System Design Document

Overview

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, processing logic, and external interfaces.

# *system requirements*

Your company has been hired by a large car manufacturer to develop software to handle various processes in their business. There are currently five different car models with 3 variations of each model with future cars designs being developed.

The group project of CPSC463 build, secure, and test an effective RESTful API using Python and Flask, HTML5, the Python microframework. Find out how to use Flask with Python to approach database access, authentication, and other common tasks.

## operating environment

* Using PyCharm to create a Flask project, and requires the below packages been installed:

Flask-JWT-Extended 3.24.1 3.24.1

Flask-SQLAlchemy 2.4.1 2.4.1

flask-marshmallow 0.11.0 0.11.0

* Testing with Postman
* Communicating JSON
* Using with SQLITE databases and requires the below tools been installed:

DB Browser for SQLite

## Functionality Summary

### Manufacturing FacilitiesMicroservice

**The manufacturing facilities (the plants that build the cars) are losing track of parts and materials and need a software tool to track all the car parts and materials (e.g. glass, bolts, chairs, steel) that have been purchased and received from suppliers. They may have over 100 different suppliers.**

**The following data associated with them:**

* Order Number
* Part number
* Part name
* Supplier name
* Order Status (Supplier preparing, shipping, delivery)
* Order date time
* Shipping date time

**The following operations will be exposed:**

* Create a new order
* Find a order by Order Number
* Delete an existing order by Order Number

The data for the user can be in the same database or different database as the other services.

### **Car dealerships** microservice

**Car dealerships need new software to order new cars from the car manufacturer. The dealerships only have so much space in their lots, depending on how big of a dealership they own.**

**The following data associated with them:**

* Order Number
* Car Make
* Model
* VIN
* Color
* Dealership name
* Order Status (Supplier preparing, shipping, delivery)
* Order date time
* Shipping date time

**The following operations should be exposed:**

* Create a new order
* Find a order by Order Number
* Delete an existing order

When retrieving lists of posts, do not include the text or resource URL for the post.

### **Retail store microservice**

**Repair shops, auto stores, and dealerships need to be able to order parts from the car manufacturer to keep their stock from running out.**

**The following data associated with them:**

* Order Number
* Part number
* Part name
* Auto store name
* Order Status (Supplier preparing, shipping, delivery)
* Order date time
* Shipping date time

**The following operations should be exposed:**

* Create a new order
* Find a order by Order Number
* Delete an existing order

### **Customer order car microservice**

**Customers need a way to order new and used cars directly from the car manufacturer that can then be picked up at a local dealership.**

**The following data associated with them:**

* Order Number
* Car Make
* Model
* VIN
* Color
* Customer name
* Order Status (Supplier preparing, shipping, delivery)
* Order date time
* Shipping date time

