**Service Project Documentation**

Bassem Osama , Mohamed Sherif , Ziad Ahmed , Ibrahem ezzat, Mohamed Haitham , Adham Shible

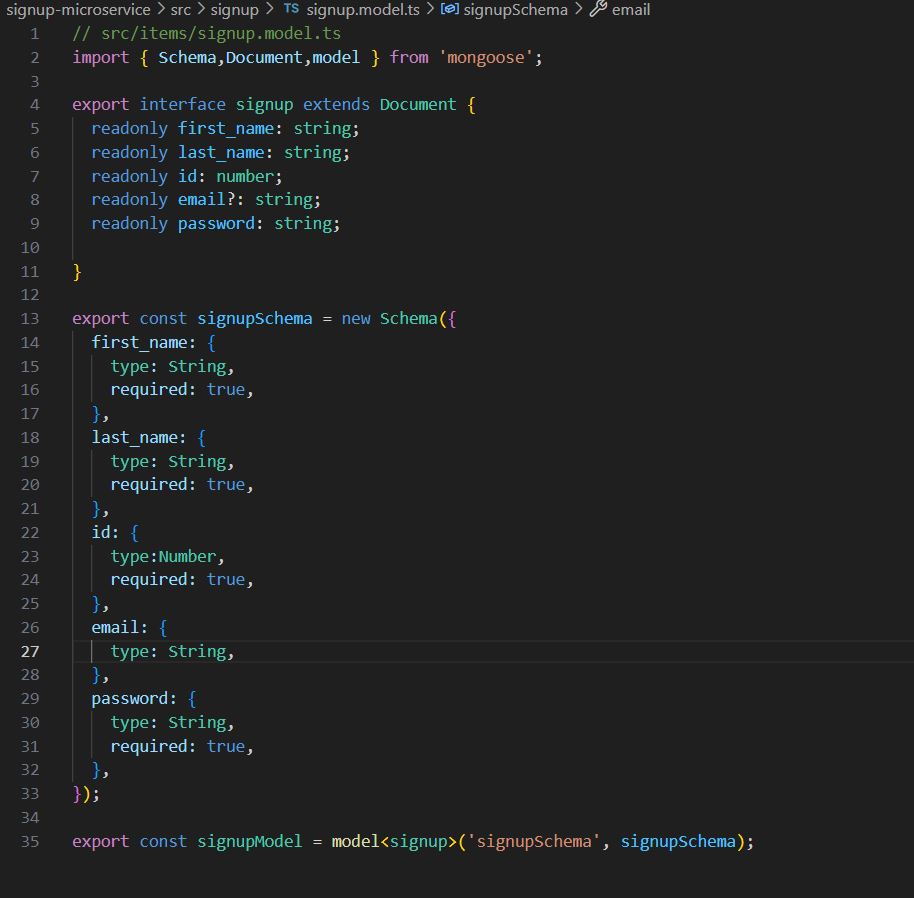
**Vehicles Inventory Management by Microservices**

(6 microservices)

**(1)**Signup Microservices

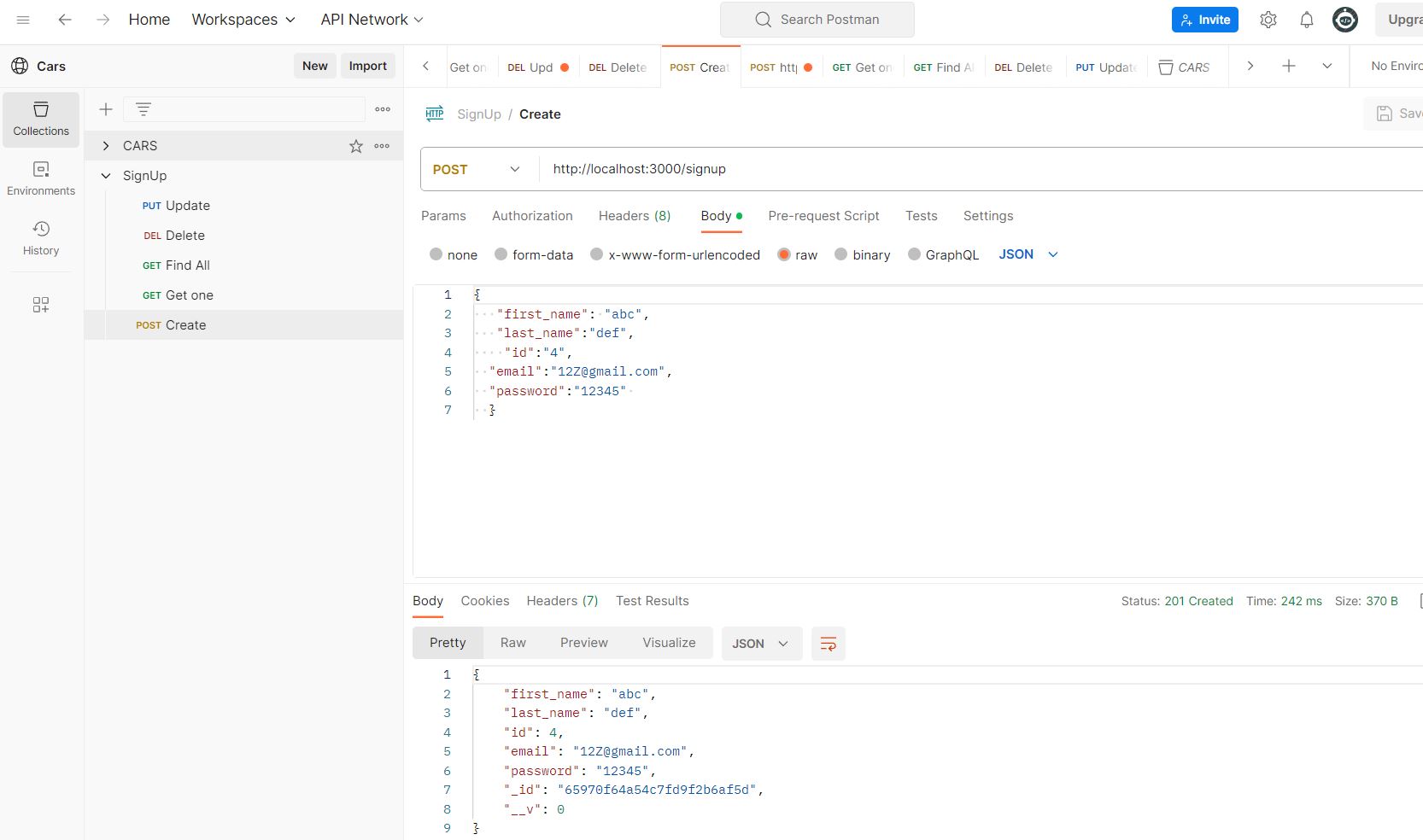
(i)Signup Microservice Schema

<https://schema.getpostman.com/json/collection/v2.1.0/collection.json>



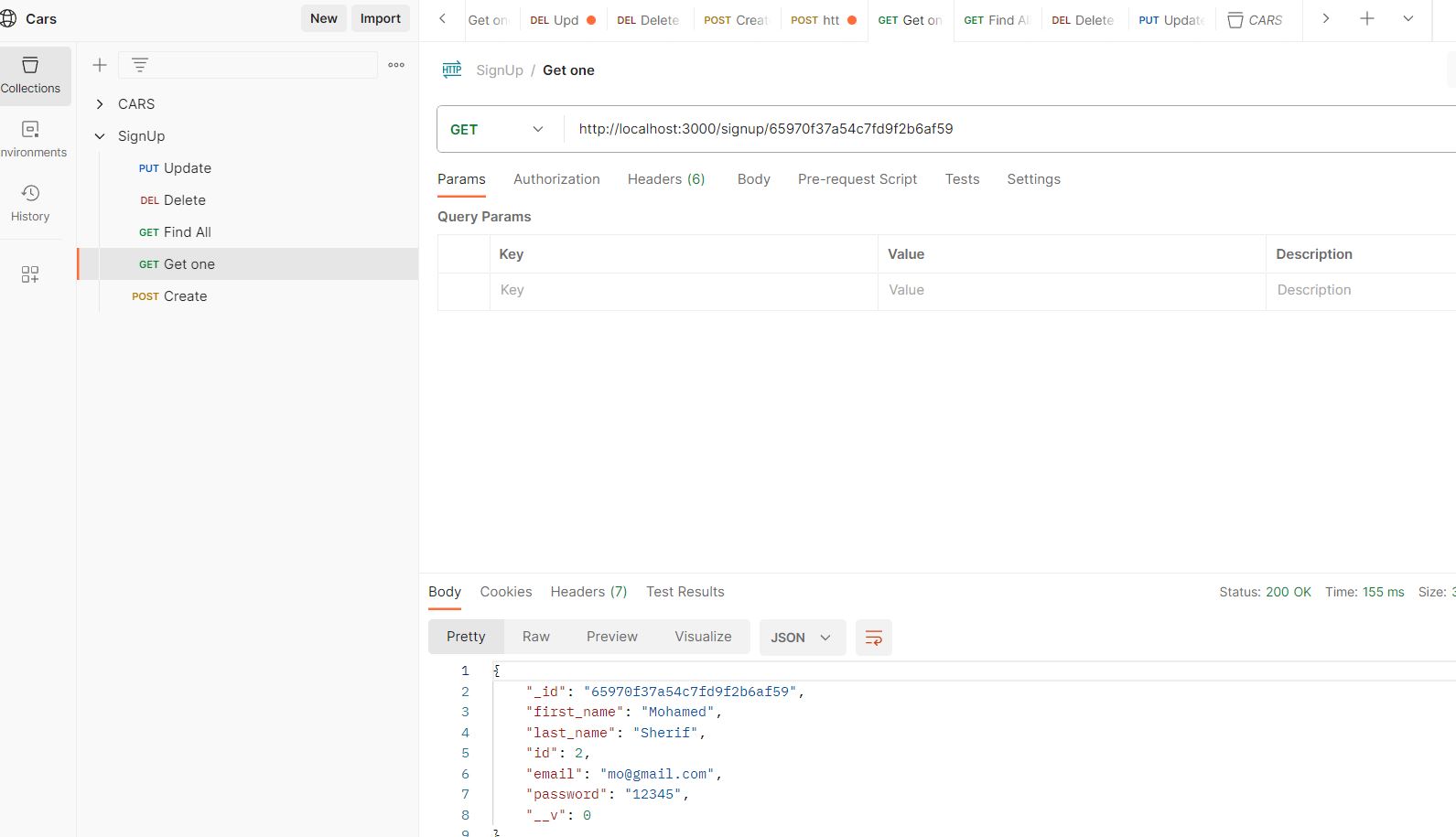
Here the Signup Schema it Contains ‘First name,Last name,Id,Email,password’ All of these it's a type of string that can be taken from the user and we got it in the database by POST ,GET , DELETE and PUT.

(ii)Signup POSTMAN test.

