**Setup**

1. Setup Docker and Docker compose in your machine
2. Navigate to “labela\_backend\_assignment” folder and run following command to run the containers. (if you are using an older docker compose version use “docker-compose” instead of “docker compose”)
   1. “docker compose up -d”
3. The backend (service: web) will run on port 8000 and the postgresql database (service : db) will run on port 5432
4. To migrate data, run command in terminal. Make sure the containers are up and running
   1. When you run the system for the first time, make a data migration first
      1. “docker compose exec web python manage.py makemigrations”
   2. “docker compose exec web python manage.py migrate”
5. After the migration, re-start the containers with following command.
   1. “docker compose restart”
6. To make a super user to access docker admin panels, use following command and follow the guidelines given by the prompt.
   1. “docker compose exec web python manage.py createsuperuser”

**Run Testing**

* + - 1. To run the testcases, use following command, and make sure all the containers are up and running
         1. “docker compose exec web python manage.py test autocompany”

**Usage**

Create a super user to access the db via Django admin (ref : setup 6.)

User creation and login routs doesn’t require authentication headers, so first create a user (ref API Reference 2.) and then Login with created user credentials (ref: API Reference 3.) to get a refresh and access tokens

All other routes required to authenticated, so include the access token as a “Bearer” token in the request headers.

Find the “Gapstars.postman\_collection.json” file and import it to Postman to have all the route example requests.

**API Reference**

Send all request data in application/JSON format

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Type | URL | Description | Sample Request data JSON |
| 1 | GET | http://localhost:8000/users/users\_list | Return all clients | {} |
| 2 | POST | http://localhost:8000/users/signup | Register new client |  |
| 3 | POST | Localhost:8000/users/login | Login user |  |
| 4 | GET | http://localhost:8000/api/parts | Return all parts | {} |
| 5 | POST | http://localhost:8000/api/parts | Add new part to the system |  |
| 6 | GET | http://localhost:8000/api/parts/<int:id> eg: http://localhost:8000/api/parts/1 “the id is the part id” | Get details of a specific part | {} |
| 7 | GET | http://localhost:8000/api/cart | Get items and details of user’s shopping cart |  |
| 8 | PUT | http://localhost:8000/api/cart | Add new item to the cart |  |
| 9 | DELETE | http://localhost:8000/api/cart | Remove item from cart |  |
| 10 | POST | http://localhost:8000/api/purchase | Purchase items in current cart |  |
| 11 | GET | http://localhost:8000/api/orders | Get all the orders of certain user |  |
| 12 | POST | http://localhost:8000/api/orders/update | Update delivery date and time of the order  p.s date format: Month <space> Date <space> Year  Time format: “H:M <space> AM/PM |  |
| 13 | POST | http://localhost:8000/users/token\_refresh | Get new access token when the access token is espire |  |