**The following operations should be exposed:**

* Create a new order
* Find a order by Order Number
* Delete an existing order

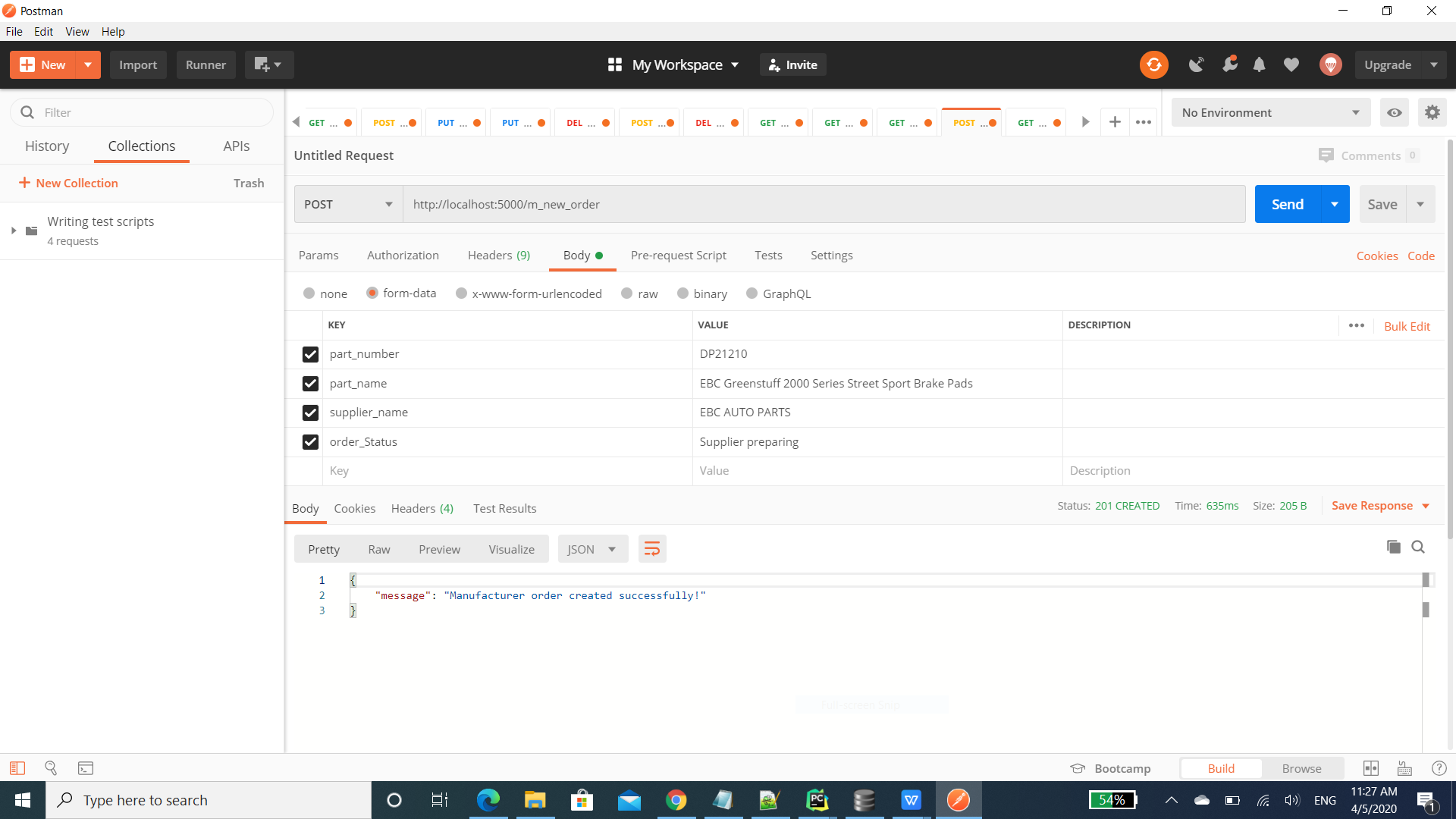
# SYSTEM ARCHITECTURE

## System Software Architecture

# Interfaces

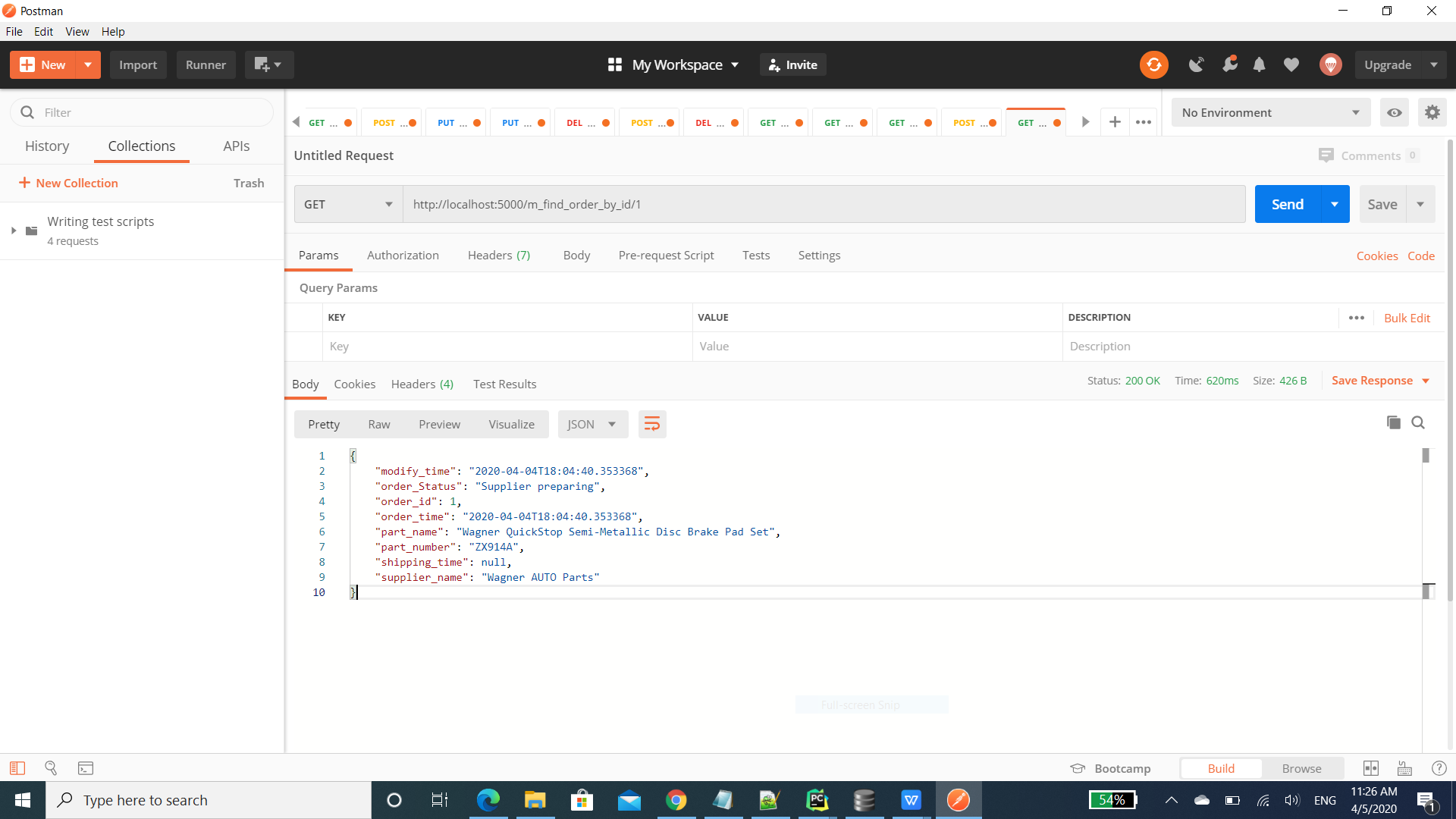
## Manufacturing Facilities Create a new order

|  |  |
| --- | --- |
| Request URL | http://localhost:5000/m\_new\_order |
| Usage | **The manufacturing facilities create** a new ORDER |
| Request Method | post |
| Parameters | **'order\_id'**, **'part\_number'**, **'part\_name'**, **'supplier\_name'**, **'order\_Status'**, **'order\_time'**, **'shipping\_time'**, **'modify\_time'** |
| Return | Fail (message=**'Manufacturer order**  **created unsuccessfully!'**), 409  Success (message=**'Manufacturer order**  **created successfully!'**), 201 |