POST 

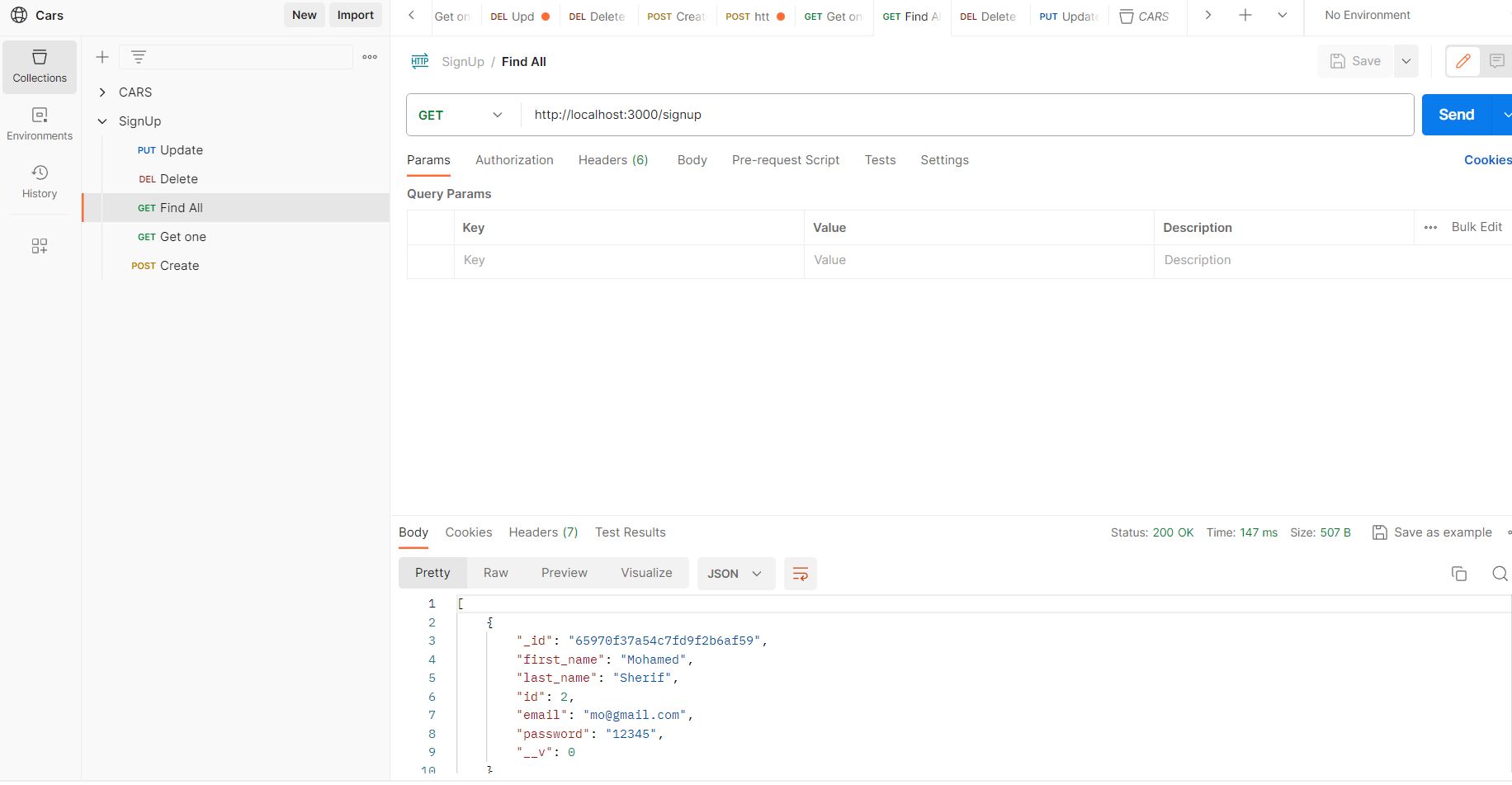
<http://localhost:3000/signup>

GET ONE by id



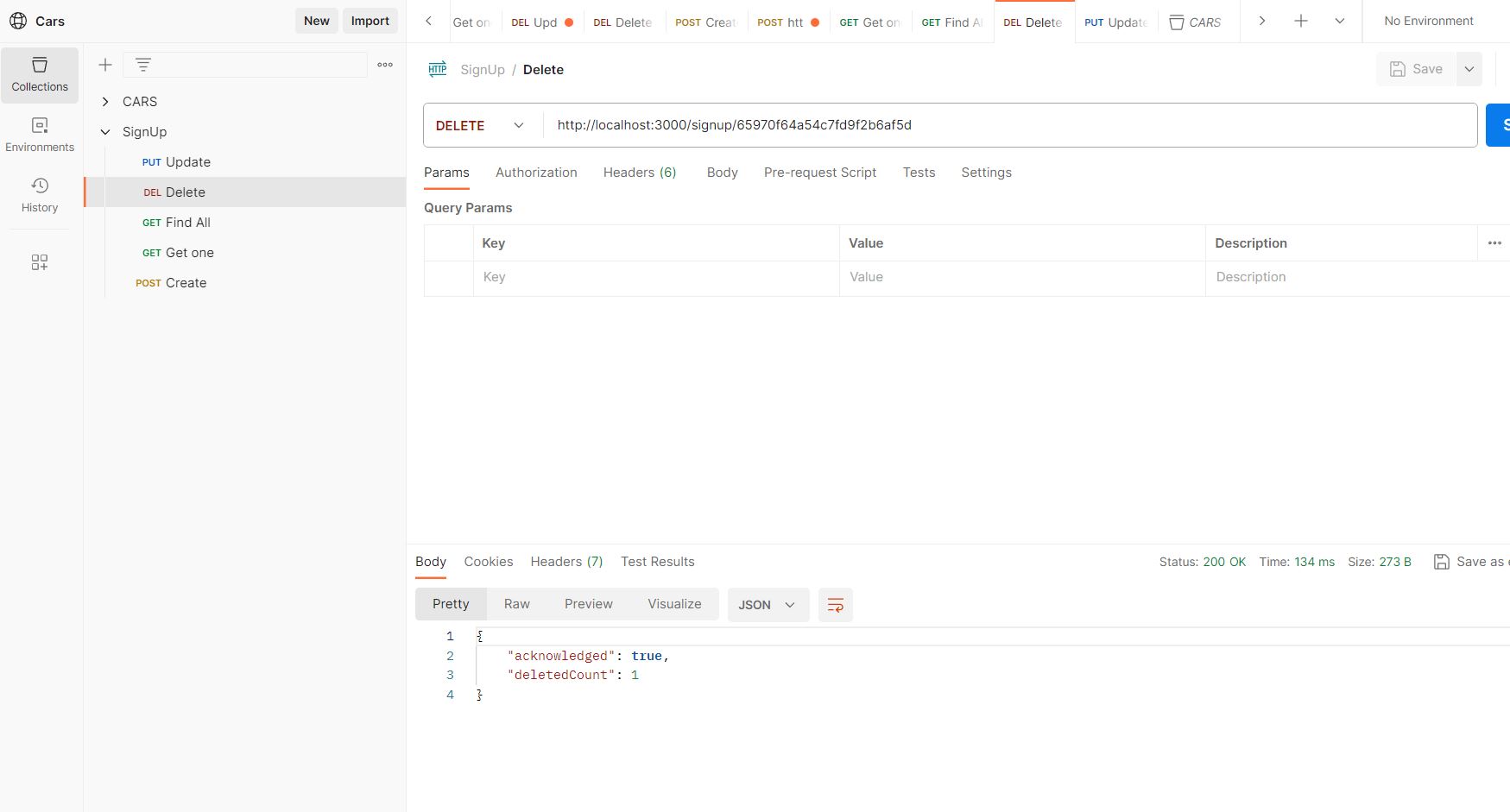
<http://localhost:3000/signup/65970f37a54c7fd9f2b6af59>

GET ALL



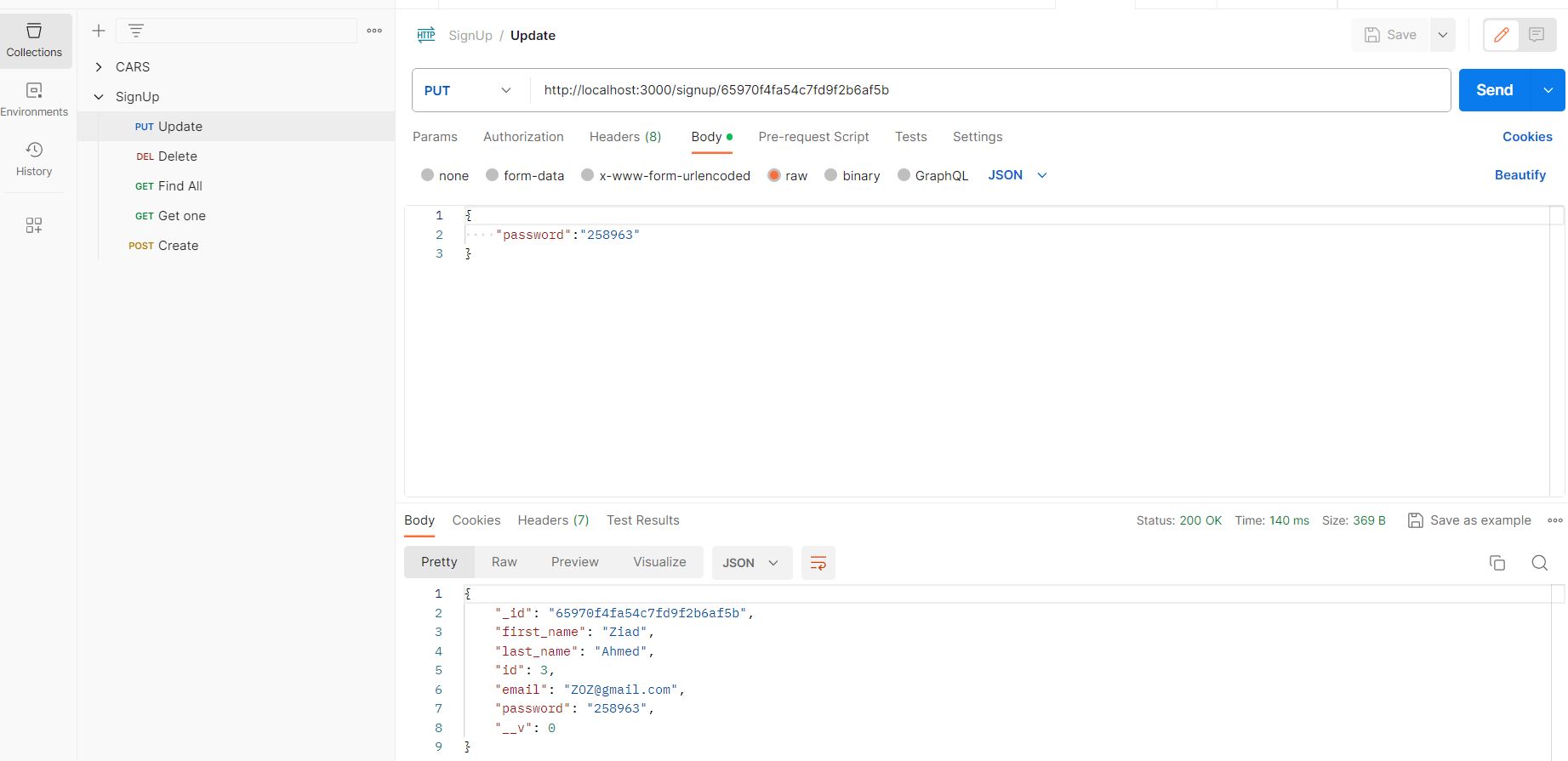
<http://localhost:3000/signup>

DELETE by id



<http://localhost:3000/signup/65970f64a54c7fd9f2b6af5d>

UPDATE(PUT)

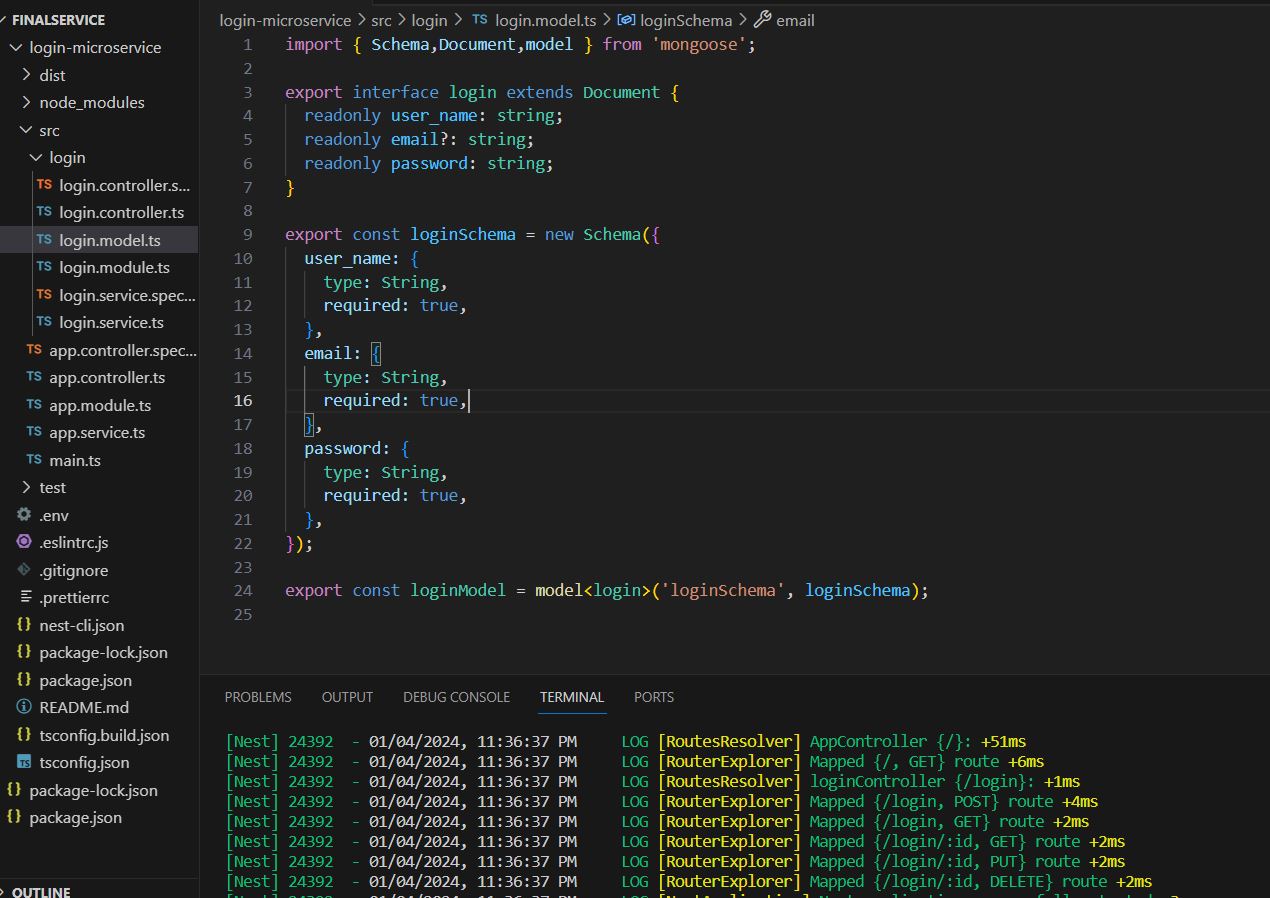


<http://localhost:3000/signup/65970f4fa54c7fd9f2b6af5b>

**(2)**Login Microservices

(i)Login Microservice Schema

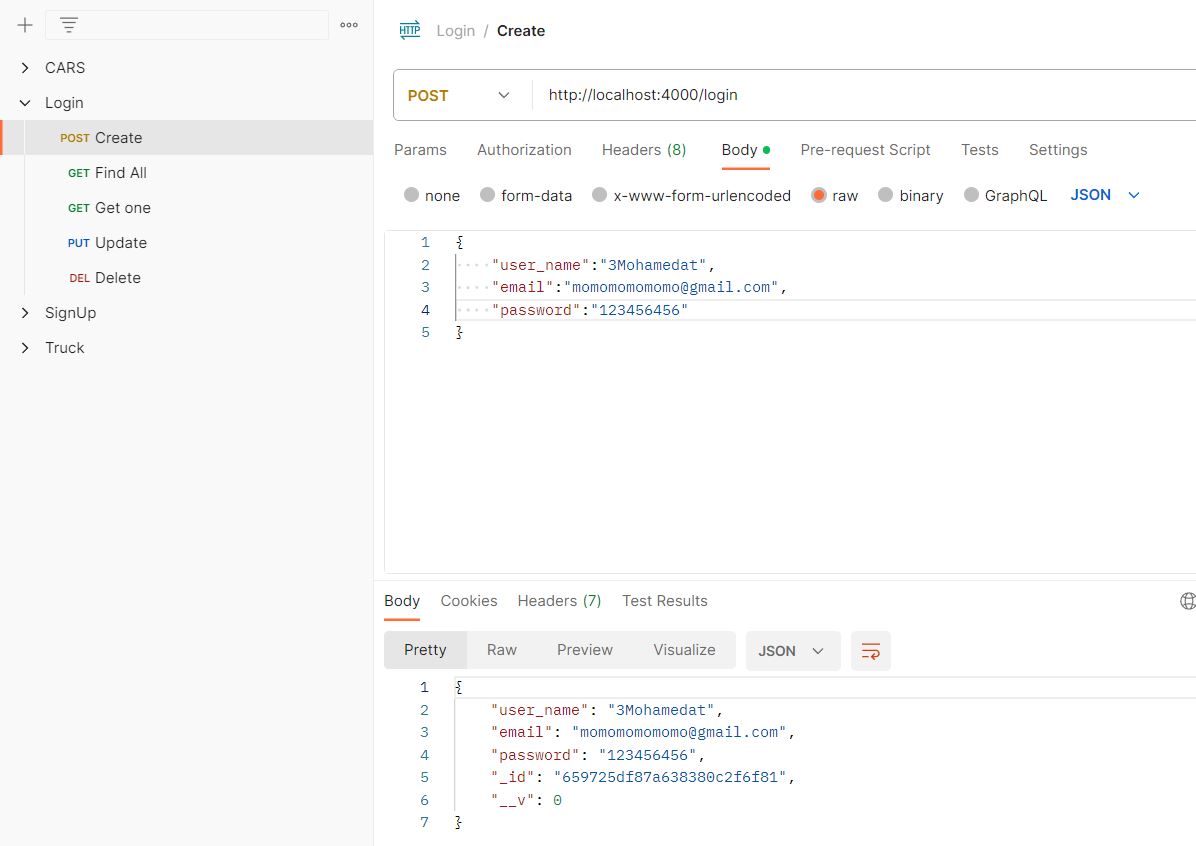
<https://schema.getpostman.com/json/collection/v2.1.0/collection.json>



Here the Login Schema it Contains ‘User name,Email,password’ All of these it's a type of string that can be taken from the user and we got it in the database by POST ,GET , DELETE and PUT.

