**Build Instructions:**

1. Download the project and next go to the project directory.

2. run command maven clean install.

3. atmservice.war file is generated in target.

4. Copy the war file to WEBAPPS folder of tomcat server and start the server.

5. Use postman to hit the service.

**Instructions to hit the service with postman:**

**1. create a customer account:**

<http://localhost:8080/atmservice/rest/customeraccounts/hareesh>

**2. Make a deposit for the customer account:**

http://localhost:8080/atmservice/rest/customeraccounts/deposits/{customerId} /{depositAmount}

Example:

<http://localhost:8080/atmservice/rest/customeraccounts/deposits/41785f5b-8230-487f-8744-c511c7078211/100>

**3. Make a withdrawal for a customer account:**

http://localhost:8080/atmservice/rest/customeraccounts/withdraws/{customerId} /{depositAmount}

Example:

<http://localhost:8080/atmservice/rest/customeraccounts/withdraws/8d32a0a9-0579-4d74-847c-81a436647db9/50>

**4. Display the balance for the customer account:**

http://localhost:8080/atmservice/rest/customeraccounts/balances/{customerId}

Example:

<http://localhost:8080/atmservice/rest/customeraccounts/balances/41785f5b-8230-487f-8744-c511c707821>

Things to consider:

**Where would you use GET and POST?**

We use GET for displaying the customer account balance.

**How would you store data?**

I used a hashmap to store the data. The key is customer account Id and the Value is the CustomerAccount.

**Would you need any locking?**

We need locking. If two joint account customers accessing the service at same time then there will be problems.

I used reentrant locks specific to the customer Id to block another customer with same Id trying to access when write is going on.