

# Ecommerce Backend

Mohamad Lakkis

November 29, 2024

## Abstract

abstract, I will write it later.

## 1 Installing required packages

```
ubuntu@laptop:~$ sudo docker pull postgres
Using default tag: latest
latest: Pulling from library/postgres
2d4c90a75ae: Pull complete
323a93b65d1: Pull complete
d5b6d8d27ff: Pull complete
2c40d0e810e: Pull complete
def41c1080d: Pull complete
f4d42e810e: Pull complete
ab0d6e7f80d: Pull complete
f4c0d0e810e: Pull complete
10e479cc17f: Pull complete
bc31f1f4d0e: Pull complete
e0b9724cd0e: Pull complete
e0f7f0d0e810e: Pull complete
4b0f2d4d10e: Pull complete
e0f7f0d0e810e: Pull complete
Digest: sha256:96ac3113c0d0e7d7f11f0c308bcb0b0da09d07eef55474282aef211
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
Docker: permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Head "http://localhost/v2/run/docker.sock?ping": dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
ubuntu@laptop:~$ sudo docker run --name customers-db -e POSTGRES_USER=admin -e POSTGRES_PASSWORD=adminpass -e POSTGRES_DB=customers_db -p 5432:5432 -d postgres
7f4c2c0a21b0ed408b409272739c7b2095e3b2c2d79f25a397b07
ubuntu@laptop:~$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED    STATUS    PORTS    NAMES
7f4c2c0a21b    postgres  "docker-entrypoint.s..." 19 seconds ago    Up 18 seconds    0.0.0.0:5432->5432/tcp, :::5432->5432/tcp    customers-db
ubuntu@laptop:~$
```

Figure 1: Installing postgresQL image from docker hub, and running it

```
ubuntu@laptop:~$ psql -h localhost -U admin -d customers_db
Password for user admin:
psql (14.13 (Ubuntu 14.13-0ubuntu2.22.04.1), server 17.2 (Debian 17.2-1.pgdip20h-1))
WARNING: psql major version 14, server major version 17.
Some psql features might not work.
Type "help" for help.

customers_db=# CREATE TABLE customers (
id SERIAL PRIMARY KEY,
full_name VARCHAR(100) NOT NULL,
username VARCHAR(50) UNIQUE NOT NULL,
password VARCHAR(100) NOT NULL,
age INT NOT NULL,
address TEXT NOT NULL,
gender VARCHAR(20) NOT NULL,
marital_status VARCHAR(20) NOT NULL,
wallet_balance NUMERIC DEFAULT 0.0
);
CREATE TABLE
customers_db=# \dt
List of relations
Schema | Name      | Type  | Owner
-----+-----+-----+-----
public | customers | table | admin
(1 row)

customers_db=# SELECT * FROM customers;
 id | full_name | username | password | age | address | gender | marital_status | wallet_balance
----+-----+-----+-----+-----+-----+-----+-----+-----
(0 rows)

customers_db=# \q
ubuntu@laptop:~$
```

Figure 2: Creating a database(for now the now for customers ONLY) in postgresQL and checking if it is created, by running a query on it

## 2 Database Details (point 3 with additional details)

### 2.1 customers Database

- **Host:** localhost
- **Port:** 5432
- **Database Name:** customers\_db
- **User:** admin
- **Password:** adminpass

## 3 API Design for customer Service, and testing using postman

*Note: For the details: for each API call write an example, description and comments for the fields(i.e. point 5 of the requirements check this file:) [api\\_endpoints.md](#)*