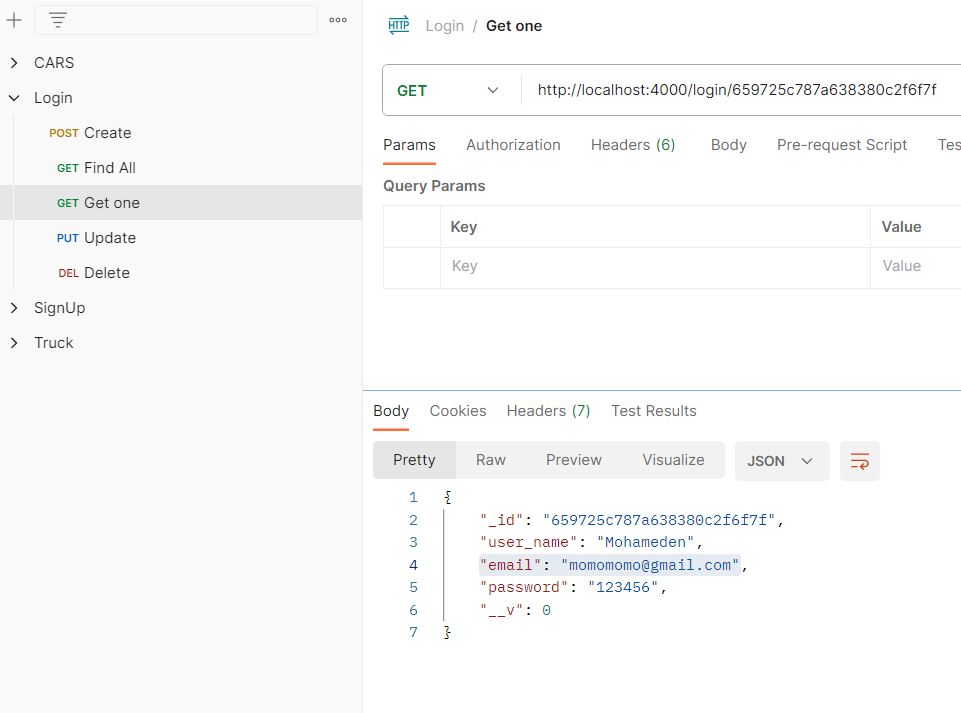
(ii)Login POSTMAN test.

POST



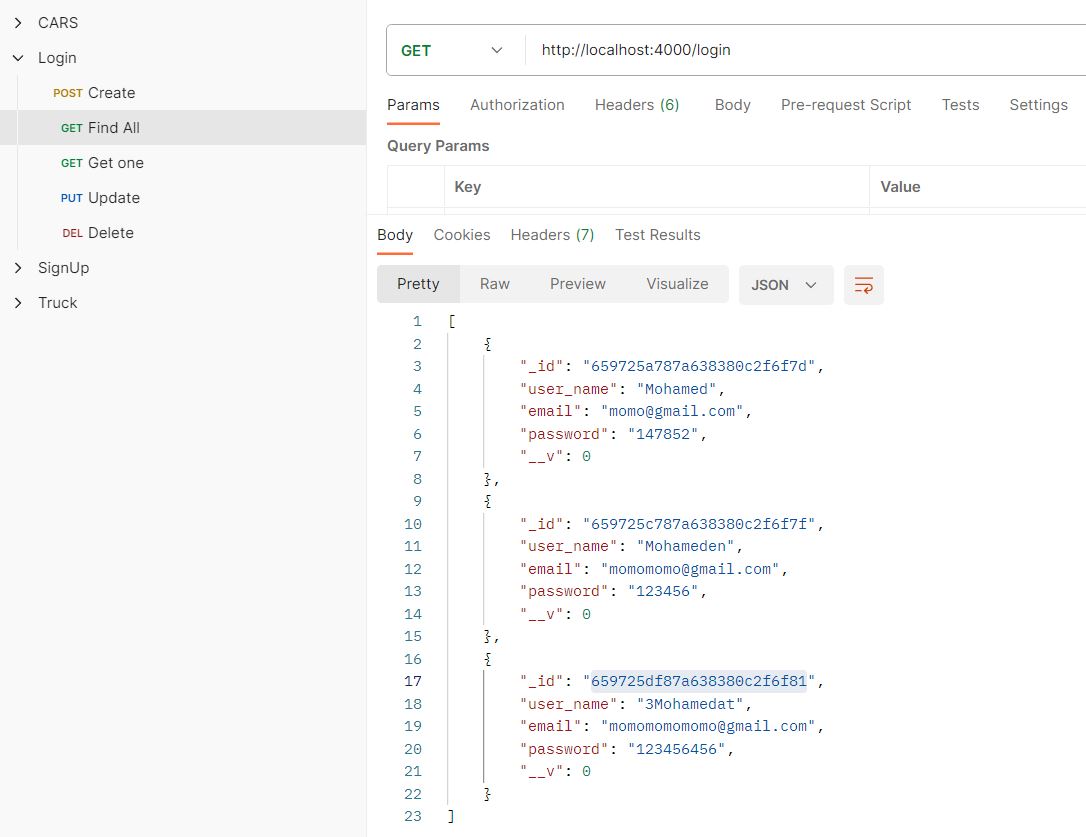
<http://localhost:4000/login>

GET ONE by id



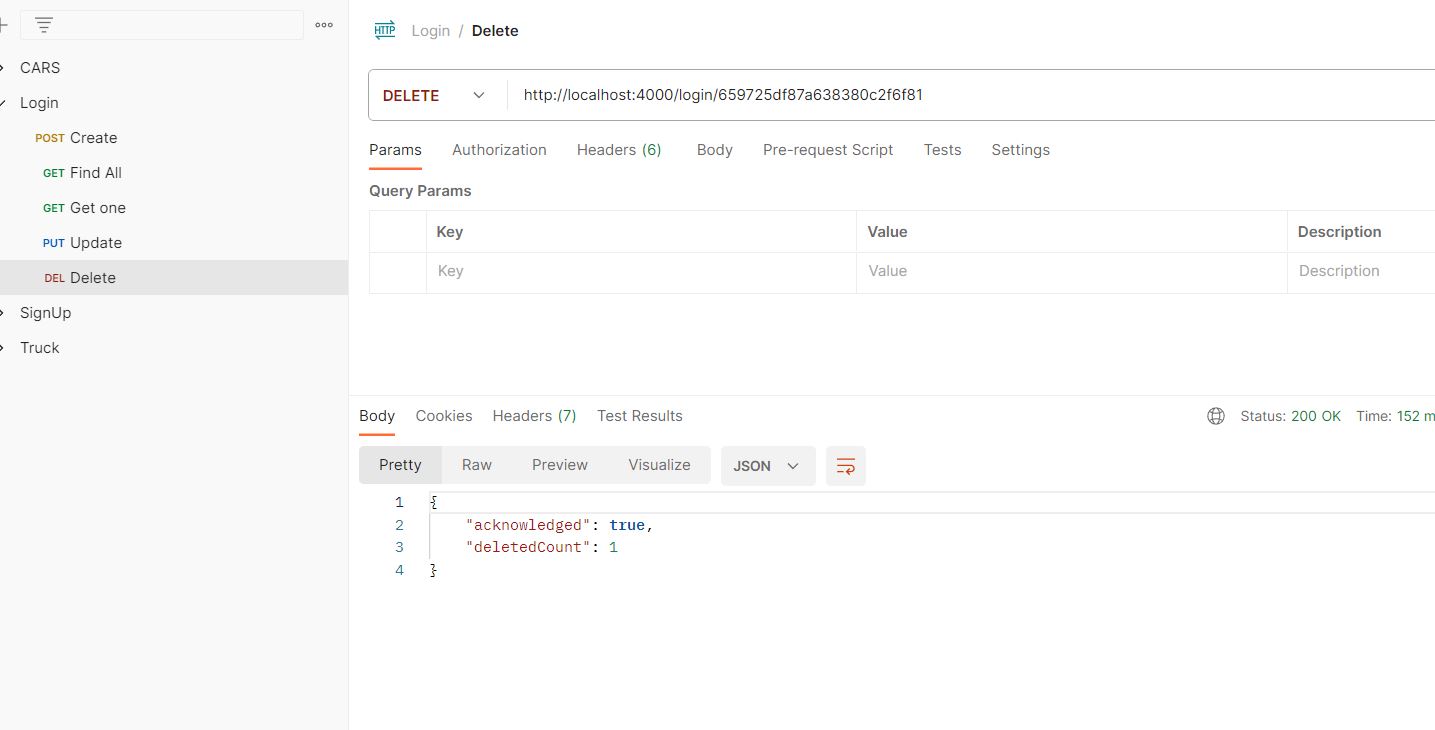
<http://localhost:4000/login/659725c787a638380c2f6f7f>

GET ALL



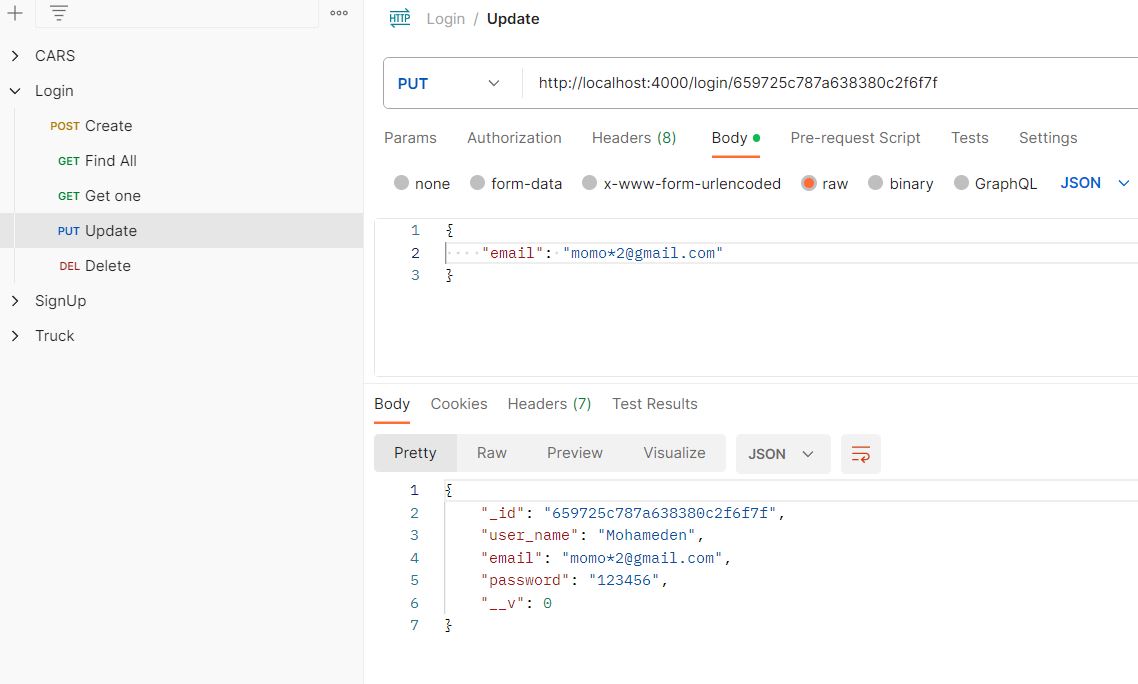
<http://localhost:4000/login>

DELETE by id



<http://localhost:4000/login/659725df87a638380c2f6f81>

UPDATE(PUT)