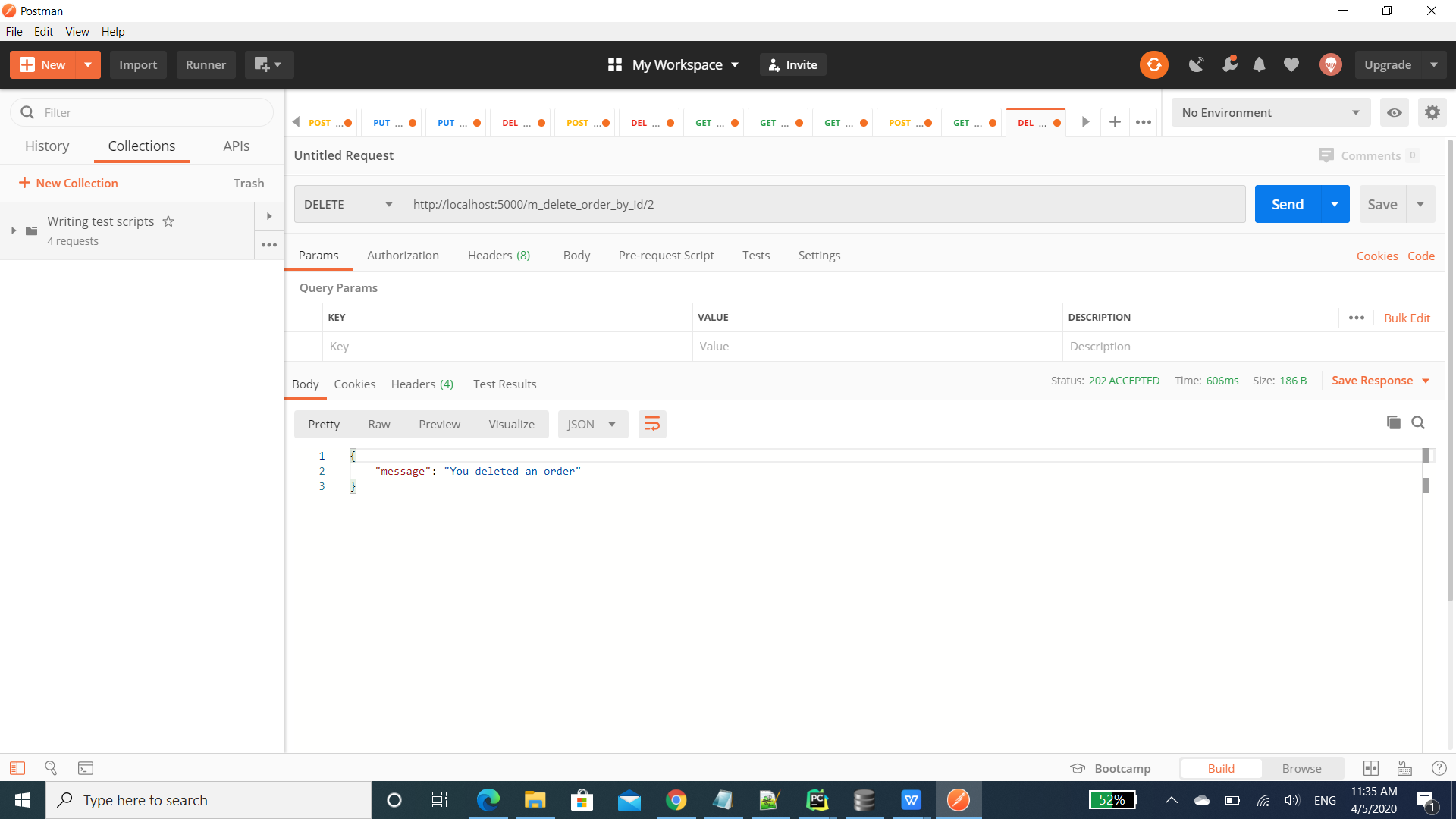
## Manufacturing Facilities Find an order by order id

|  |  |
| --- | --- |
| Request URL | http://localhost:5000/m\_find\_order\_by\_id/<int:id> |
| Usage | **The manufacturing facilities find an** ORDER by order id |
| Request Method | Get |
| Parameters | **'order\_id'** |
| Return | Fail (**'That order does not exist'**), 404  Success (message= post) |



