

Assignment Documentation

Two apps created under **autocompany** project, **car_part_sale** and **api**

In **car_part_sale**, it has all the models needed for this assignment and under **api** we have the views, serializers and urls.

With docker-composer,

```
docker-compose up -d --build
```

Then,

```
$ docker-compose exec app python manage.py migrate
```

To Create a superuser,

```
$ docker-compose exec app python manage.py createsuperuser
```

As a native or / Docker,

1. Set up the DB configuration.
2. Apply the necessary commands makemigrations, migrate
3. Create a superuser
4. Login to admin panel to see the table structure and create the necessary data or, use following API calls,
 - <http://localhost:8000/api-auth/login/>
Login to api via the browser , or via the flowing post call generate a valid token

OR

Via postman

[POST] <http://localhost:8000/api/token/>

```
{  
  "username": "<Created Username>",  
  "password": "<Created Password>"  
}
```

Above call will return a token,

Once you have the bearer token , use it to call other API calls.

5. If you need more users, you can add via the API,
- [POST] <http://localhost:8000/api/users/register>

```
{  
  
  "username" : "testUser",  
  "email" : "testUser@gmail.com",  
  "password" : "123",  
  "password2" : "123"  
  
}
```

6. You can list down users, ### added this to just to check against multiple users.

- [GET] <http://localhost:8000/api/users/>

7. Before Creating car parts , We Need to create a category for car parts.

- [POST] <http://localhost:8000/api/category/create>

```
{  
  "slug": "wheel",  
  "name": "wheel, Tyre",  
  "description": "Before we get into the types of wheels, you should familiarize  
yourself with the various components of the wheel, which include three main  
parts: the tire, the rim, and the hub. The tire, which is the width of the wheel, goes  
around the rim and gives the wheel grip on the road surface."  
}
```

- You can list down all the created categories ,
[GET] <http://localhost:8000/api/categories/>
- Detail view of a category,
[GET] <http://localhost:8000/api/category/<id>>
- Update category,
[POST] <http://localhost:8000/api/category/update/<id>>

```
{  
  "slug": "glass",  
  "name": "glass",  
  "description": "window, wiper, Updated description "  
}
```

- Delete category,
[DELETE] <http://localhost:8000/api/category/delete/<id>>

8. We have to add **Parts** via the admin page, then we can query them with the following api endpoint, - (#to save time I didn't add this to the api.)

- [GET] <http://localhost:8000/api/parts/>
- Detail view of a part,
[GET] <http://localhost:8000/api/part/details/<id>>

```
{
  "id": 1,
  "category_name": "wheel, Tyre",
  "slug": "tyre",
  "name": "Tyre",
  "model": "CEAT",
  "description": "good condition , Brand new",
  "price": "5500.00",
  "stock": 98,
  "created": "2023-09-22T20:43:17.680036Z",
  "Category": 1
}
```

9. Now we have parts, so we can create a cart now,

- You can add parts and number of quantity you want(use part id),

[POST] <http://localhost:8000/api/cart/create/>

```
{
  "part": "1",
  "part_qty": 1
}
```

It will add like following,

```
{
  "id": 11,
  "user": "Janitha",
  "part": "Tyre",
  "part_qty": 1,
  "created_at": "2023-09-23T18:26:34.904471Z"
}
```

- You can add any valid parts to the cart.

10. To see what's on your cart, you can call the following api endpoint,

- [GET] <http://localhost:8000/api/cart/create/>

```
[
  {
    "id": 11,
    "user": "Janitha",
    "part": "Tyre",
    "part_qty": 1,
    "created_at": "2023-09-23T18:26:34.904471Z"
  },
  {
    "id": 12,
    "user": "Janitha",
    "part": "leather seat cover",
    "part_qty": 40,
    "created_at": "2023-09-23T18:30:18.526044Z"
  }
]
```

11. Delete Part(Item) from the cart,

- [DELETE] <http://localhost:8000/api/cart/delete/<id>>

12. Once we are done with adding Parts(items) to the cart, we can create an order,

- Create an Order

[POST] <http://localhost:8000/api/order/create/>

```
{
  "name": "De Silva",
  "email": "test@gmail.com",
  "address": "232, Galle Road, Gorakana, Panadura",
  "payment_mode": "Card",
}
```

```
"delivery_date": "2023-10-2T13:56:59Z"
```

```
}
```

- Delivery_date - Date time field we can define a date and time for the delivery.
- Once we create the order, the cart will be empty.
- You can list down the your order list,
[GET] <http://localhost:8000/api/orders/>

- You can view single order by order id,
[GET] <http://localhost:8000/api/order/><order id>

```
{  
  "id": 8,  
  "user": "Janitha",  
  "name": "De Silva",  
  "email": "test@gmail.com",  
  "address": "232, Galle Road, Gorakana, Panadura",  
  "total_price": 5500.0,  
  "payment_mode": "Card",  
  "delivery_date": "2023-10-02T13:56:59Z",  
  "status": "Completed",  
  "tracking_no": "514a0e1203bf_8522611",  
  "created_at": "2023-09-23T18:37:45.064607Z",  
  "updated_at": "2023-09-23T18:59:20.406097Z"  
}
```

- Once we are done with the payments we/system can only update the delivery status of the order,
- Once Order shipped
[GET] <http://localhost:8000/api/order/update/><order id>/shipped
- Once Order Delivered
[GET] <http://localhost:8000/api/order/update/><order id>/delivered

** Based on the quantity Items are available to order. Once it is ordered **Part** quantity will be reduced.