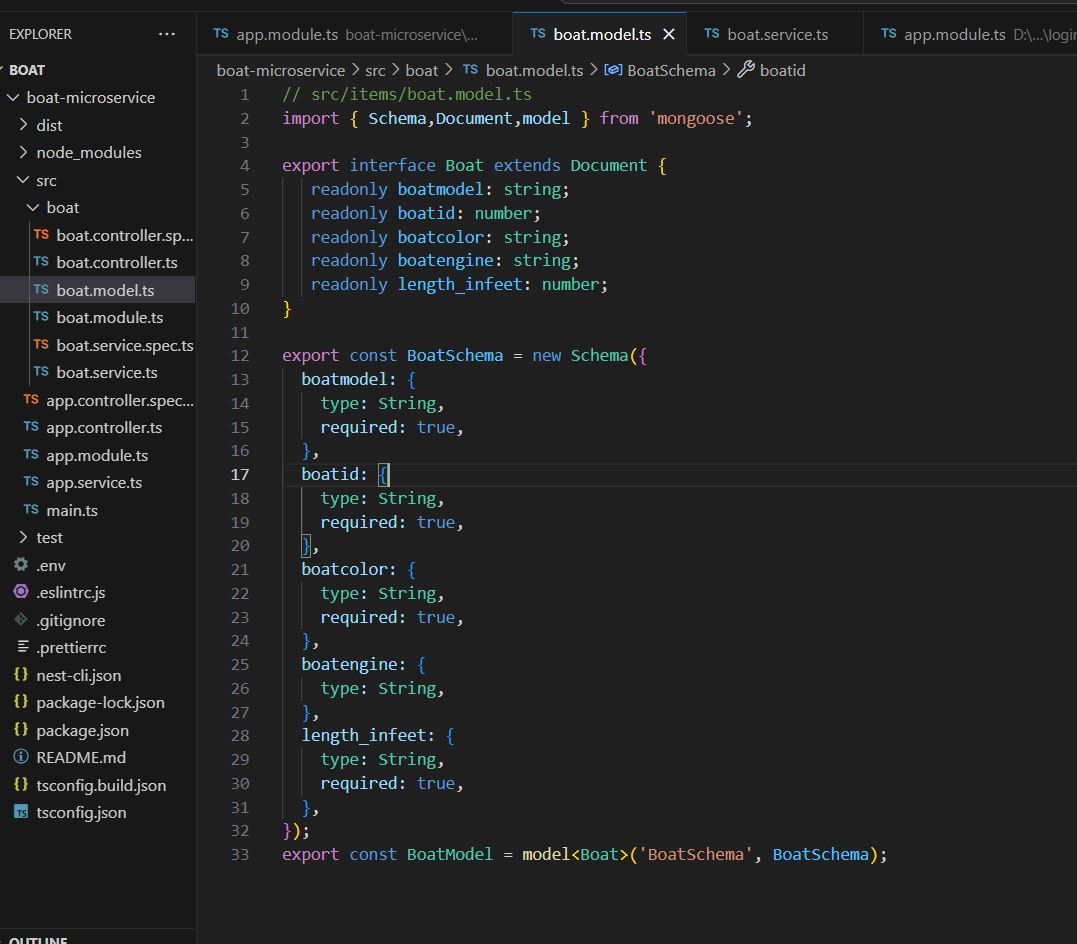


<http://localhost:4000/login/659725c787a638380c2f6f7f>

**(3)**Boat Microservice

(i)Boat Microservice Schema

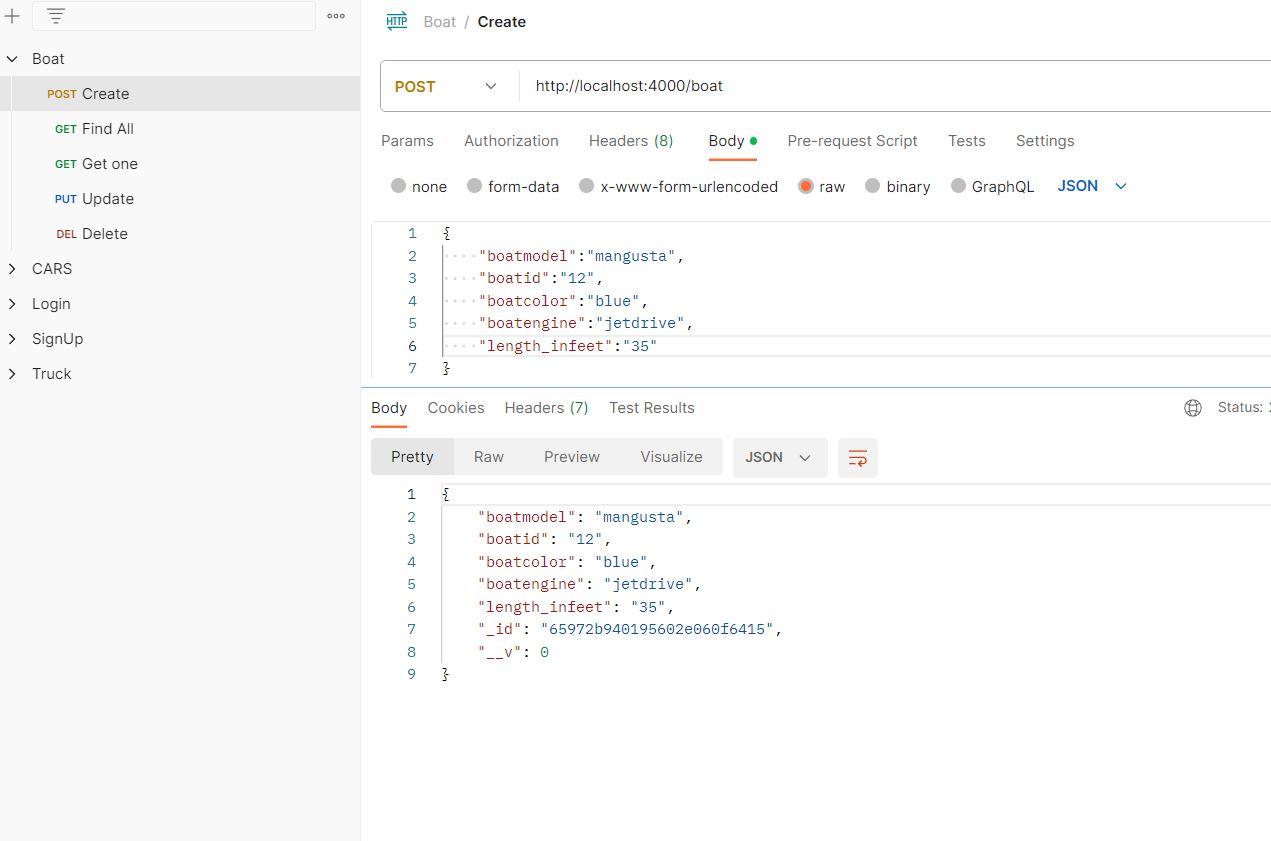
<https://schema.getpostman.com/json/collection/v2.1.0/collection.json>



Here the Boat Schema it Contains ‘Boat Model,Boat Id,Boat Color,Boat Engine and length in feet’ All of these it's a type of string and we got it in the database by POST ,GET , DELETE and PUT if the user want to search or add or remove or edit the product.

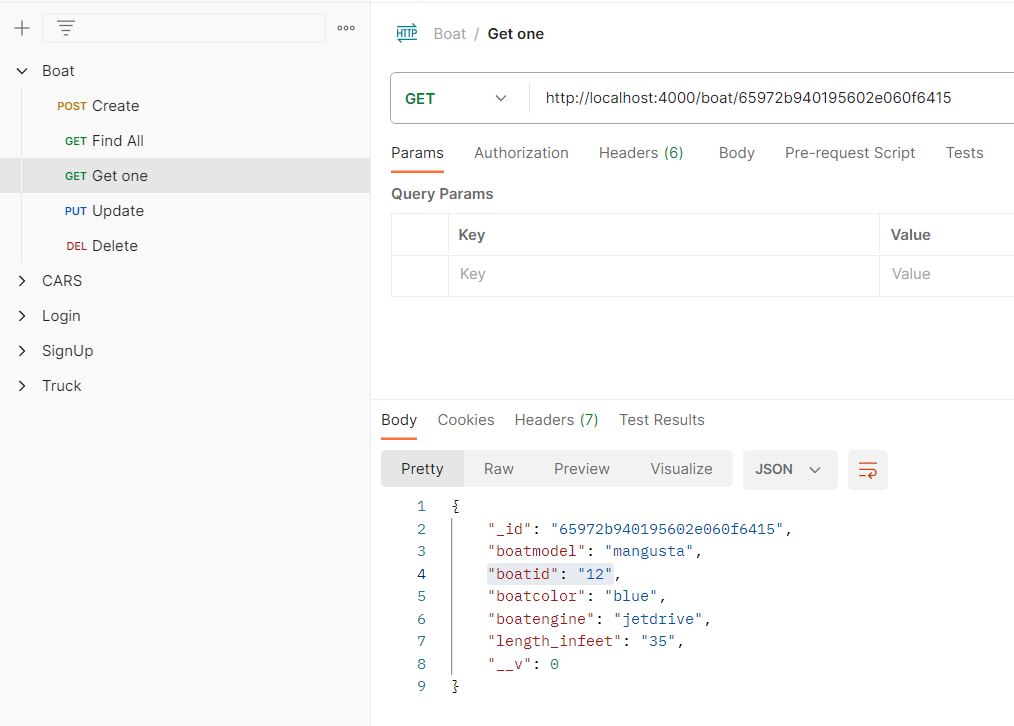
(ii)Boat POSTMAN test.

POST



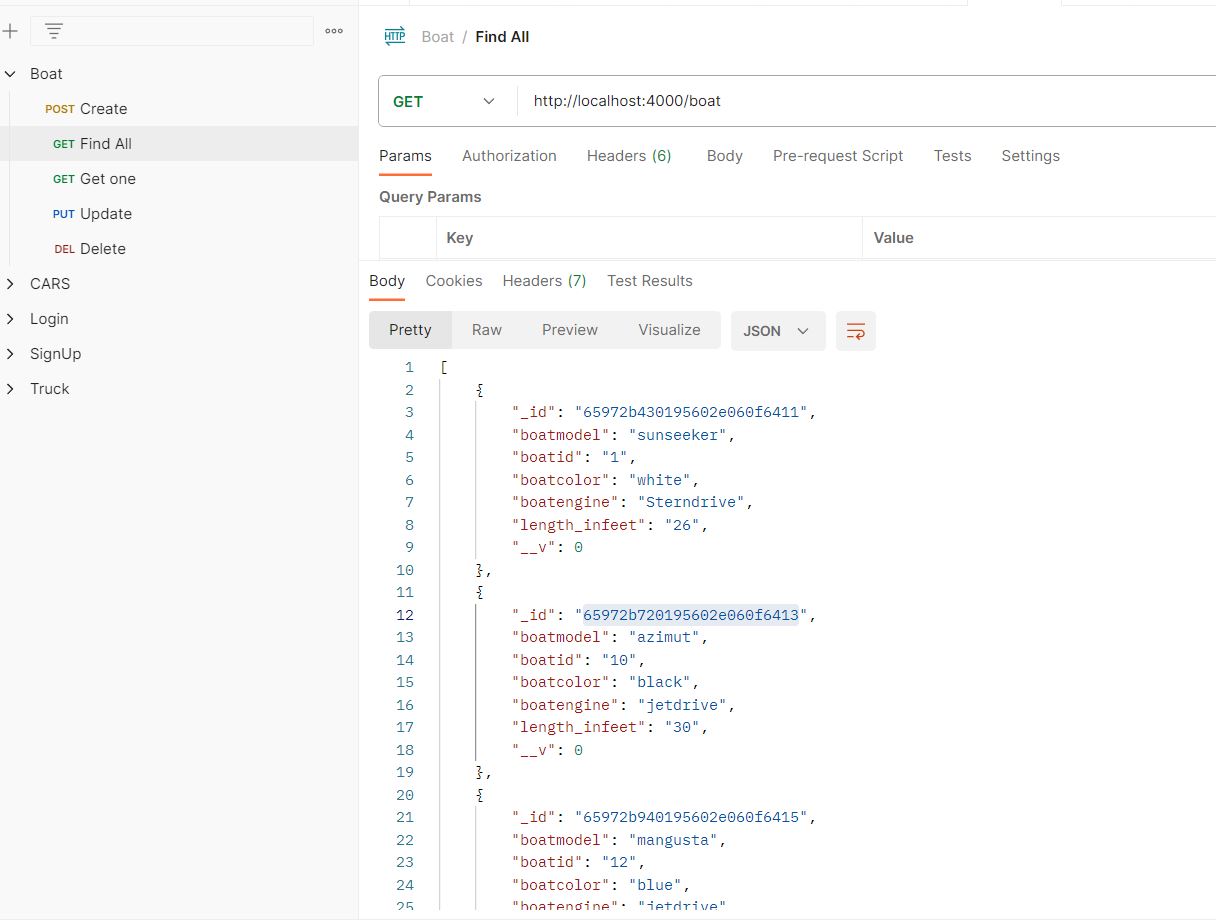
<http://localhost:4000/boat>

GET ONE by id



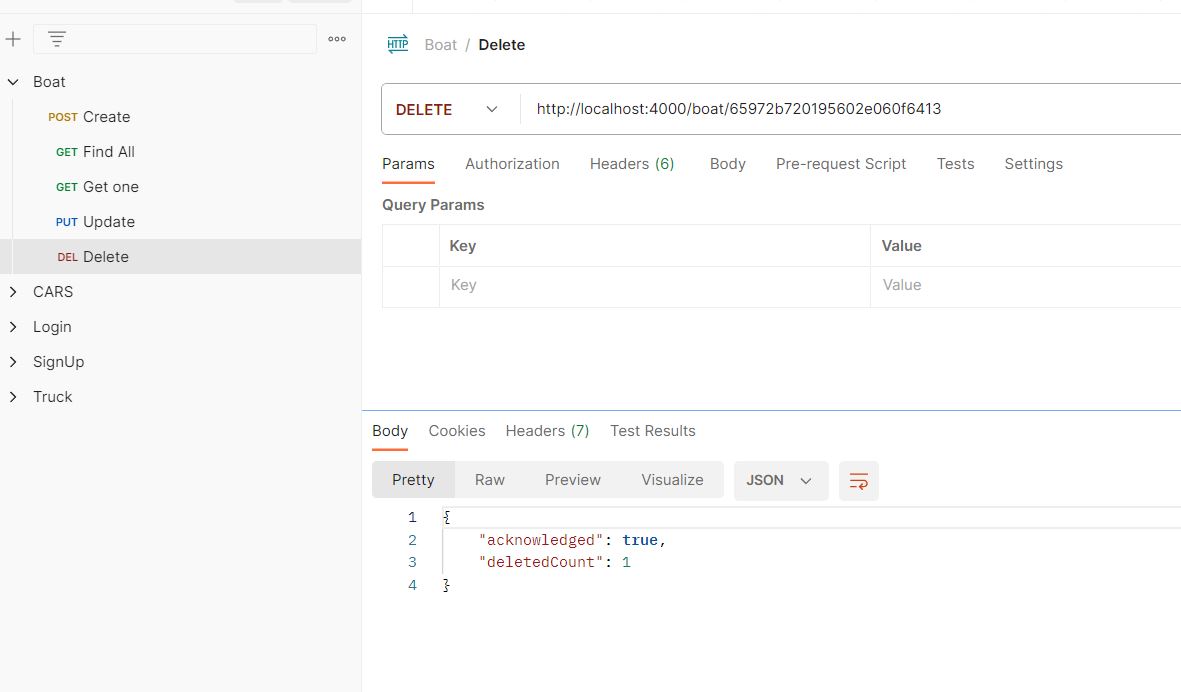
<http://localhost:4000/boat/65972b940195602e060f6415>

GET ALL



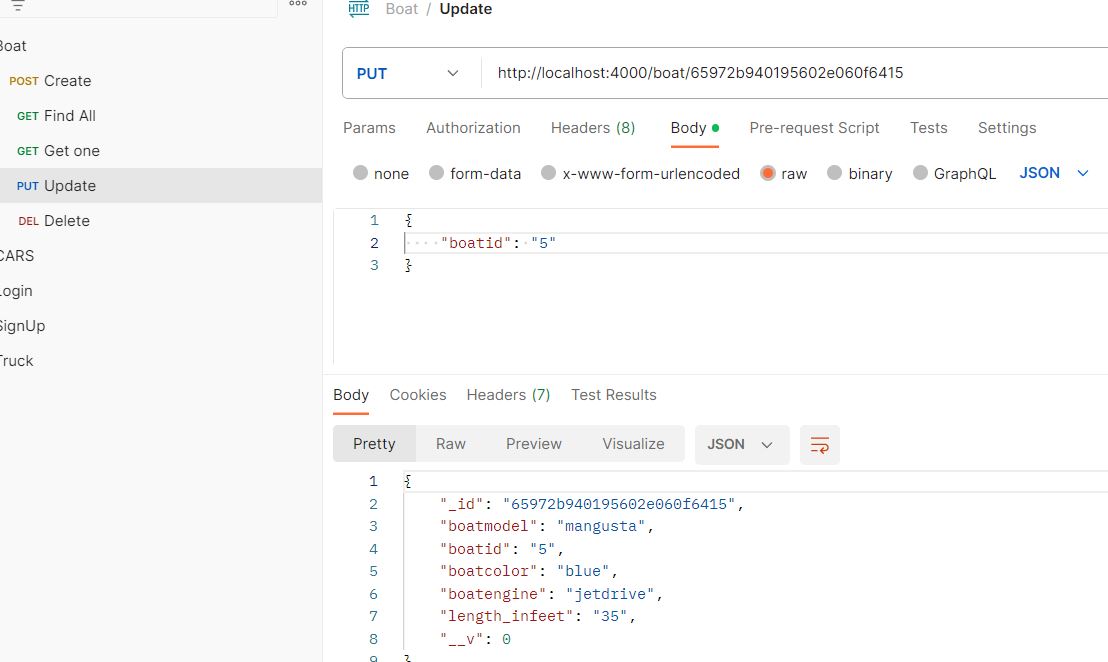
<http://localhost:4000/boat>

DELETE by id



<http://localhost:4000/boat/65972b720195602e060f6413>

UPDATE(PUT)