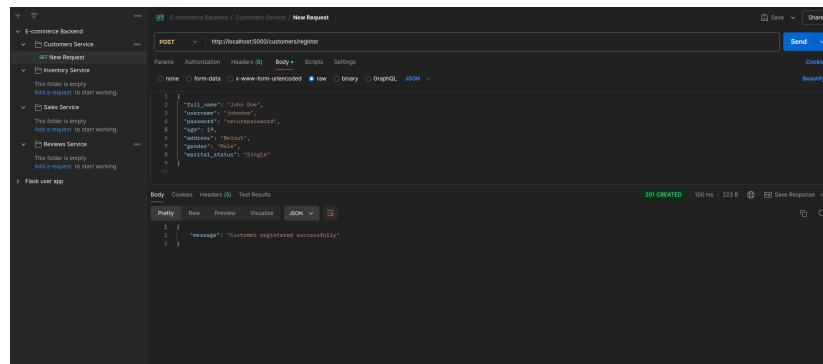


Figure 3: Customer registration

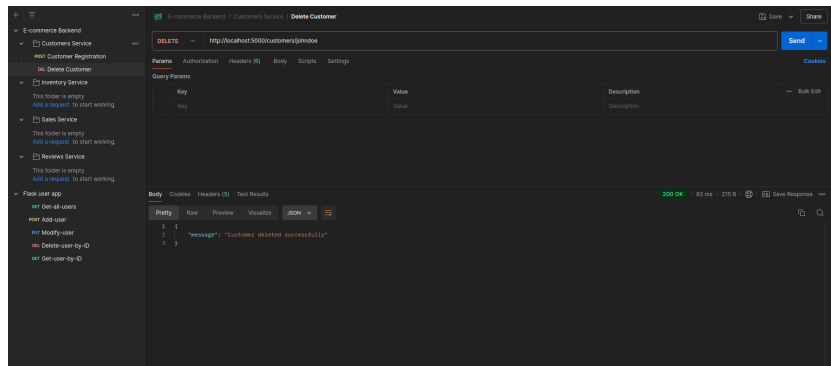


Figure 4: Delete customer

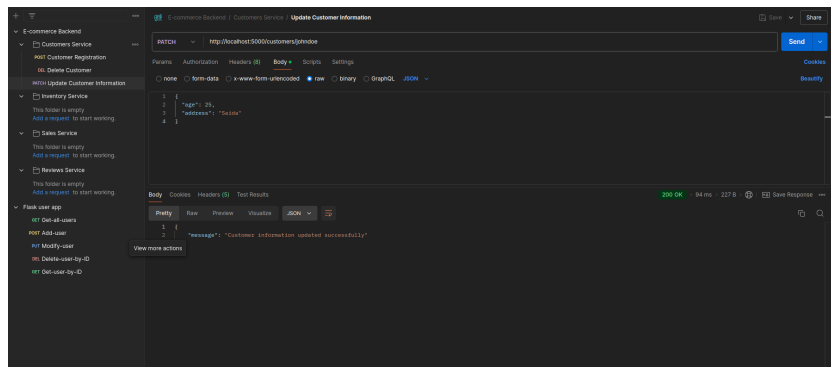


Figure 5: Update customer information

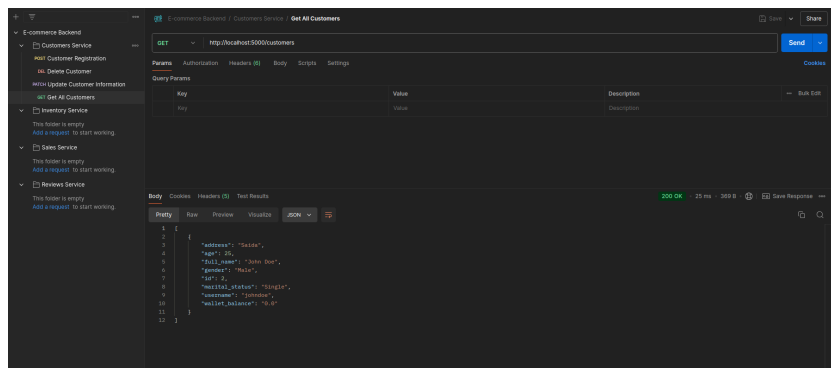


Figure 6: Get all customers

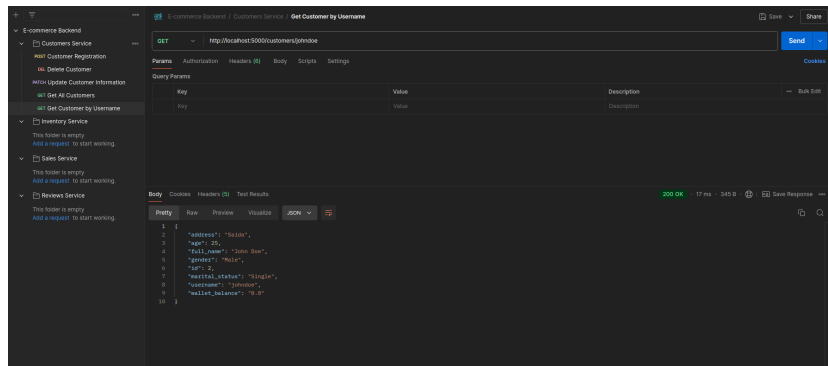


Figure 7: Get customer per username

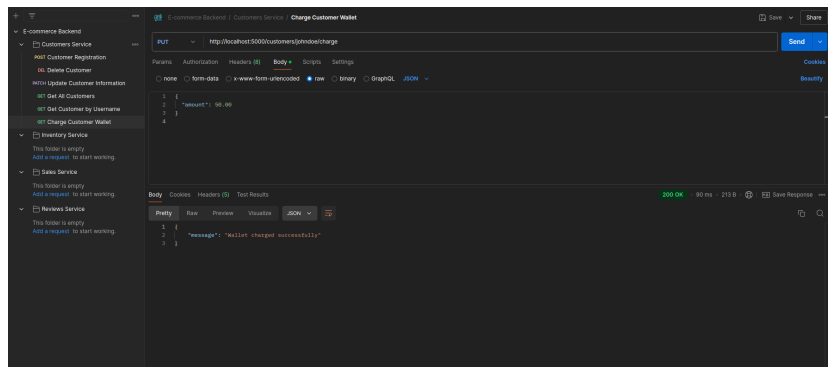


Figure 8: Charge customer wallet in dollars

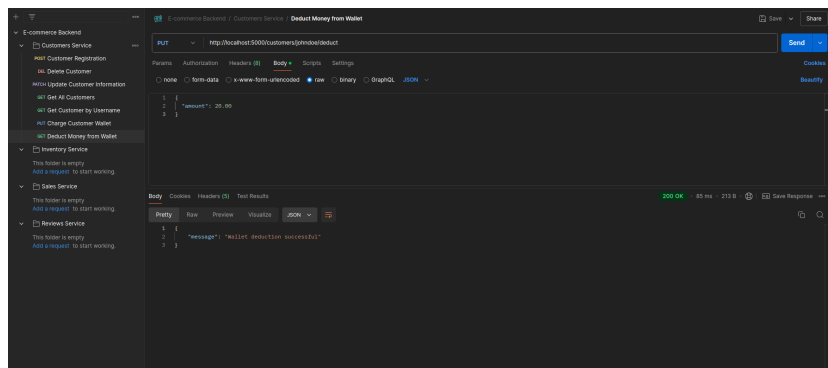


Figure 9: Deduct money from the wallet