"name":"Mohamed","surname":"12","balance":"123"}

**Account Service:** avail endpoints to create and retrieve accounts.

1. **Account**: retrieve account by id.

Method: GET

URL: [http://localhost:8080/BlueHarvestAssessment/api/account/{id}](http://localhost:8080/BlueHarvestAssessment/api/account/%7bid%7d)

Header:

Accept:application/json

Content-Type:application/json;charset=utf-8

Sample Response:

{"id":1,"initialCredit":10.0,"transactions":[{"id":1,"amount":10.0}]}

1. **Account By Customer Id**: retrieve account by customer id.

Method: GET

URL: [http://localhost:8080/BlueHarvestAssessment/api/accounts/{id}](http://localhost:8080/BlueHarvestAssessment/api/accounts/%7bid%7d)

Header:

Accept:application/json

Content-Type:application/json;charset=utf-8

Sample Response:

[{"id":1,"initialCredit":10.0,"transactions":[{"id":1,"amount":10.0}]},{"id":2,"initialCredit":10.0,"transactions":[{"id":2,"amount":10.0}]}]

1. **Create Account:** create account

Method: POST

URL: <http://localhost:8080/BlueHarvestAssessment/api/account/>

Header:

Accept:application/json

Content-Type:application/json;charset=utf-8

Body:

{"initialCredit":"10","customerID":"2"}

**Transaction Service:** avail endpoints to create and retrieve accounts.

1. **Transition**: retrieve transaction by id.

Method: GET

URL: [http://localhost:8080/BlueHarvestAssessment/api/transaction/{id}](http://localhost:8080/BlueHarvestAssessment/api/transaction/%7bid%7d)

Header:

Accept:application/json

Content-Type:application/json;charset=utf-8

Sample Response:

{"id":1,"amount":10.0}

1. **Transactions By Account Id**: retrieve transaction by account id.

Method: GET

URL: [http://localhost:8080/BlueHarvestAssessment/api/transactions/{id}](http://localhost:8080/BlueHarvestAssessment/api/transactions/%7bid%7d)

Header:

Accept:application/json

Content-Type:application/json;charset=utf-8

Sample Response:

[{"id":1,"amount":10.0},{"id":3,"amount":-2.0}]

1. **Create Transaction:** create transaction

Method: POST

URL: <http://localhost:8080/BlueHarvestAssessment/api/transaction/>

Header:

Accept:application/json

Content-Type:application/json;charset=utf-8

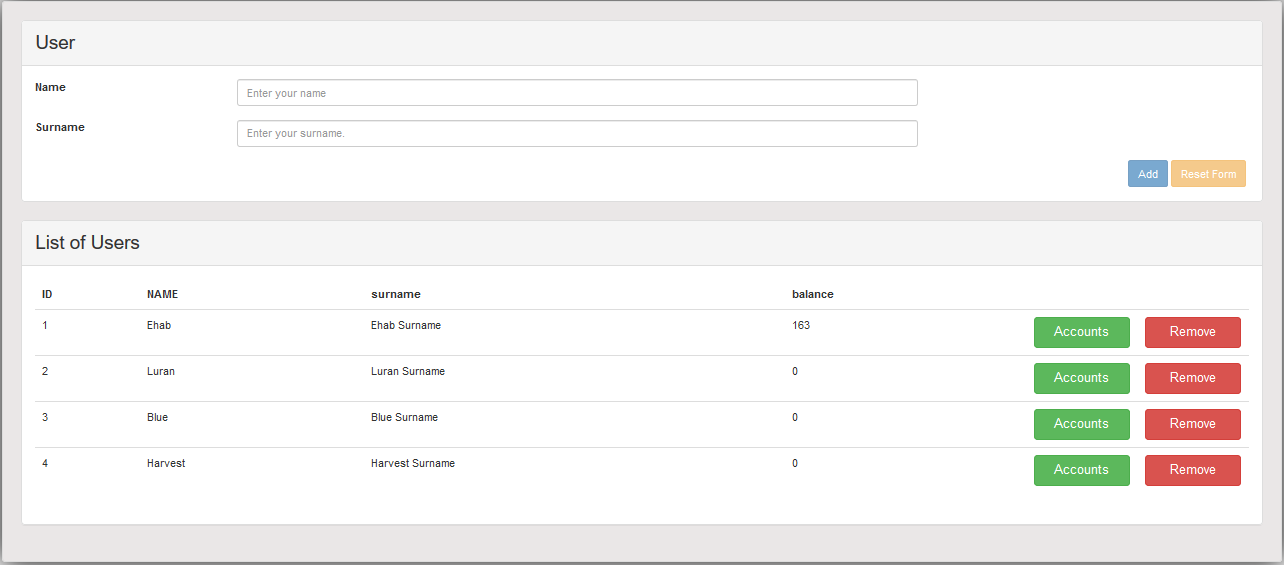
Body:

{"amount":"-2","account":{"id":"1"}}

# Test FRONTEND

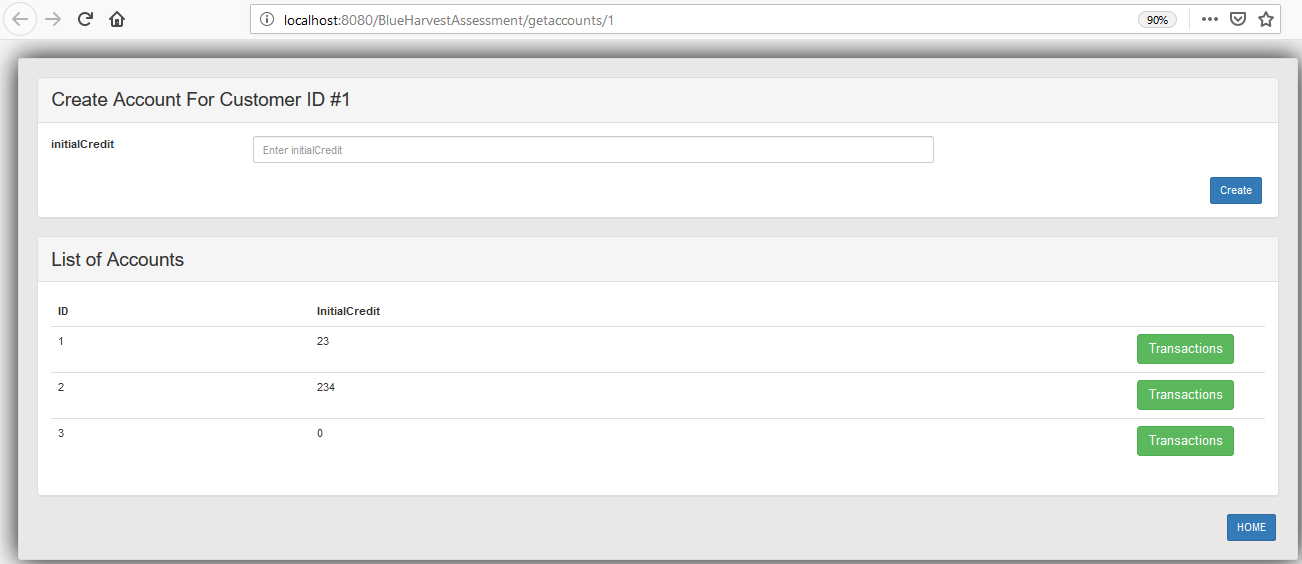
Home page URL: <http://localhost:8080/BlueHarvestAssessment/#/>

You can create or remove Users and show user accounts.



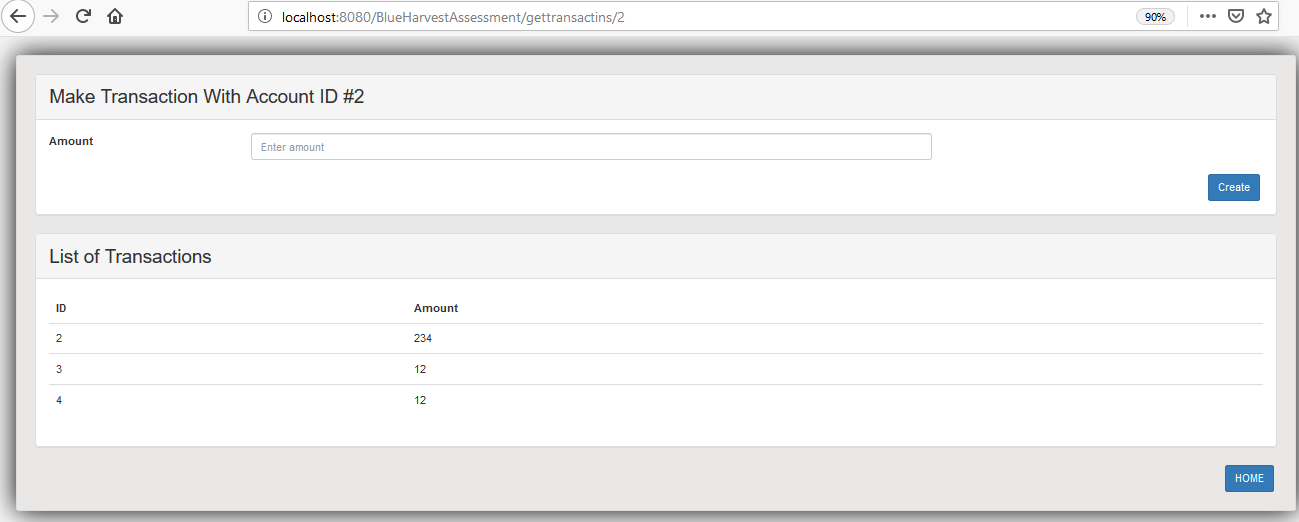
Accounts Page:

You can create account for specific customer with initial balance, and show account transactions



Transaction Page:

You can make and show customer transactions:



# Personal info

Name: Ehab ElKashef

Email: [ehab.elkashef@vodafone.com](mailto:ehab.elkashef@vodafone.com)

Mobile: +201003365705