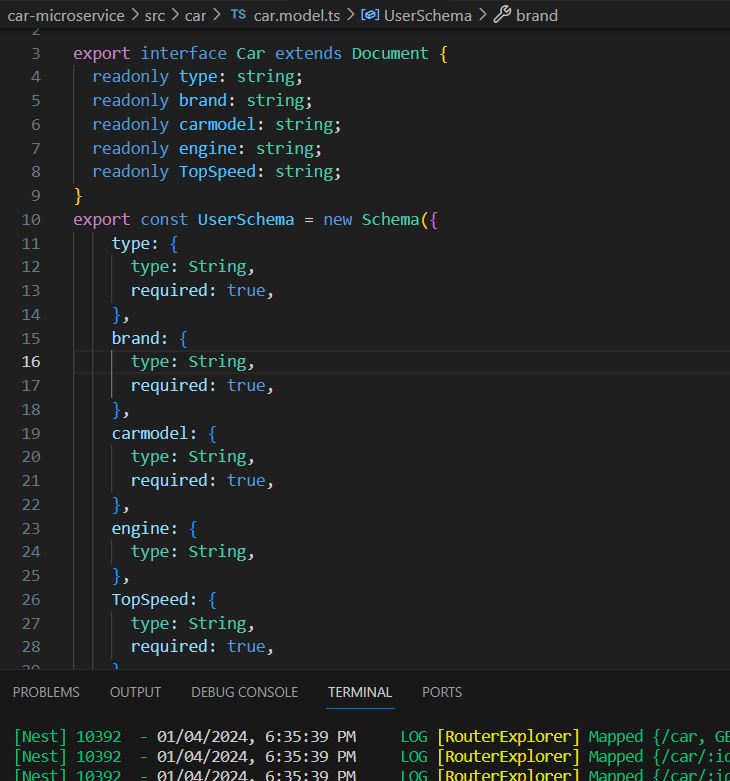


<http://localhost:4000/boat/65972b940195602eD60f6415>

**(4)**Car Microservice

(i)Car Microservice Schema

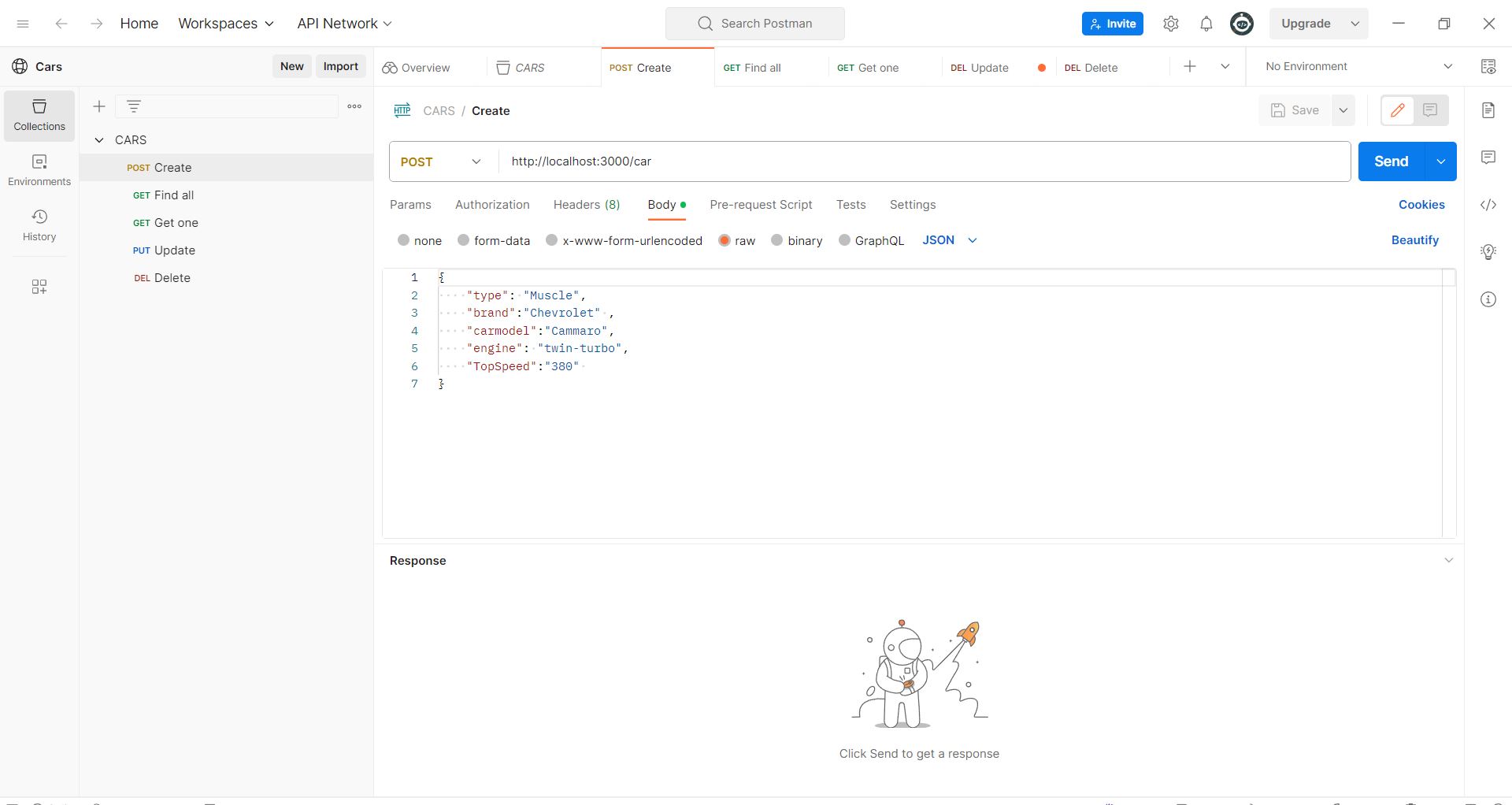
<https://schema.getpostman.com/json/collection/v2.1.0/collection.json>



Here the Car Schema it Contains ‘Brand,Carmodel,Car Engine and Top Speed’ All of these it's a type of string and we got it in the database by POST ,GET , DELETE and PUT if the user want to search or add or remove or edit the product.

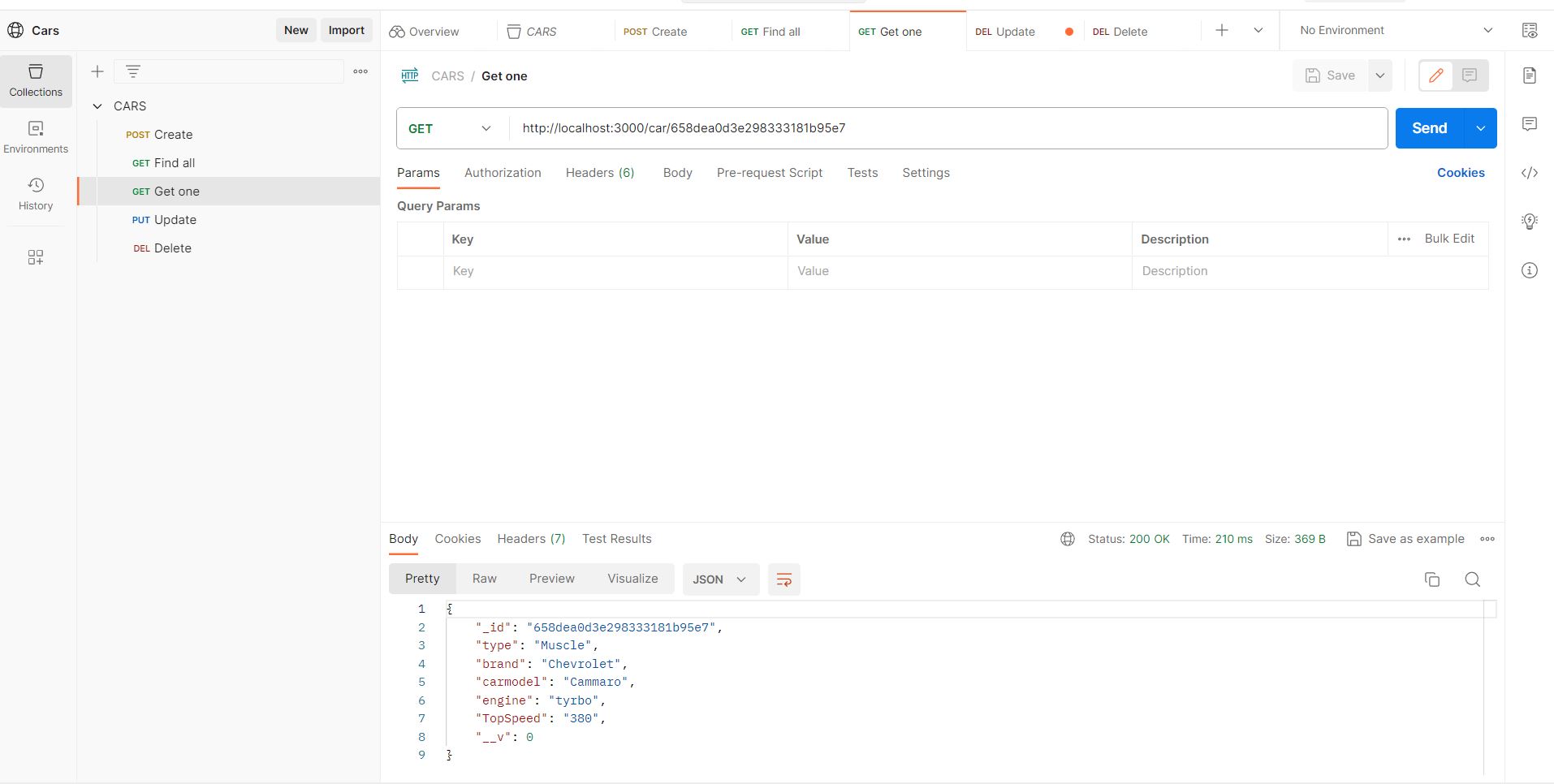
(ii)Car POSTMAN test.

POST



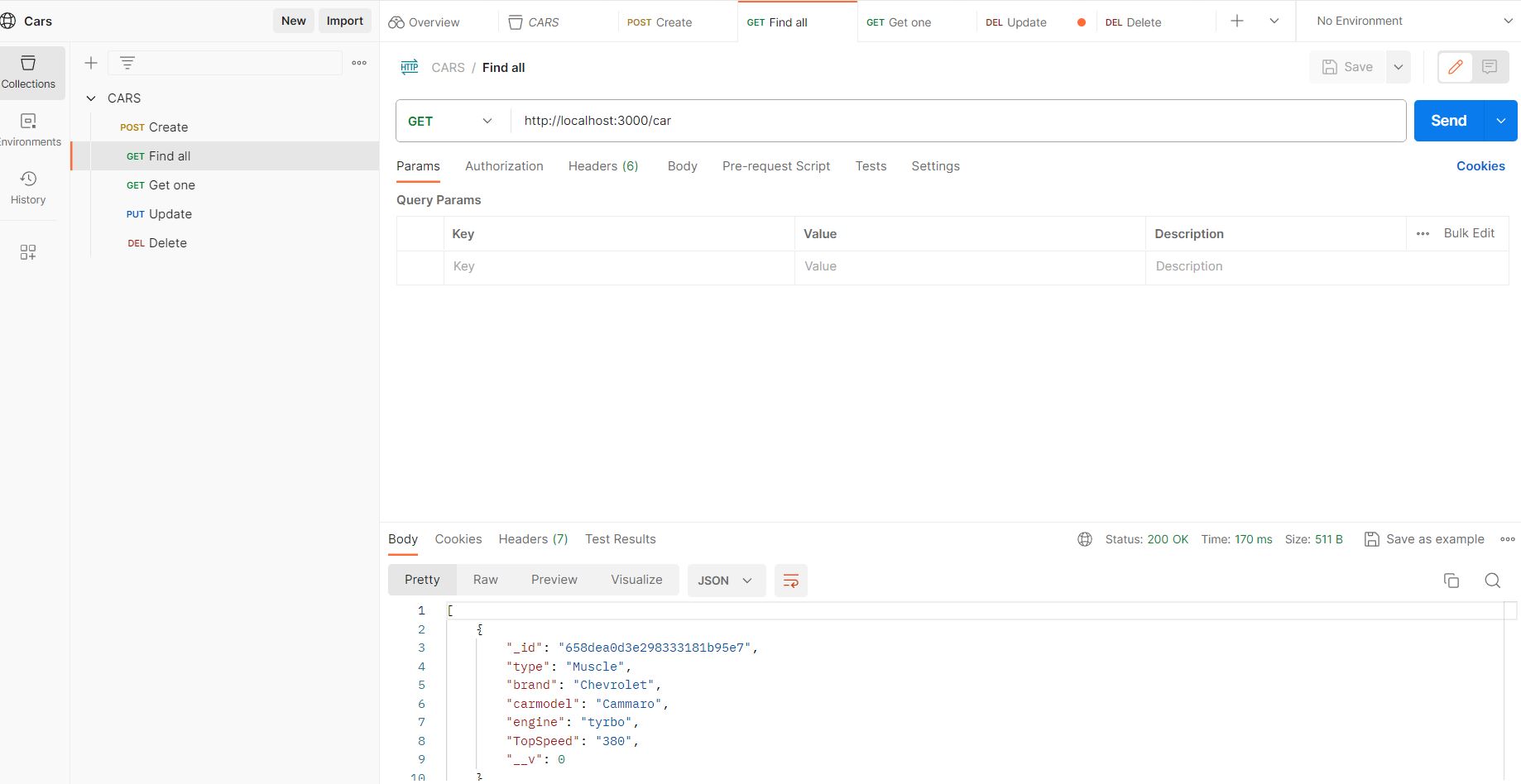
<http://localhost:3000/car>

GET ONE by id



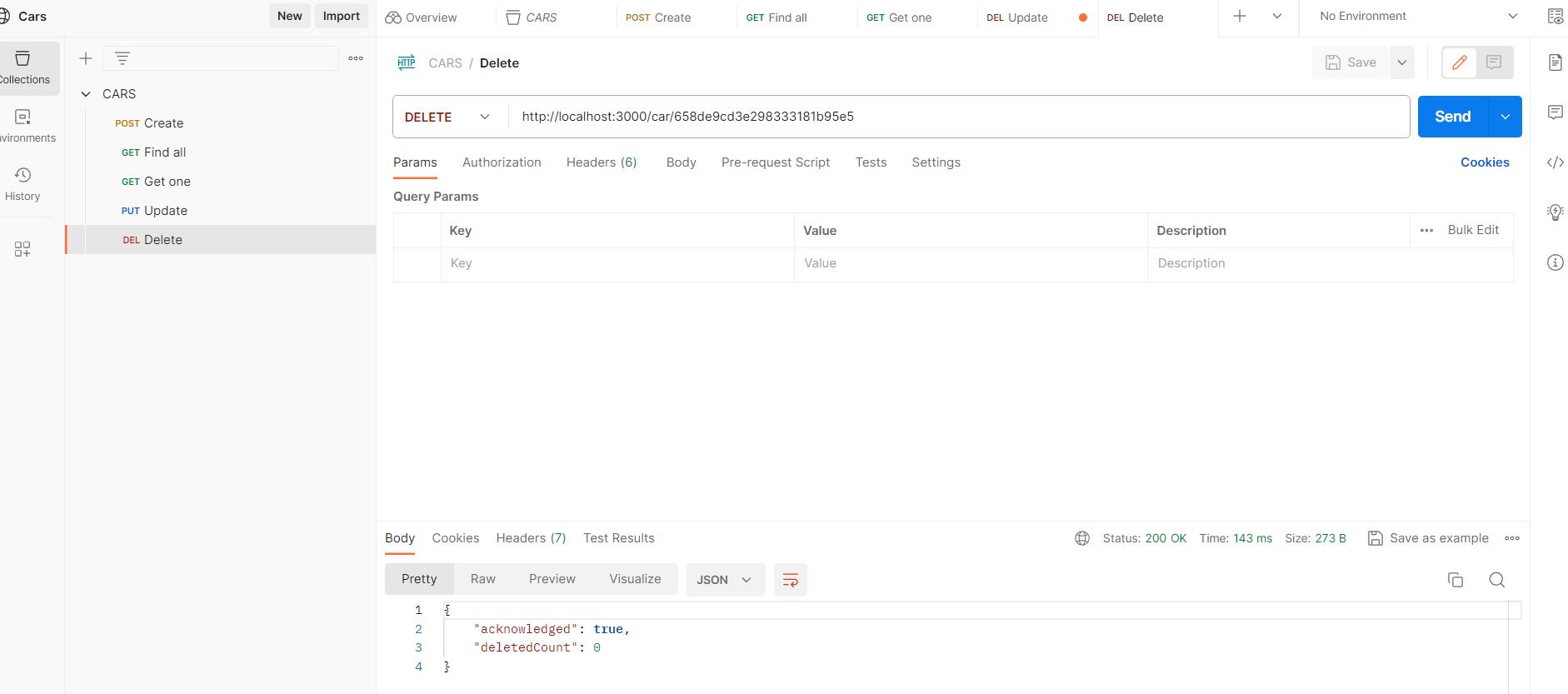
<http://localhost:3000/car/658dea0d3e298333181b95e7>

