

Project Zero

For Project 0, I've made a basic database for managing the types of coffee in for a coffee shop.

To execute the program:

1. go inside project folder and type: ***npm install*** in the command line then press enter
2. once installed, type and run: ***npm run dev*** to start the server
3. to end server press ***ctrl+c***
4. to run tests type and enter: ***npm run test*** in the command line
5. to run coverage test type and enter: ***npm run test:coverage*** in the command line

Technologies used:

Scripting Language:

- Typescript/Javascript

Runtime environment :

- NodeJS

Developer environment :

- Nodemon

Server Framework:

- Express-js

Database manager:

- PostgreSQL

Testing Framework:

- Jest
- SuperTest

Logging Framework:

- Express Middleware
- Morgan Logger
- Logging.ts (self-made)

Table Definitions:

Coffee Table:

```
CREATE TABLE Coffee(  
  id SERIAL PRIMARY KEY,  
  coffee_name varchar(30),  
  shots_of_espresso integer,  
  container varchar(30),  
  total_calories integer,  
  contains_milk boolean,  
  price numeric(20, 2)  
);
```

Users Table:

```
CREATE TABLE Users(  
  user_email varchar(30) PRIMARY KEY,
```

```
user_name varchar(30),
user_password varchar(30),
user_role varchar(30)
);
```

Insert test data inside the tables:

```
INSERT INTO Coffee(coffee_name, shots_of_espresso, container, total_calories,
contains_milk, price)
VALUES ('Espresso', 1, 'Espresso Cup', 3, FALSE, 3.0),
('Double Espresso', 2, 'Espresso Cup', 6, FALSE, 4.0),
('Red Eye', 1, 'Coffee Mug', 0, FALSE, 3.5),
('Black Eye', 2, 'Coffee Mug', 0, FALSE, 4.5),
('Americano', 1, 'Coffee Mug', 15, FALSE, 4.5);
```

```
INSERT INTO Users(user_email, user_name, user_password, user_role)
VALUES ('paul@gmail.com', 'Paul', '1234', 'Admin'),
('john@gmail.com', 'John', '5678', 'User');
```

TEST ROUTES:

Log-in:

localhost:5000/auth/

```
{
  "user_email": "john@gmail.com",
  "user_password": "5678"
}
```

```
{
  "user_email": "paul@gmail.com",
  "user_password": "1234"
}
```

get all:

localhost:5000/coffee

search by id:

localhost:5000/coffee/3

search by keyword:

localhost:5000/coffee/search/container/Espresso Cup

save coffee item:

localhost:5000/coffee/ then POST, body, then json

```
{
  "coffee_name": "Long Black",
  "shots_of_espresso": 2,
```

```
    "container": "Coffee Mug",
    "total_calories": 3,
    "contains_milk": false,
    "price": "4.00"
}
```

update coffee item:

```
{
  "coffee_name": "Short Black",
  "shots_of_espresso": 1,
  "container": "Espresso Cup",
  "total_calories": 1,
  "contains_milk": false,
  "price": "2.00"
}
```

delete coffee item:

localhost:5000/coffee/6 **then** DELETE