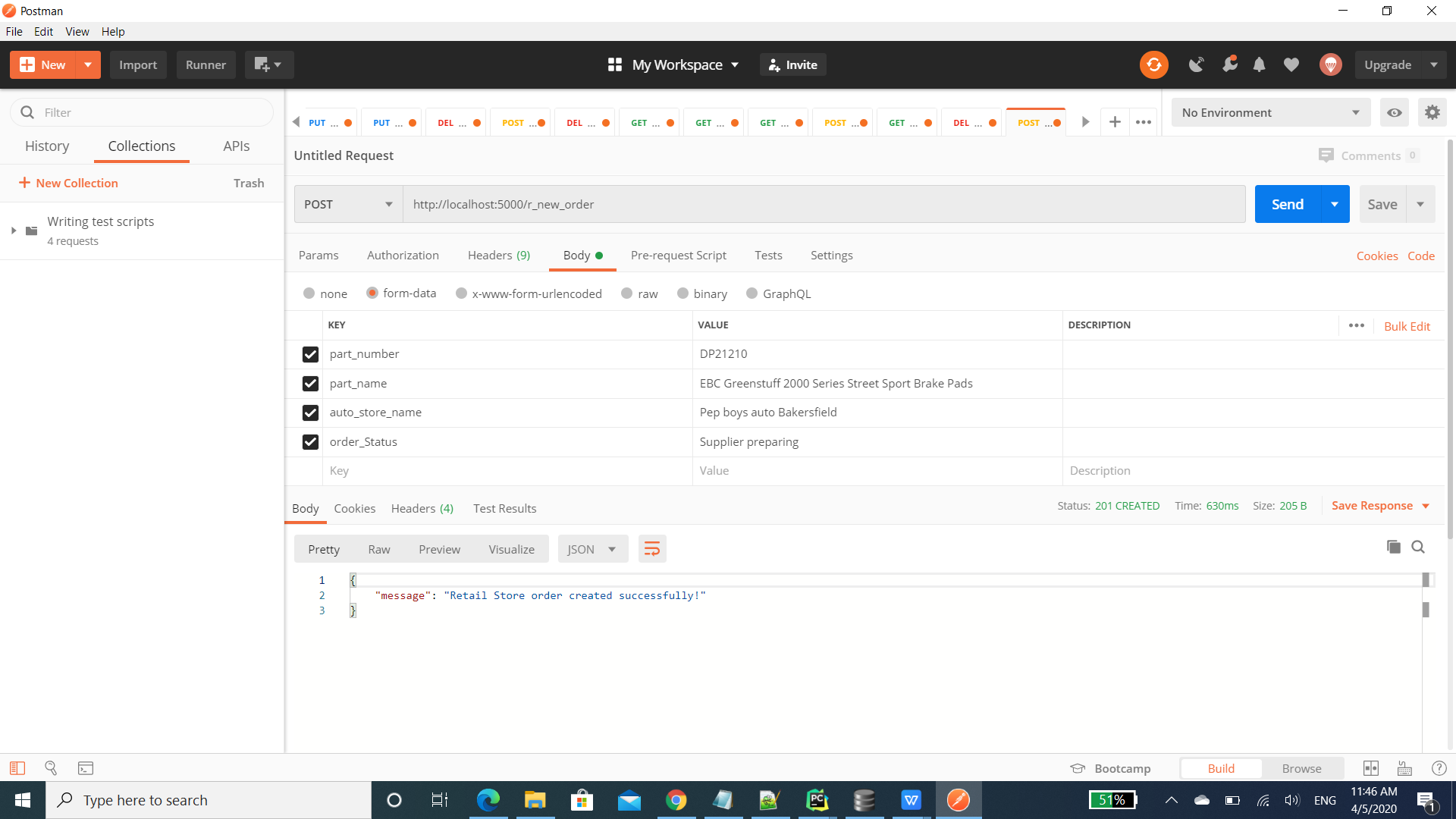
## Manufacturing Facilities Delete an existing order by Order Number