GET ALL



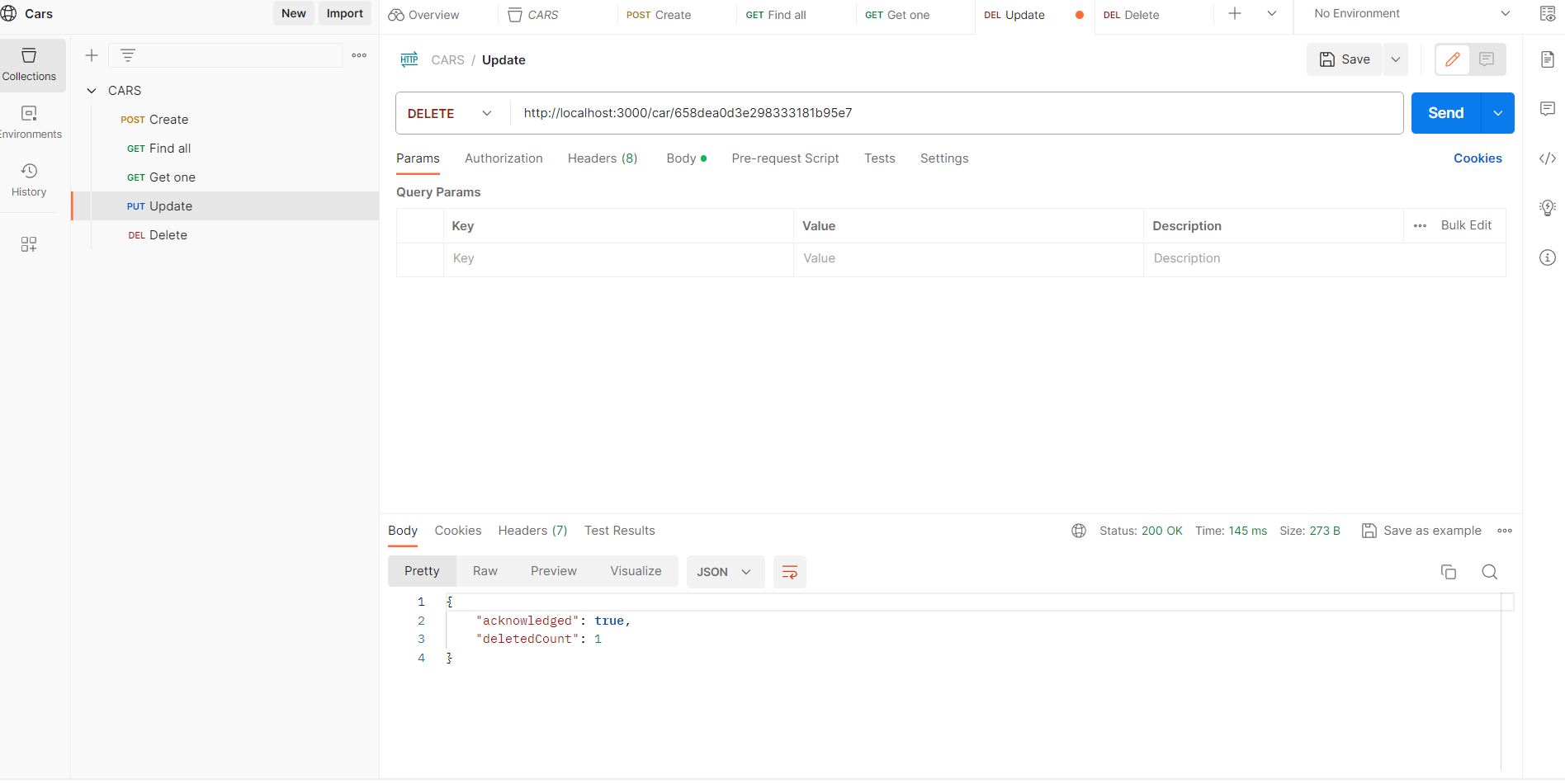
<http://localhost:3000/car>

DELETE by id



<http://localhost:3000/car/658de9cd3e298333181b95e5>

UPDATE(PUT)

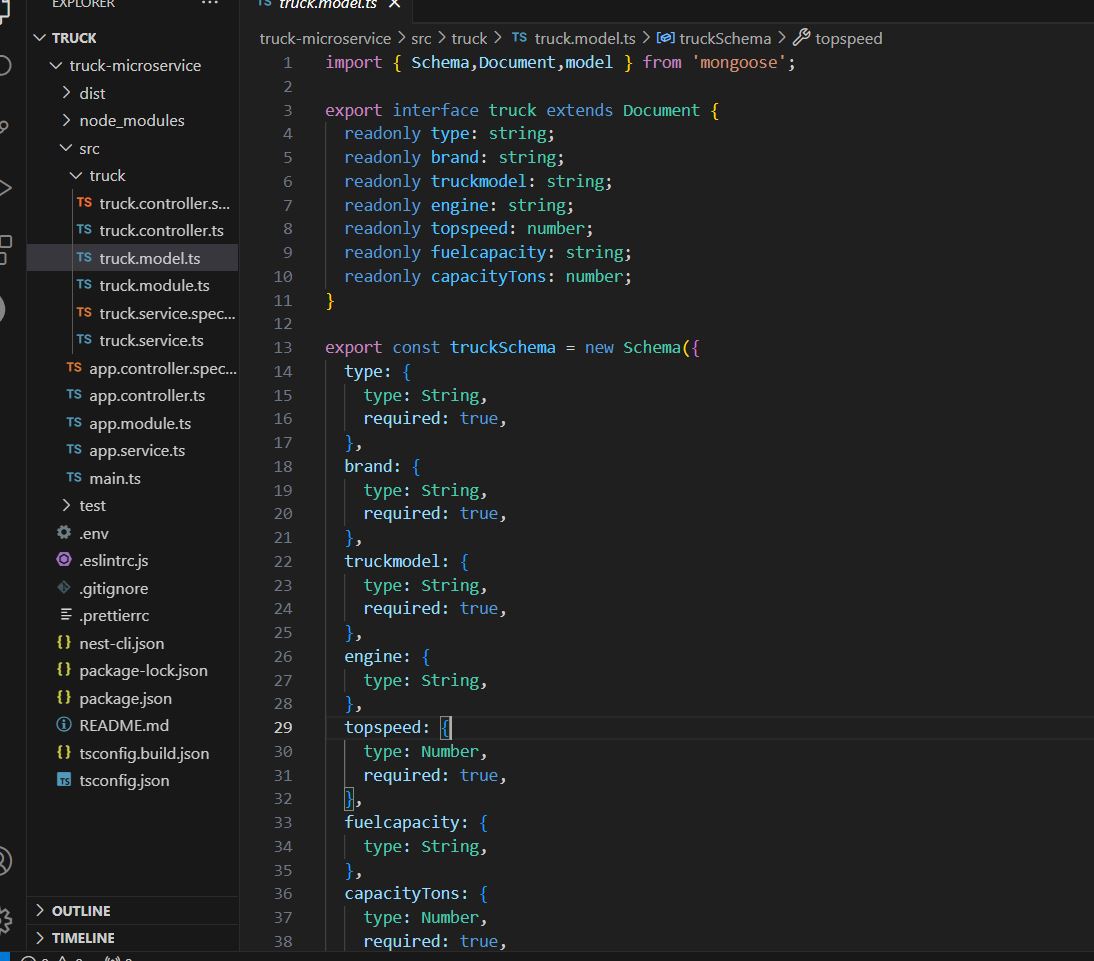


<http://localhost:3000/car/658dea0d3e298333181b95e7>

**(5)**Truck Microservice

(i)Truck Microservice Schema

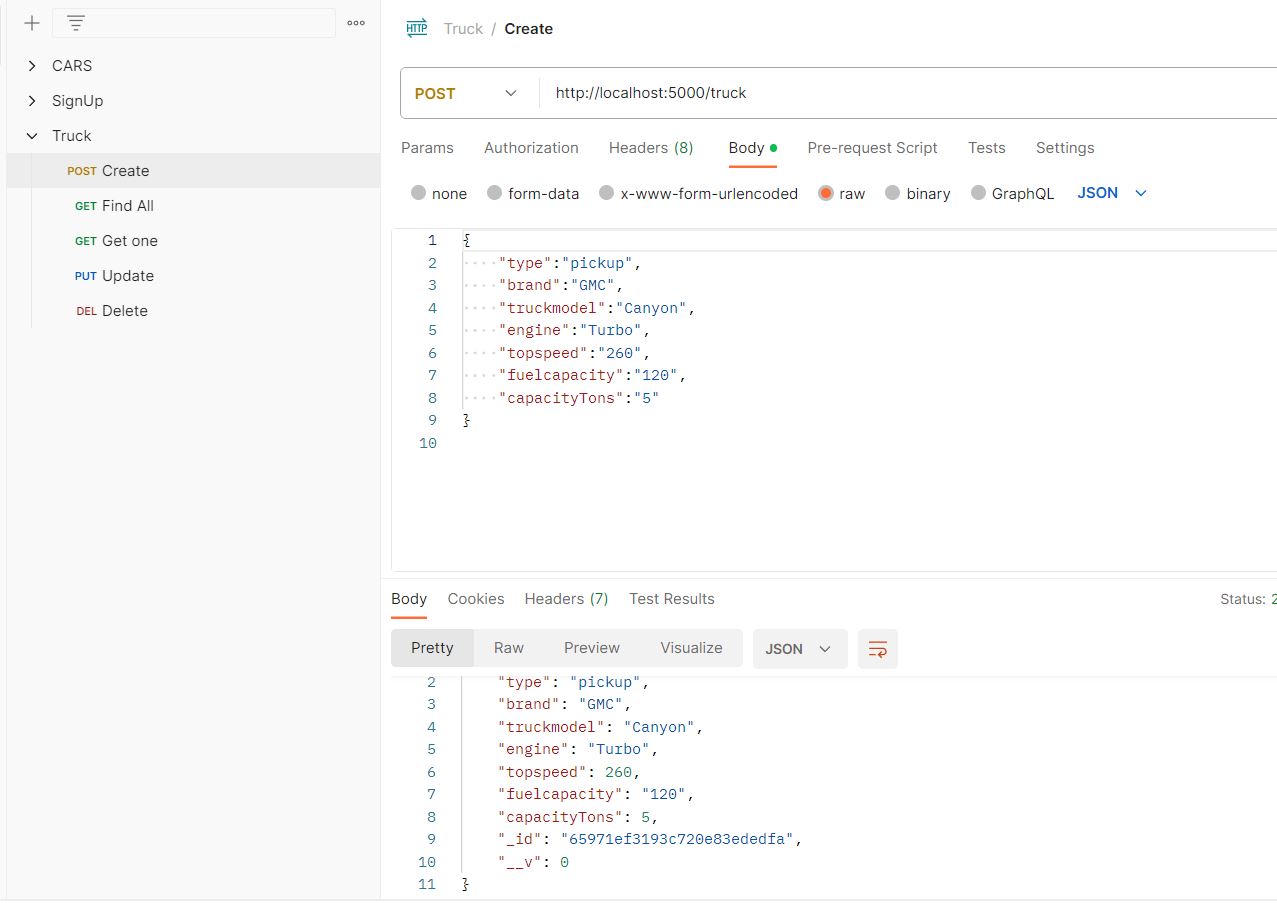
<https://schema.getpostman.com/json/collection/v2.1.0/collection.json>



Here the Truck Schema it Contains ‘Brand,Truckmodel,Truck Engine,Top Speed,Fuel capacity and Capacity tons’ All of these it's a type of string and number of integers and we got it in the database by POST ,GET , DELETE and PUT if the user want to search or add or remove or edit the product.

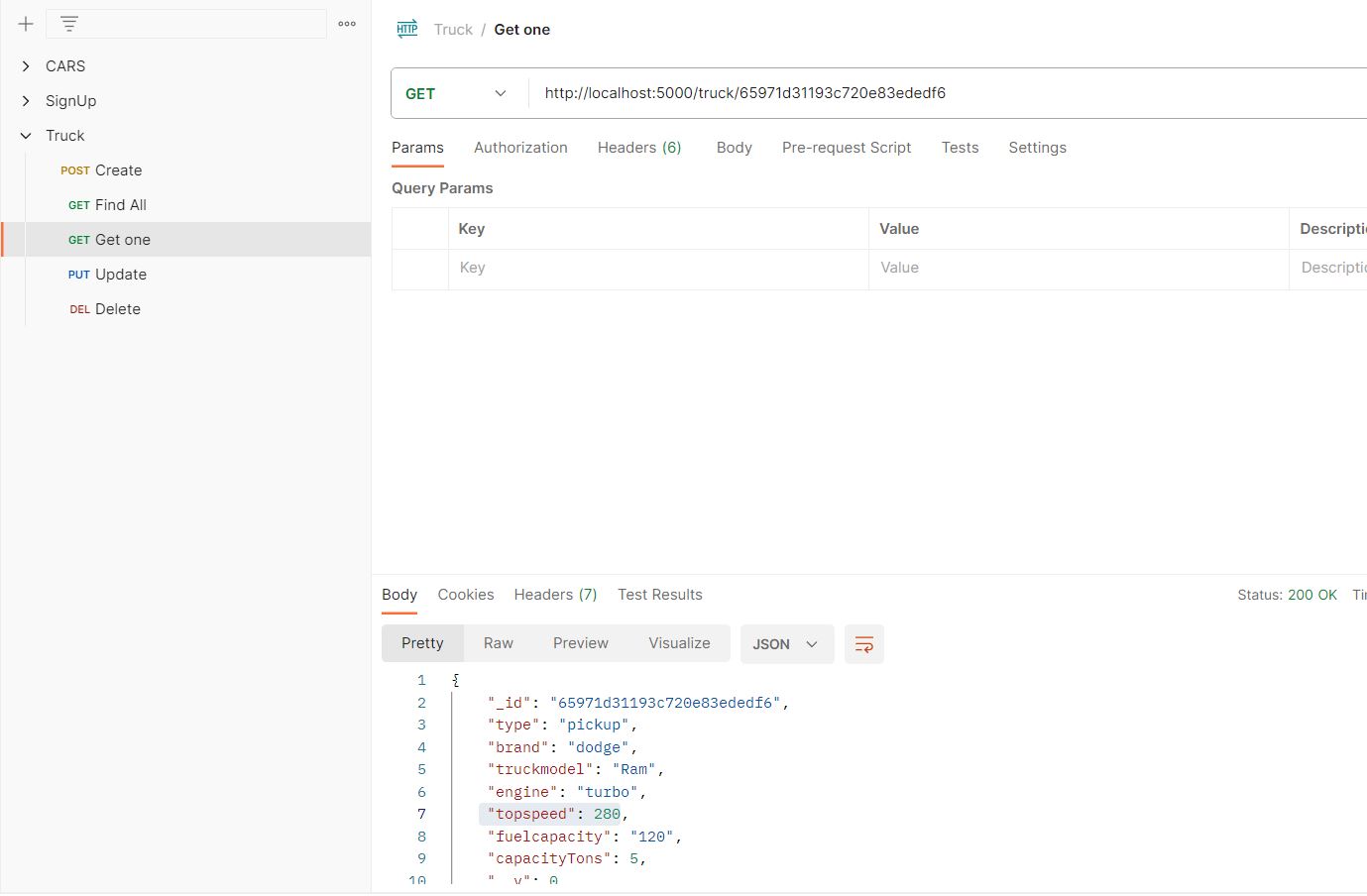
(ii)Truck POSTMAN test.

POST



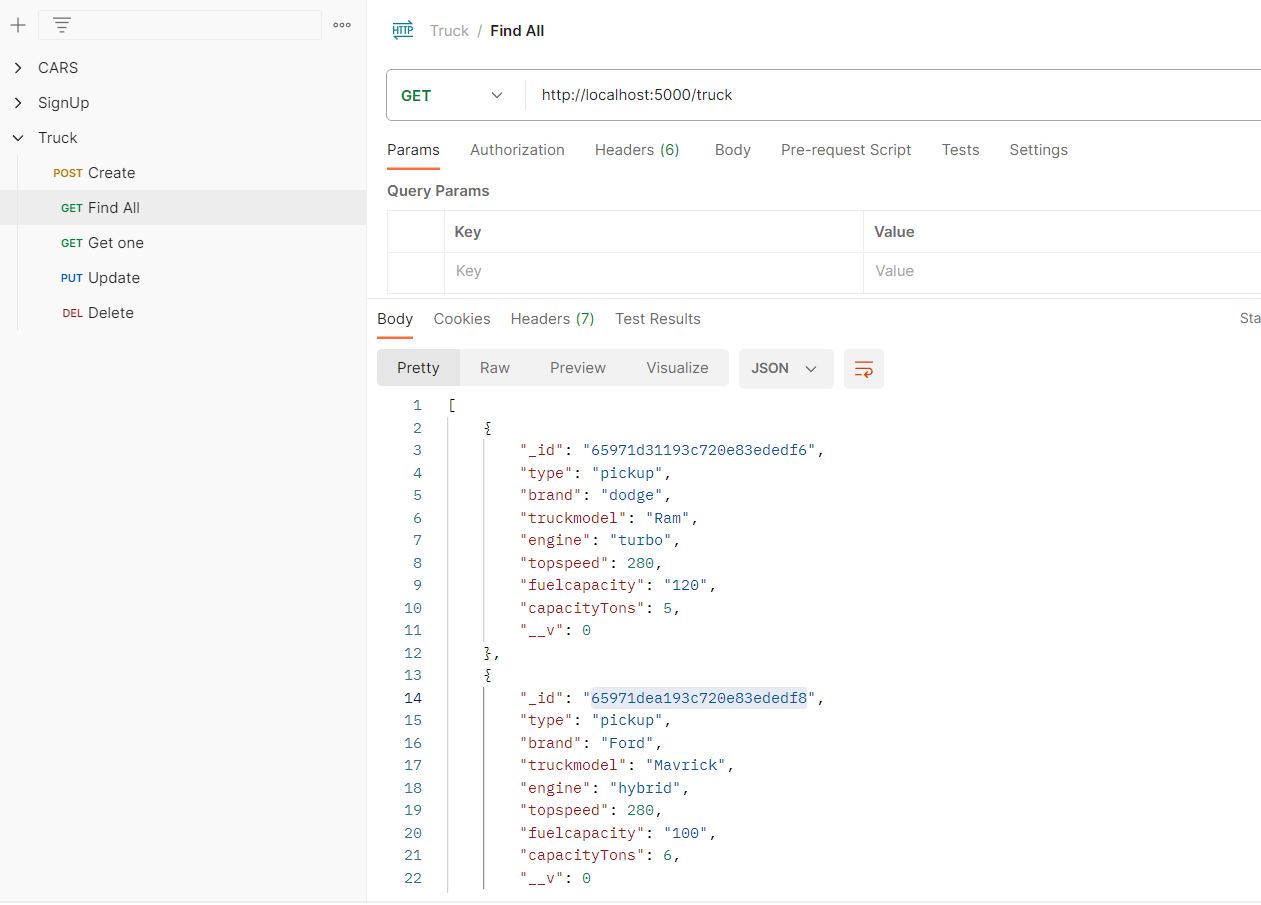
<http://localhost:5000/truck>

GET ONE by id



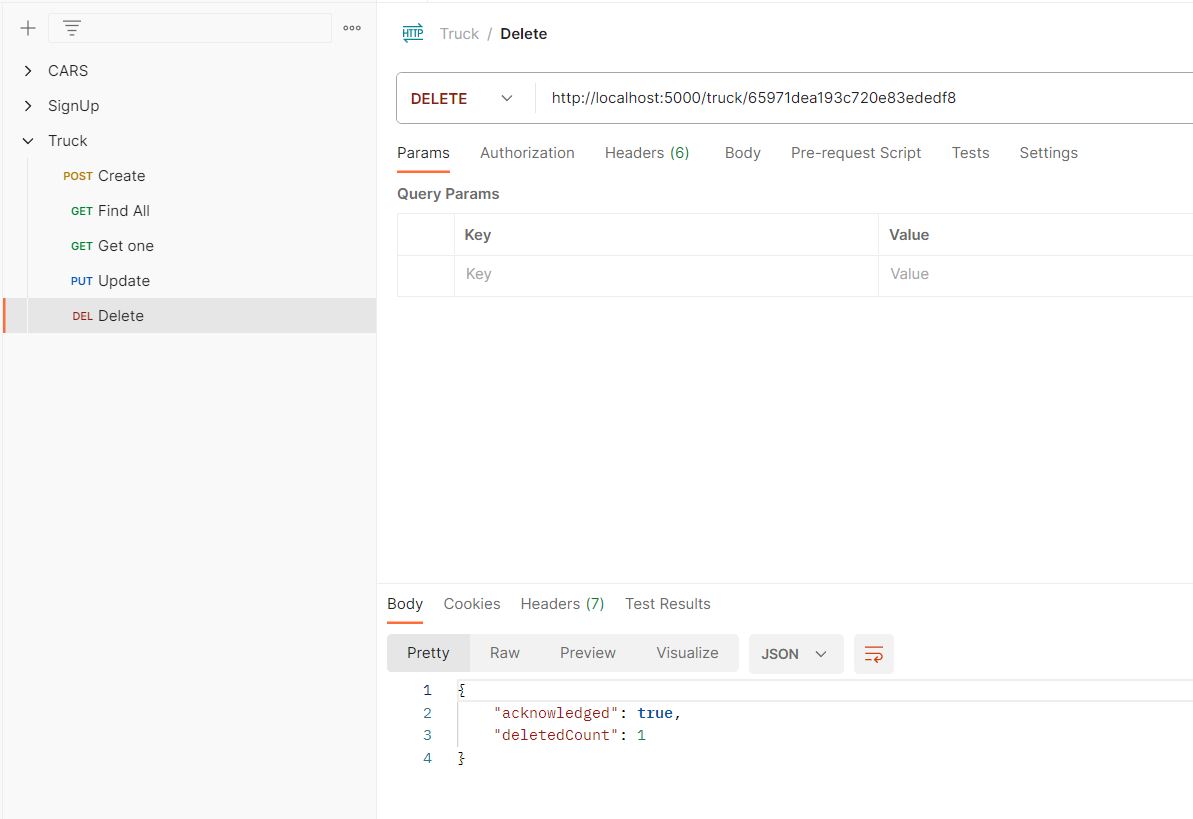
<http://localhost:5000/truck/65971d31193c720e83ededf6>

GET ALL



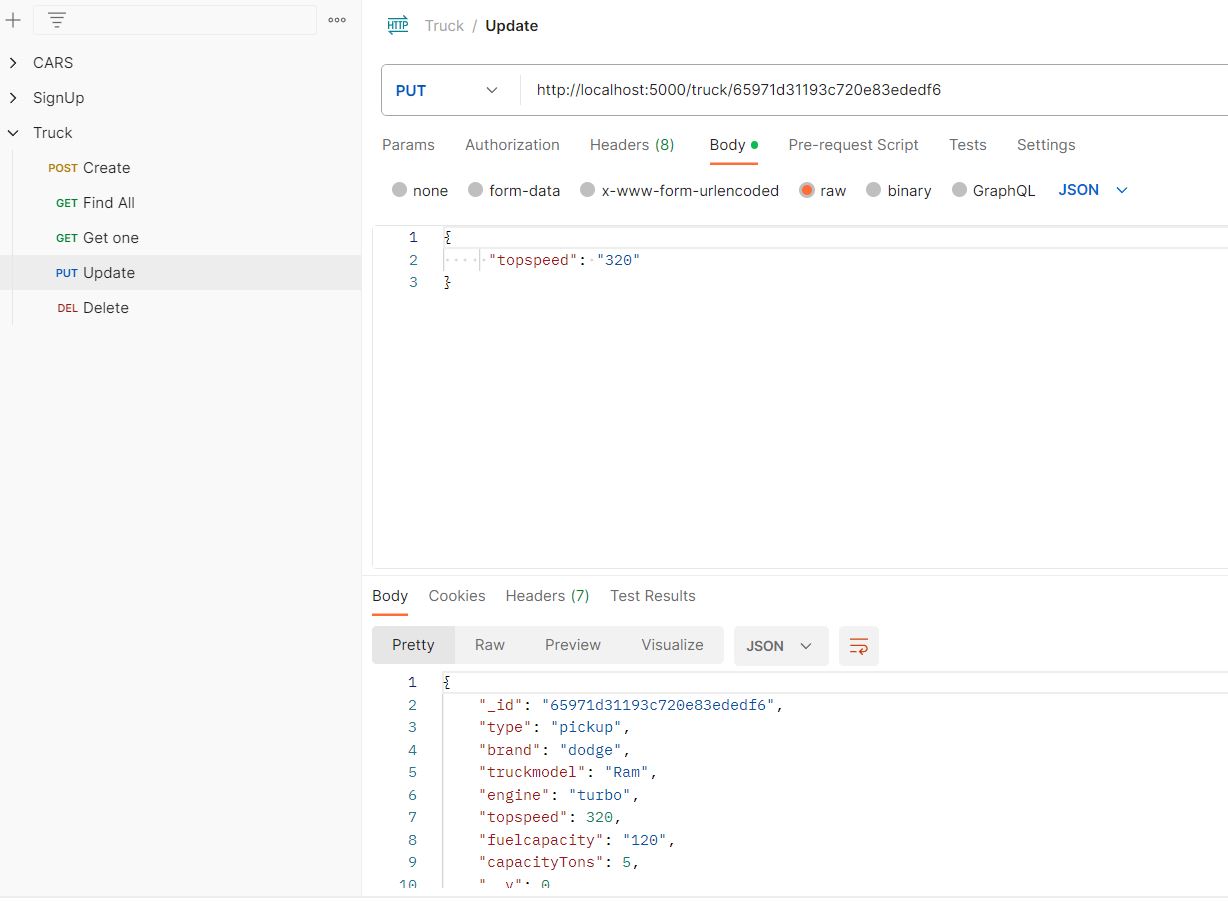
<http://localhost:5000/truck>

DELETE by id



<http://localhost:5000/truck/65971dea193c720e83ededf8>

UPDATE(PUT)

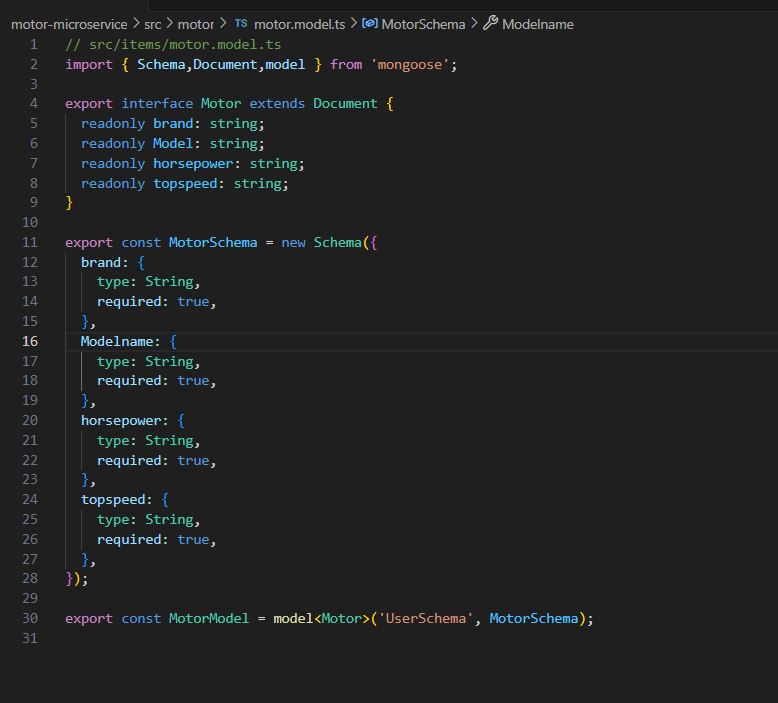


<http://localhost:5000/truck/65971d31193c720e83ededf6>

**(6)**Motorcycle Microservice

(i)Motorcycle Microservice Schema.