|  |  |
| --- | --- |
| Request URL | http://localhost:5000/m\_delete\_order\_by\_id/<int:id> |
| Usage | **The manufacturing facilities** Delete an exiting order |
| Request Method | delete |
| Parameters | **'id'** |
| Return | Fail (**'That order does not exist'**), 404  Success (message=**'You deleted a order**), 202 |



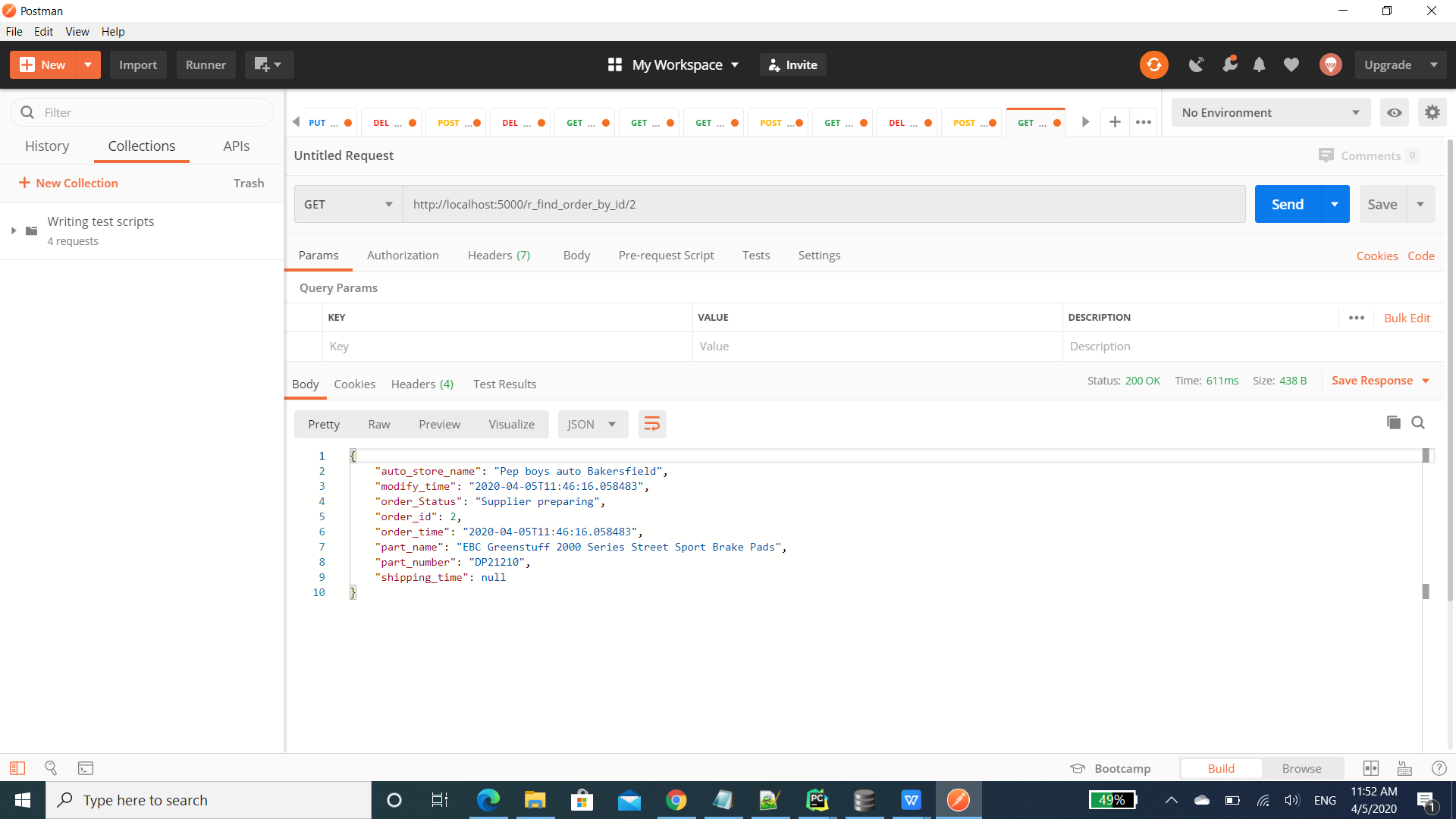
## **Retail store** Create a new order

|  |  |
| --- | --- |
| Request URL | http://localhost:5000/r\_new\_order |
| Usage | **Create** a new ORDER |
| Request Method | post |
| Parameters | **'order\_id'**, **'part\_number'**, **'part\_name'**, **'**auto\_store\_name**'**, **'order\_Status'**, **'order\_time'**, **'shipping\_time'**, **'modify\_time'** |
| Return | Fail (message=**'Retail store order**  **created unsuccessfully!'**), 409  Success (message=**'Retail store order**  **created successfully!'**), 201 |



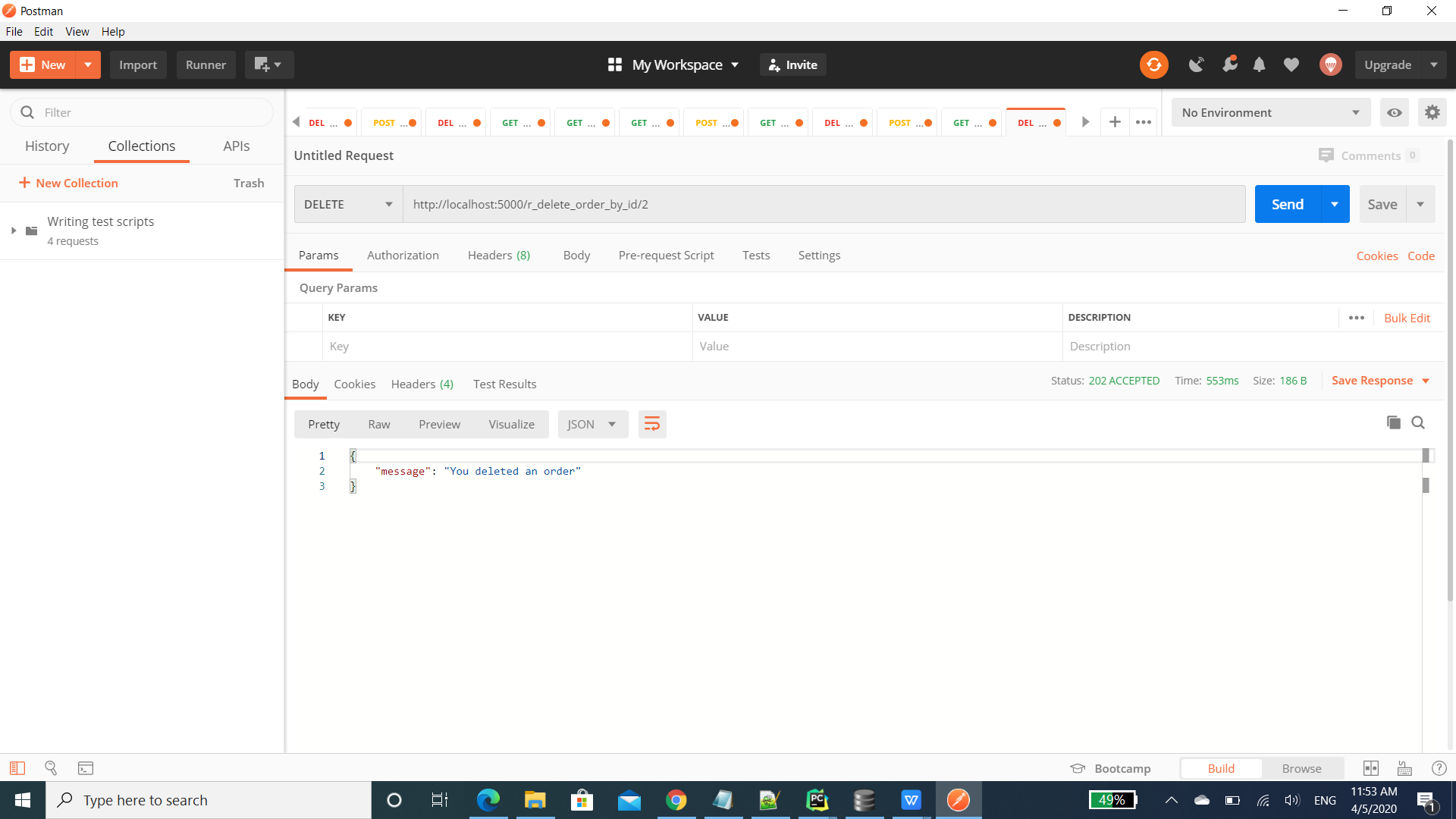
## **Retail store** Find an order by order id

|  |  |
| --- | --- |
| Request URL | http://localhost:5000/r\_find\_order\_by\_id/<int:id> |
| Usage | **find an** ORDER by order id |
| Request Method | Get |
| Parameters | **'order\_id'** |
| Return | Fail (**'That order does not exist'**), 404  Success (message= post) |



## **Retail store** Delete an existing order by Order Number

|  |  |
| --- | --- |
| Request URL | http://localhost:5000/r\_delete\_order\_by\_id/<int:id> |
| Usage | Delete an exiting order |
| Request Method | delete |
| Parameters | **'id'** |
| Return | Fail (**'That order does not exist'**), 404  Success (message=**'You deleted a order**), 202 |



# DATABASE DESIGN

Add new table TB\_Manufacture, TB\_Retail\_Store.

## TB\_Manufacture Data Description

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TABLE NAME** | **FIELD** | **TYPE** | **NULL** | **DEFAULT** |
| TB\_Manufacture | ORDER\_ID | INT(11) | NOT NULL | AUTO |
|  | Part\_number | VARCHAR | NOT NULL | NOT |
|  | Part\_name | VARCHAR | NOT NULL | NOT |
|  | Supplier\_name | VARCHAR | NOT NULL | NOT |
|  | Order\_Status | VARCHAR | NOT NULL | Supplier preparing  (Supplier preparing, shipping, delivery) |
|  | Order\_date\_time | DATETIME | NOT NULL | AUTO\_NOW |
|  | Shipping\_date\_time | DATETIME |  |  |
|  | MODIFY\_TIME | DATETIME | NOT NULL | NOT |
|  |  |  |  |  |
|  |  |  |  |  |
| TB\_Retail\_Store | ORDER\_ID | INT(11) | NOT NULL | AUTO |
|  | Part\_number | VARCHAR | NOT NULL | NOT |
|  | Part\_name | VARCHAR | NOT NULL | NOT |
|  | Auto\_store\_name | VARCHAR | NOT NULL | NOT |
|  | Order\_Status | VARCHAR | NOT NULL | Supplier preparing  (Supplier preparing, shipping, delivery) |
|  | Order\_date\_time | DATETIME | NOT NULL | AUTO\_NOW |
|  | Shipping\_date\_time | DATETIME |  |  |
|  | MODIFY\_TIME | DATETIME | NOT NULL | NOT |

# EXTERNAL INTERFACES