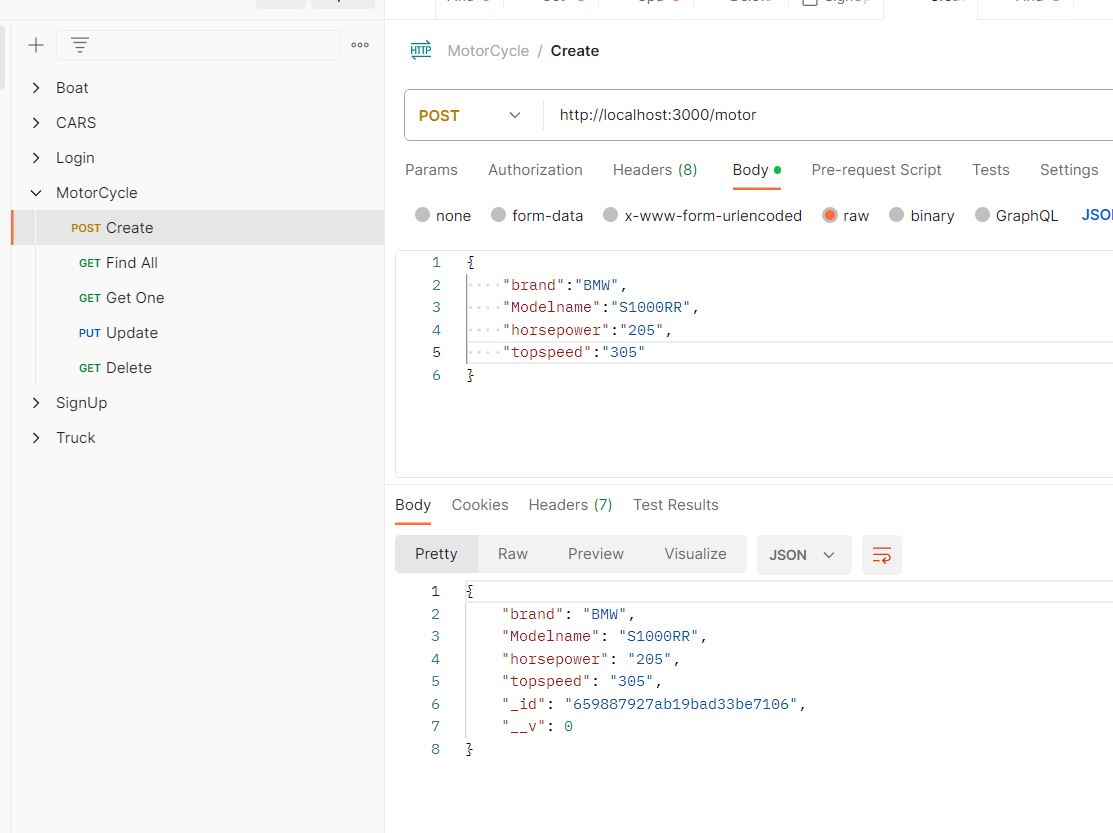
<https://schema.getpostman.com/json/collection/v2.1.0/collection.json>



Here the Motorcycle Schema it Contains ‘Brand,Model name,Horsepower and Top Speed’ All of these it's a type of string and we got it in the database by POST ,GET , DELETE and PUT if the user want to search or add or remove or edit the product.

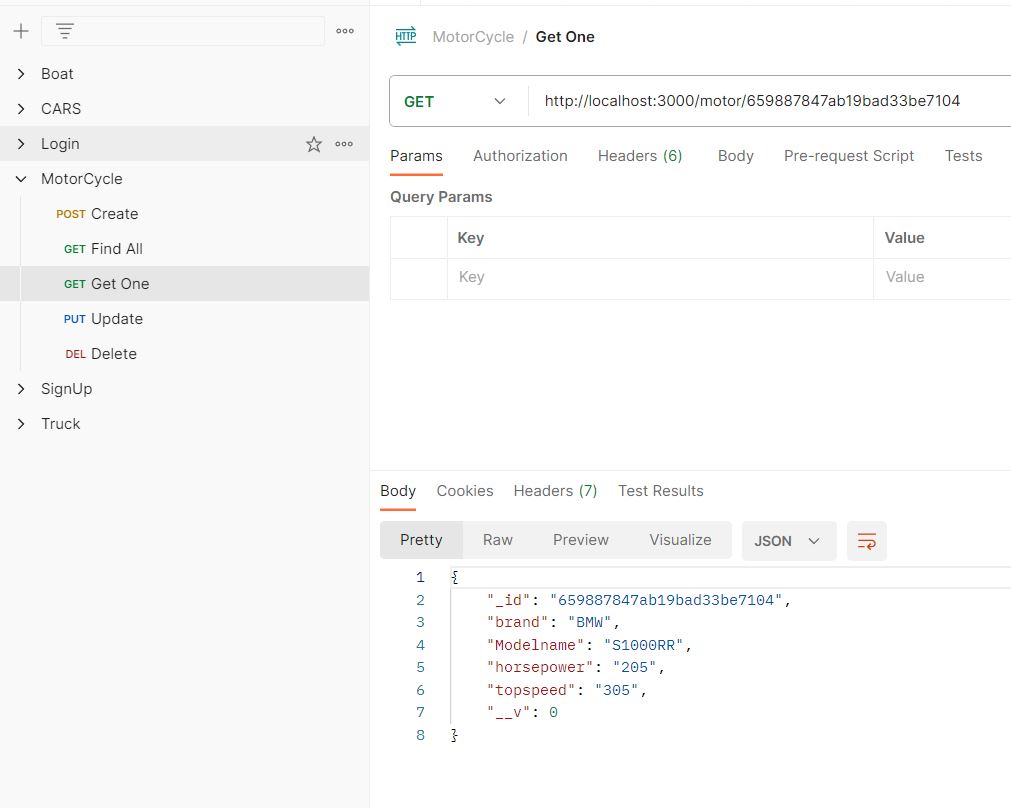
(ii)Motorcycle POSTMAN test.

POST



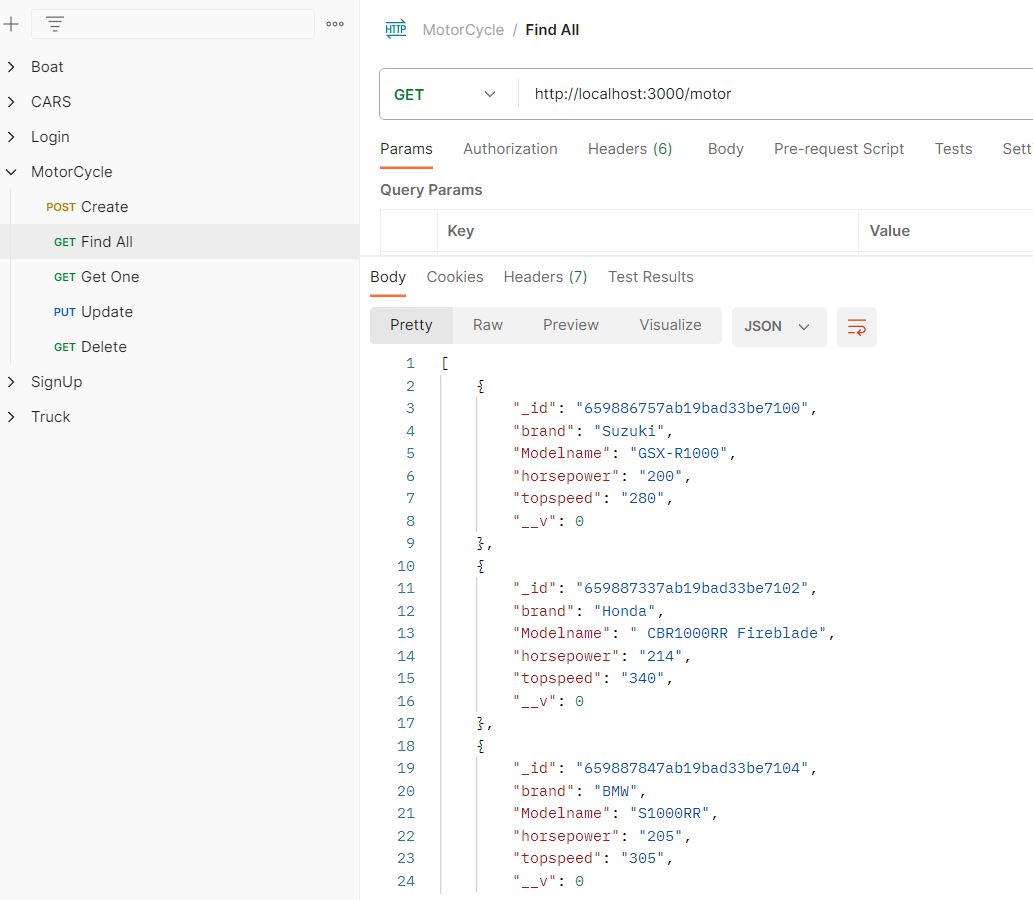
<http://localhost:3000/motor>

GET ONE by id



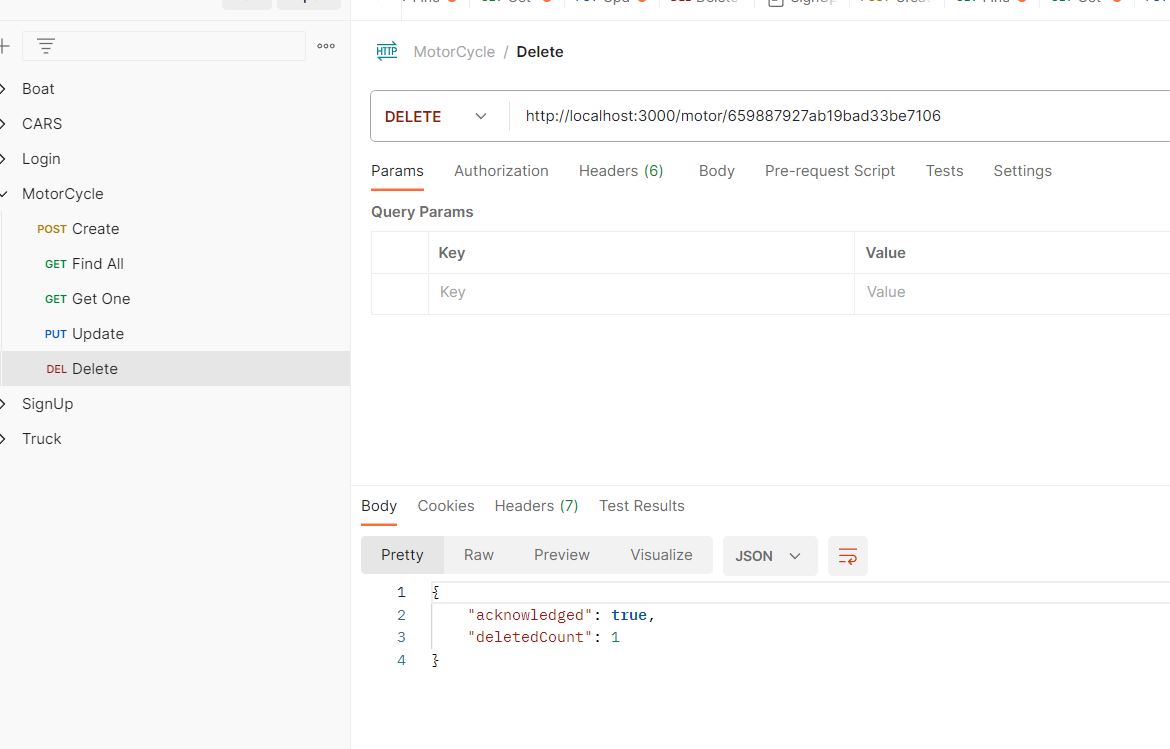
<http://localhost:3000/motor/659887847ab19bad33be7104>

GET ALL



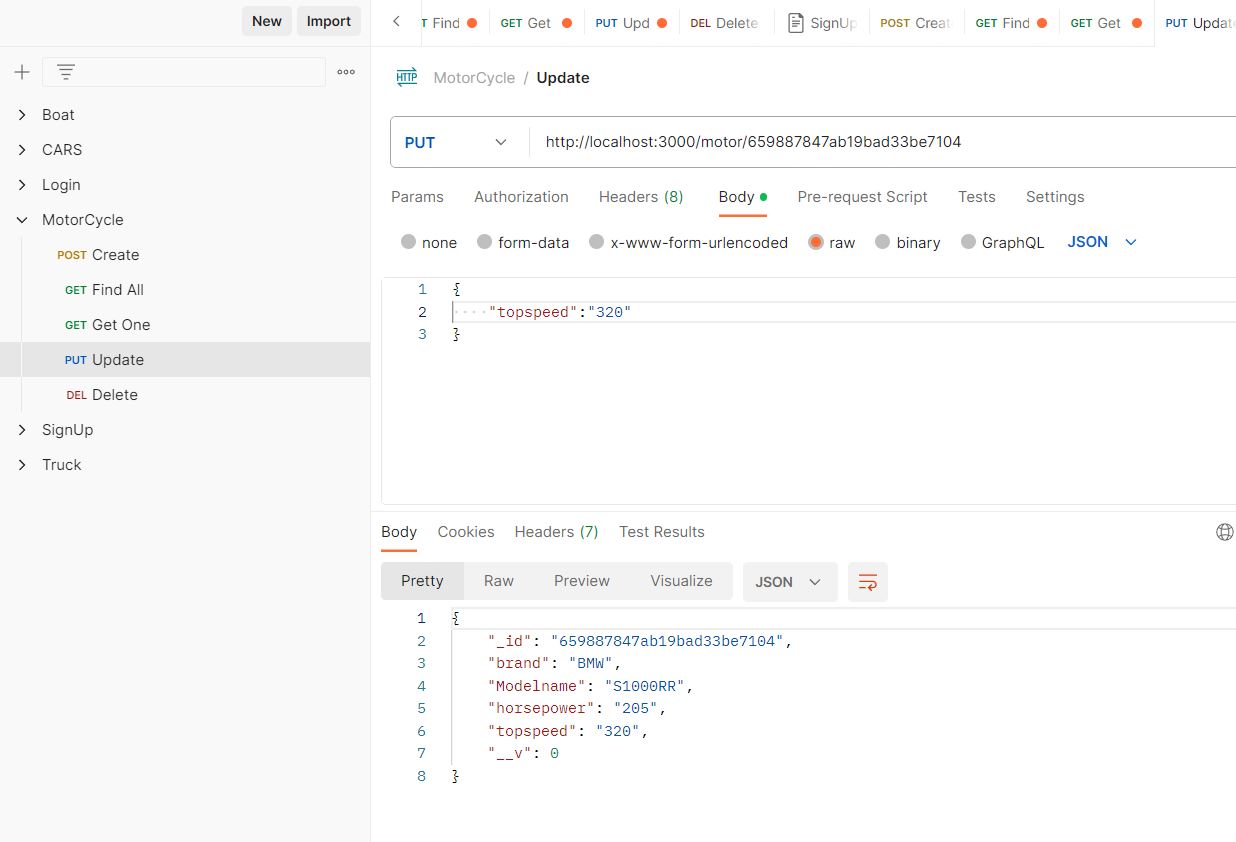
<http://localhost:3000/motor>

DELETE by id



<http://localhost:3000/motor/659887927ab19bad33be7106>

UPDATE(PUT)



<http://localhost:3000/motor/659887847ab19bad33be7104>

**(7)**Conclusion

All of these Microservices are connected to the same DATABASE in MongoDB which make the system more oriented and makes the services easier and more coupled to each other.

**(8)**Github Links

[BassemO1/Service-Project (github.com)](https://github.com/BassemO1/Service-Project)

[boat-microservice · BassemO1/Service-Project@f3c0dfa (github.com)](https://github.com/BassemO1/Service-Project/commit/f3c0dfa4d319381714e06468cc3daefd7b8be582)

[SignUP · BassemO1/Service-Project@99f5e17 (github.com)](https://github.com/BassemO1/Service-Project/commit/99f5e175e2b018c657454822cfac3a06db5b1c68)

[Add files via upload · BassemO1/Service-Project@6e58f30 (github.com)](https://github.com/BassemO1/Service-Project/commit/6e58f30ad21843a25c2094078f4c4d3e6ca3e198)

[Add files via upload · BassemO1/Service-Project@9daf3f3 (github.com)](https://github.com/BassemO1/Service-Project/commit/9daf3f32085c41f3d54ed491255f758f7620c473)

[Add files via upload · BassemO1/Service-Project@b0e86d6 (github.com)](https://github.com/BassemO1/Service-Project/commit/b0e86d6c763618fedb2fa31d2f4180169aaa592c)

**(9)**App & Technologies

Reactjs,Nodejs,Postman,vs code,Nestjs,MongoDB

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |