Hotel Reservation System

Project Member

- ❑ Member
  - ❖Ms Anushree Joshi
  - ❖Ms Shivani Gole
  - ❖Ms Anupriya Kannan
  - ❖Mrs. Kausar Sheikh
  - ❖Ms Sangeetha T R
  - ❖Ms Priyanka Parab
- ❑ Guided By
  - ❖Prof. Indrakka Mali

Index

Introduction

- ❑ Hotel Reservation System is a application, developed for hotel room booking.
- ❑ It takes customer details like customer name and contact number and stores in the database.
- ❑Customers can book any available room and can also order menu according to there choice.
- ❑ The application is easy to use, it also validate the information at the time of storing so that later user will not face any inconvenience.
- ❑It keeps the record of customer's room and menu.
- ❑Hotel manager can add menu and room in the database.

Objective

- ❑ The purpose of the hotel reservation system is to automate the existing system manual system with the help of computerized equipment and full-fledged computer software, fulfilling their requirements so that their valuable data/information can be stored with easy accessing and manipulation of the same. The required software and hardware are easy to work with.
- ❑ Hotel reservation system as described above, can lead to error-free, secure, reliable and fast management systems.

- It can assist the user to concentrate on their activities rather to concentrate on the record keeping.
- Thus, it will help an organization in better utilization of resources.
- The organization can maintain computerized records without redundant entries.
- That means that one need not be distracted by information that is not relevant while being able to reach the information.

Software & hardware used

- ❑ Software Used
  - ❖Operating System : Windows 11
  - ❖Language : Java
  - ❖IDE : Spring Boot, Postman
  - ❖Backend : MySQL
- ❑ Hardware Used
  - ❖ CPU : Intel core i3
  - ❖RAM : 8GB
  - ❖Hard Disk : 1TB Hard Disk or minimum

Back End

- ❑Core Java
- ❑Spring Boot
- ❑Spring Data JPA
- ❑Hibernate
- ❑Spring Boot Web
- ❑MySQL database

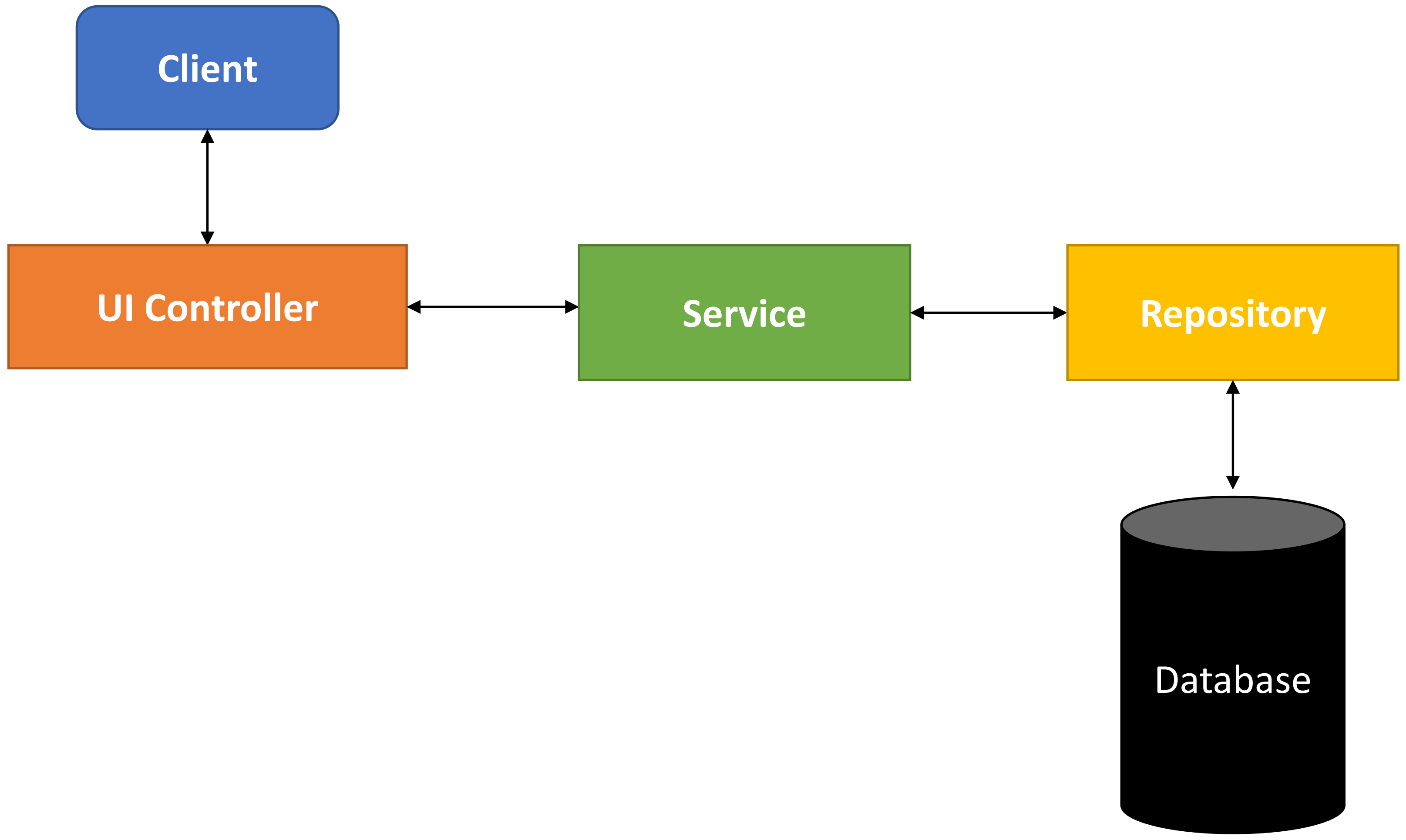
Proposed System

- ❑Customer can add there information, and book hotel room and menu according to their choice.
- ❑ Customer can see their bill and make payment according to that bill.
- ❑Hotel manager can add Menu and Rooms in the database and fix their prices according to their standard rates.

About Back-End

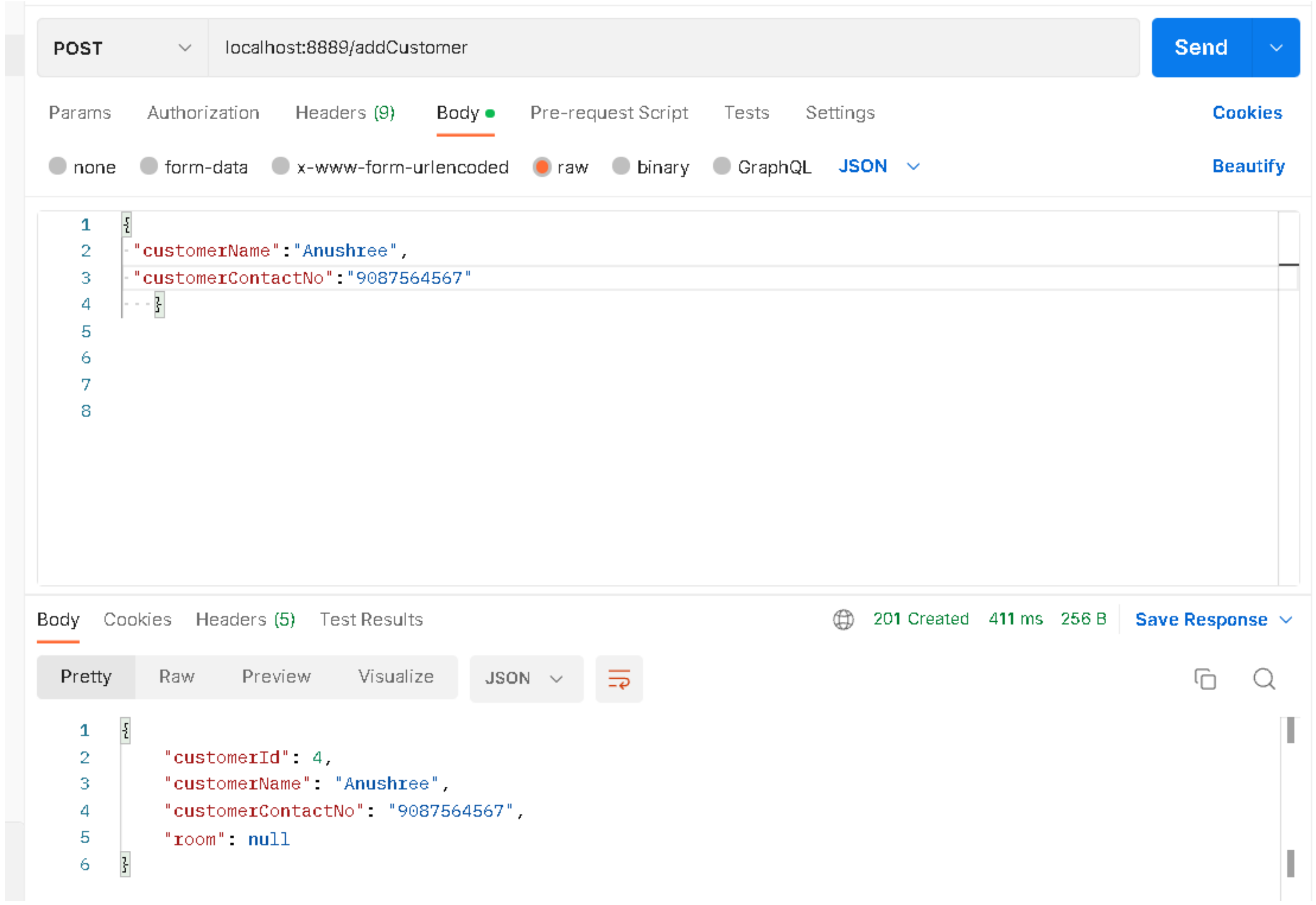
- ❑There are mainly four operation will be performed by the user ex. Insert, update, retrieve /fetch, delete.
- ❑These operation will be performed using the spring boot framework, core java and spring boot web , spring data JPA, and hibernate.
- ❑For connectivity purpose we are using MySQL database. It is connected to the java and database.

Working of Back-End



- **UI Controller** - In Spring Boot, the UI Controller class is responsible for processing incoming requests, preparing a model, and returning the view to be rendered as a response.
- **Service** - Components are the class file which contains @Service annotation. These class files are used to write business logic in a different layer.
- **Repository** - Repository is a specialization of @Component annotation which is used to indicate that the class provides the mechanism for storage, retrieval, update, delete and search operation on objects. Repository is directly connected with the database and then it can return to the repository and response to the service and it can return to the controller and then response send to the client

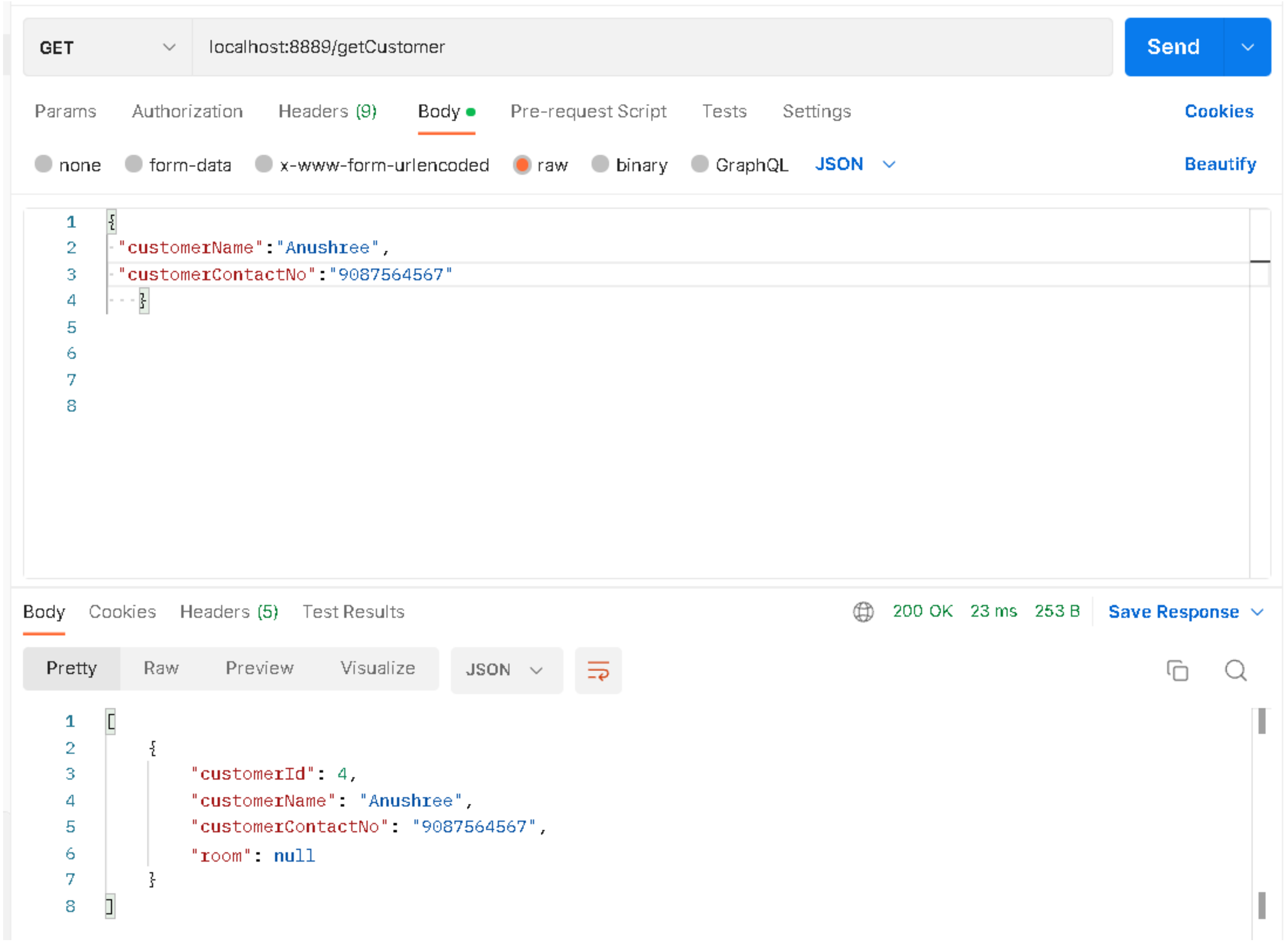
Snapshots


POST localhost:8889/addCustomer

```
{
"customerName":"Anushree",
"customerContactNo":"9087564567"
  }
```

```
{
    "customerId": 4,
    "customerName": "Anushree",
    "customerContactNo": "9087564567",
    "room": null
}
```

Snapshots

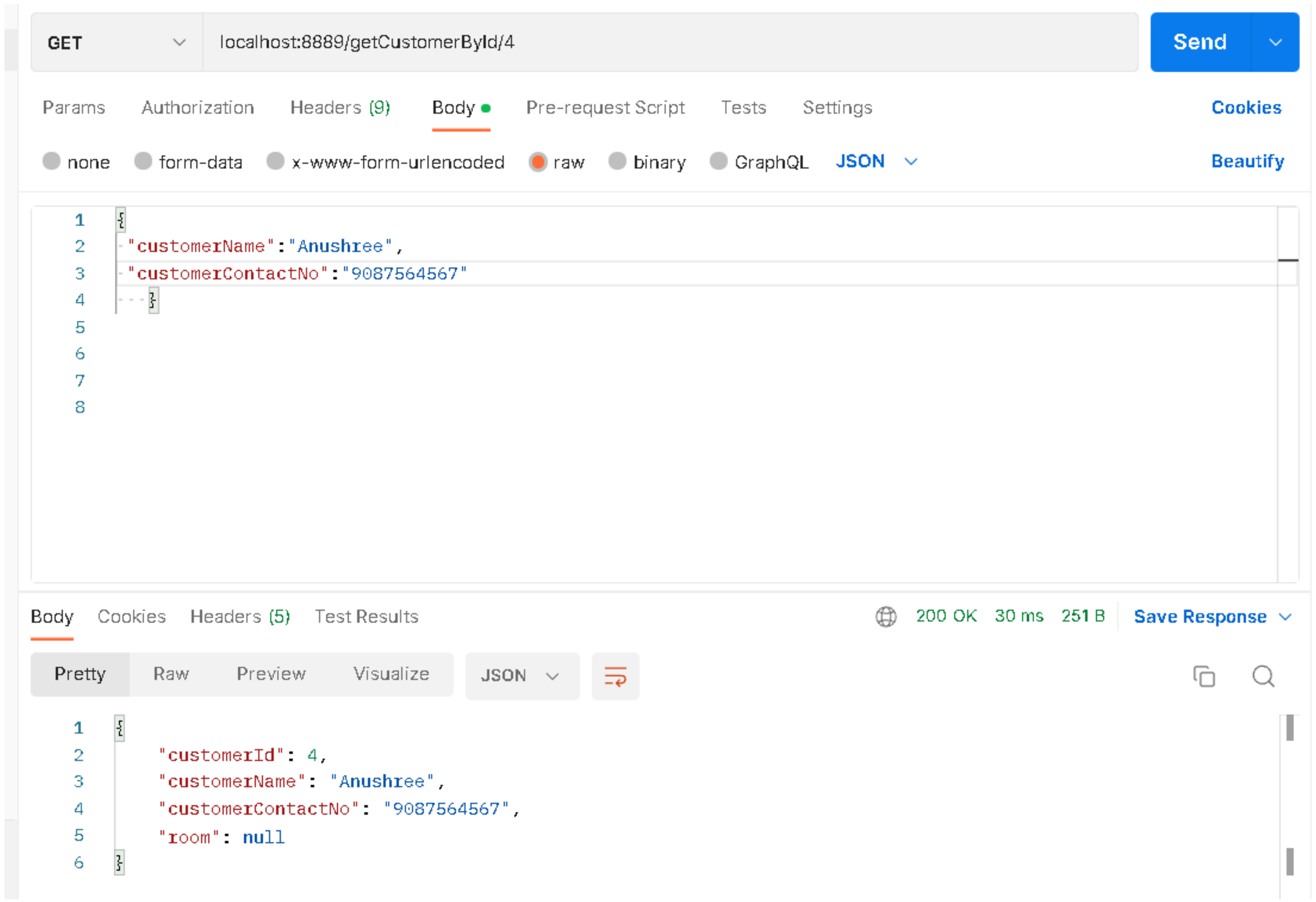


GET localhost:8889/getCustomer

```
{
"customerName":"Anushree",
"customerContactNo":"9087564567"
   }
```

200 OK 23 ms 253 B

```
[
    {
        "customerId": 4,
        "customerName": "Anushree",
        "customerContactNo": "9087564567",
        "room": null
    }
]
```

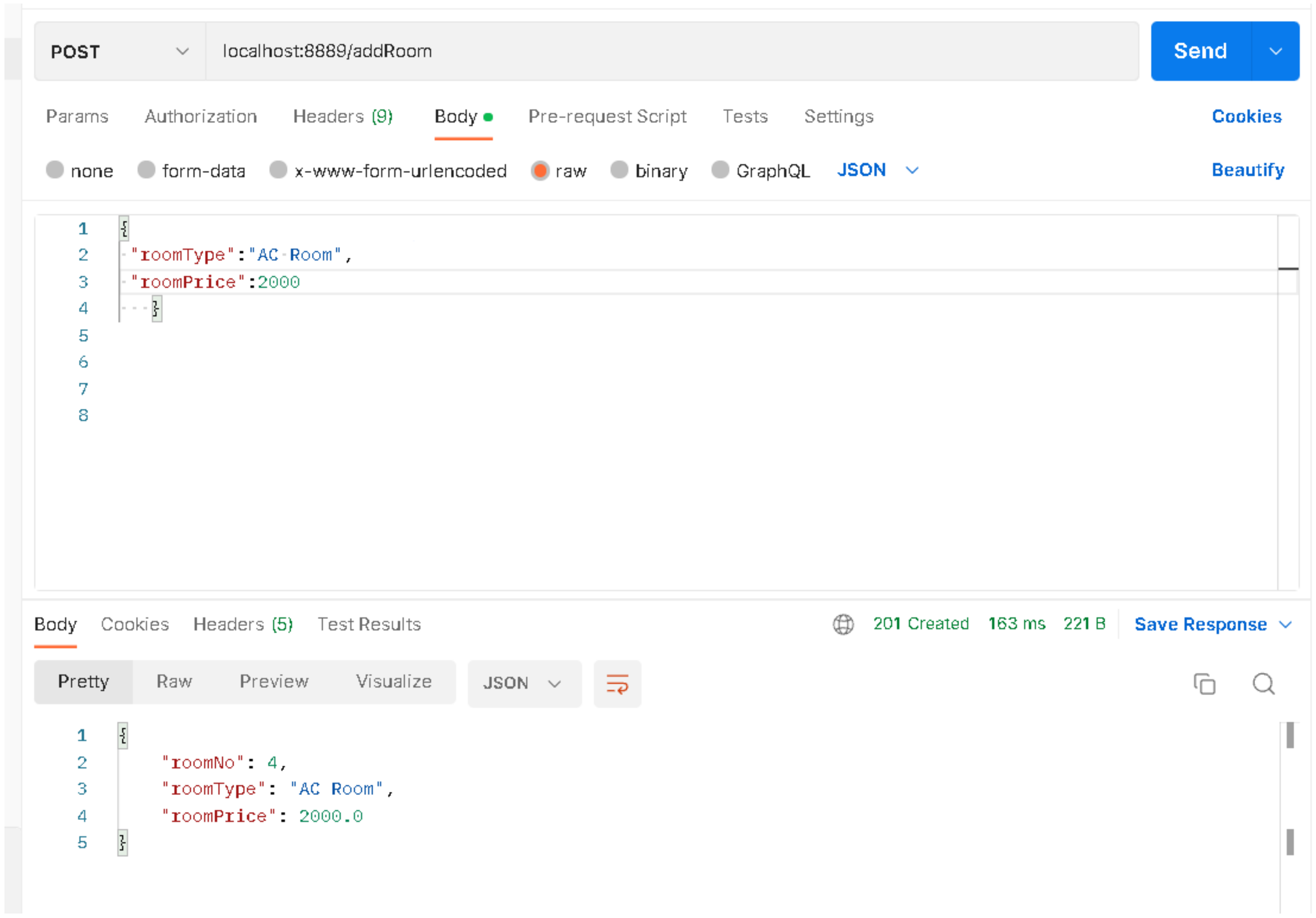


localhost:8889/getCustomerById/4

```
{
"customerName":"Anushree",
"customerContactNo":"9087564567"
}
```

200 OK 30 ms 251 B

```
{
    "customerId": 4,
    "customerName": "Anushree",
    "customerContactNo": "9087564567",
    "room": null
}
```


POST localhost:8889/addRoom

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

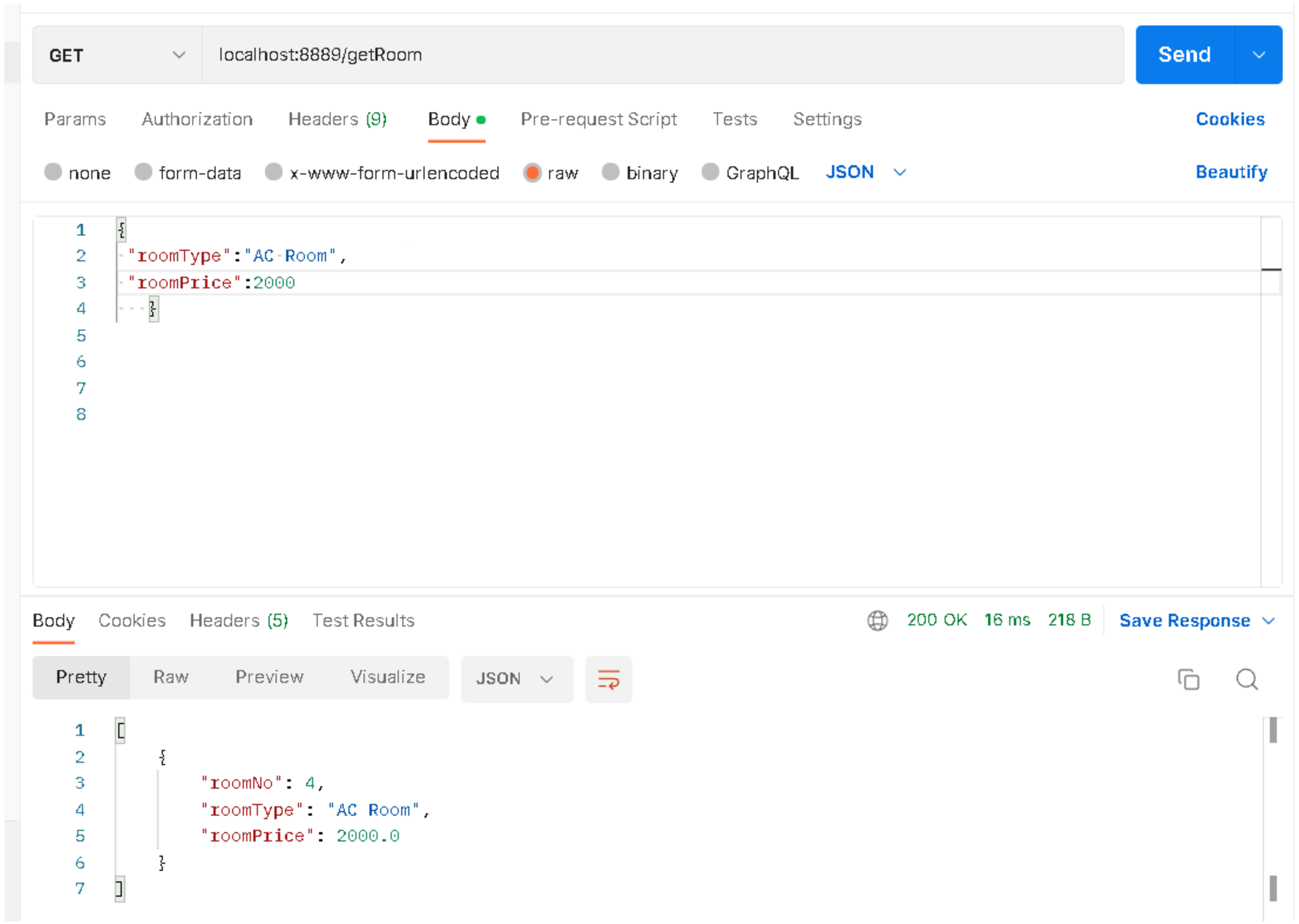
none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON Beautify

```
{
"roomType":"AC Room",
"roomPrice":2000
}
```

Body Cookies Headers (5) Test Results 201 Created 163 ms 221 B Save Response

Pretty Raw Preview Visualize JSON

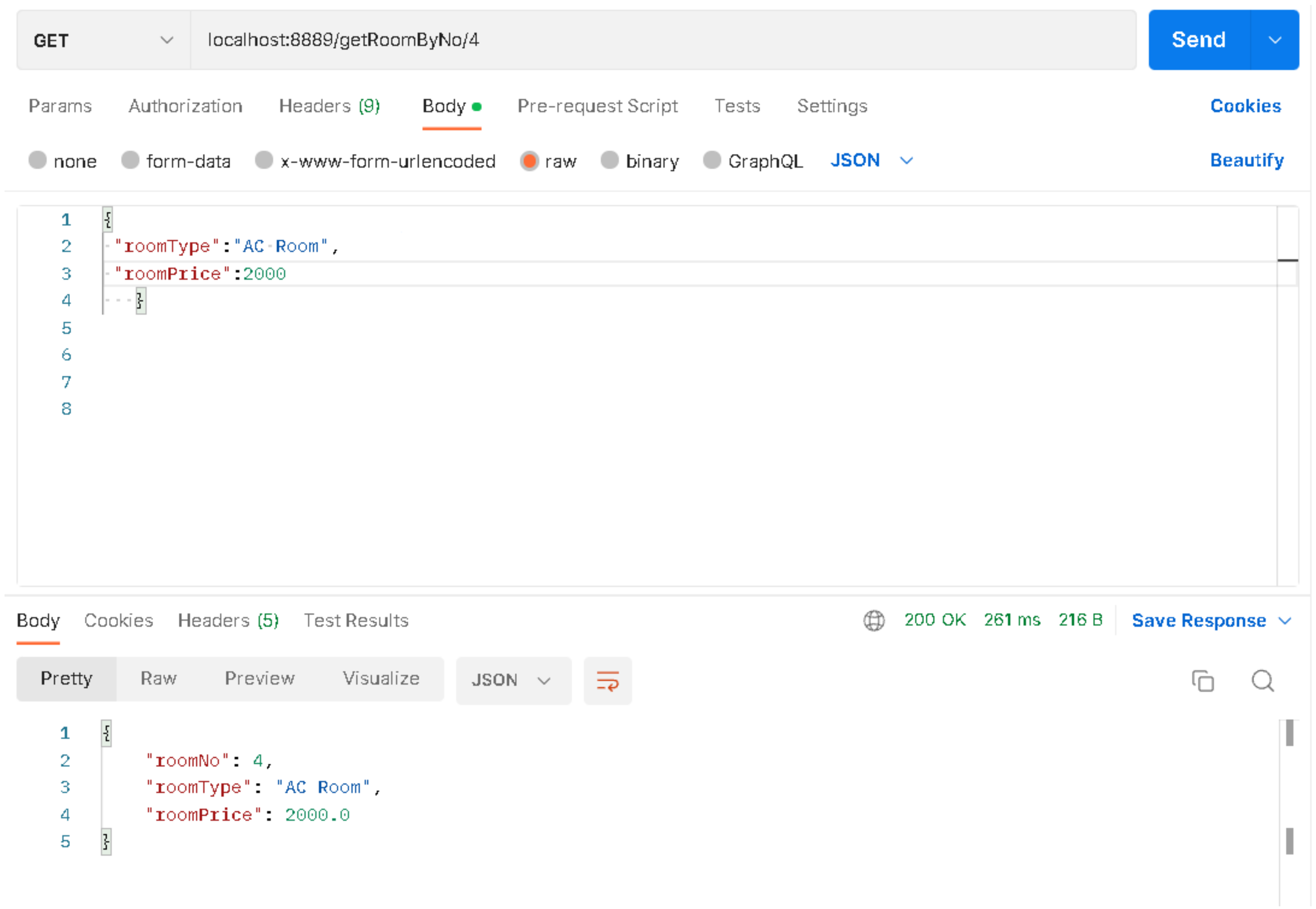
```
{
    "roomNo": 4,
    "roomType": "AC Room",
    "roomPrice": 2000.0
}
```



```
localhost:8889/getRoom
```

```
{
"roomType":"AC Room",
"roomPrice":2000
  }
```

```
[
    {
        "roomNo": 4,
        "roomType": "AC Room",
        "roomPrice": 2000.0
    }
]
```


GET localhost:8889/getRoomByNo/4

Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

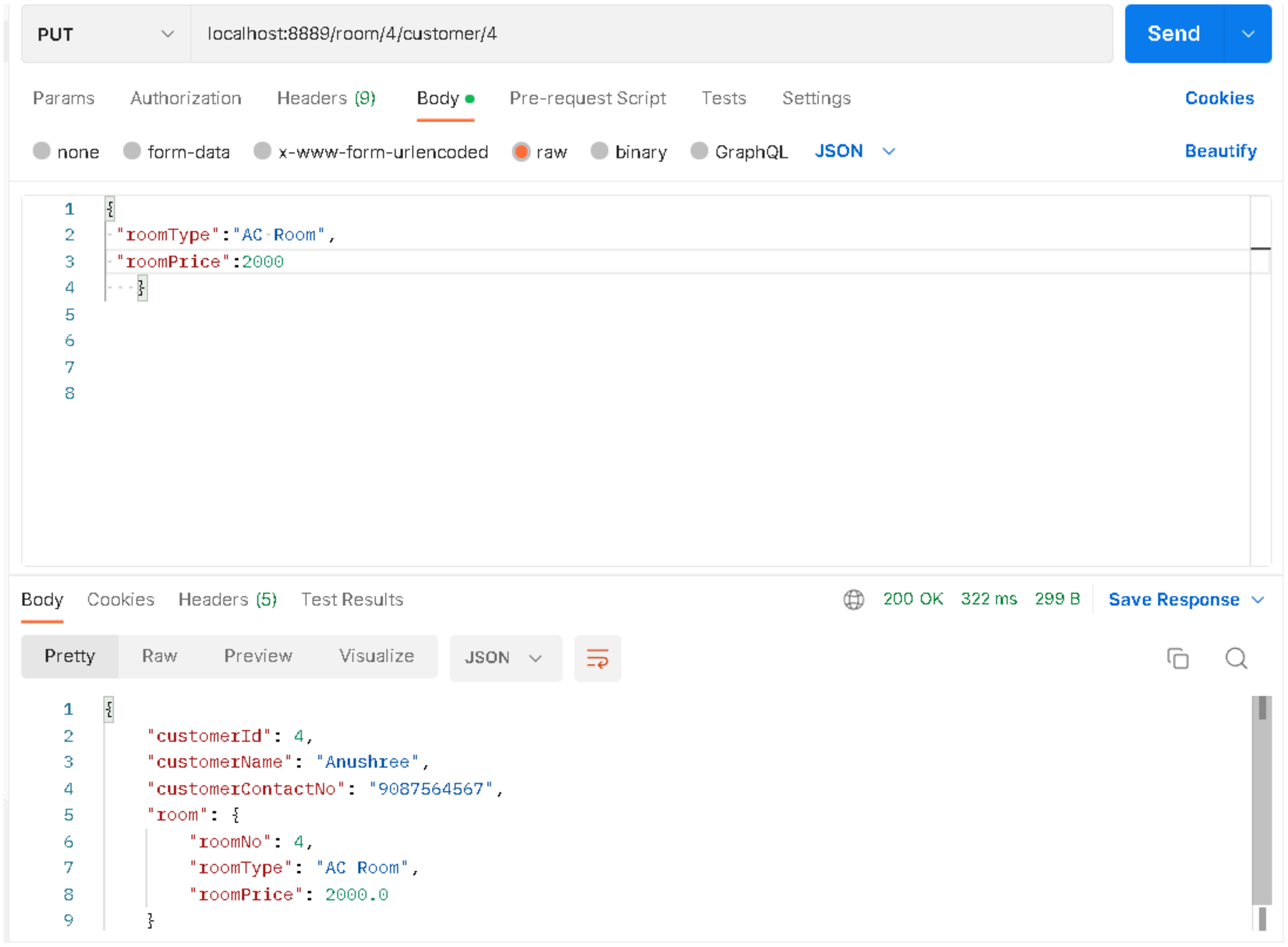
none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON Beautify

```
{
"roomType":"AC Room",
"roomPrice":2000
   }
```

Body Cookies Headers (5) Test Results 200 OK 261 ms 216 B Save Response

Pretty Raw Preview Visualize JSON

```
{
    "roomNo": 4,
    "roomType": "AC Room",
    "roomPrice": 2000.0
}
```

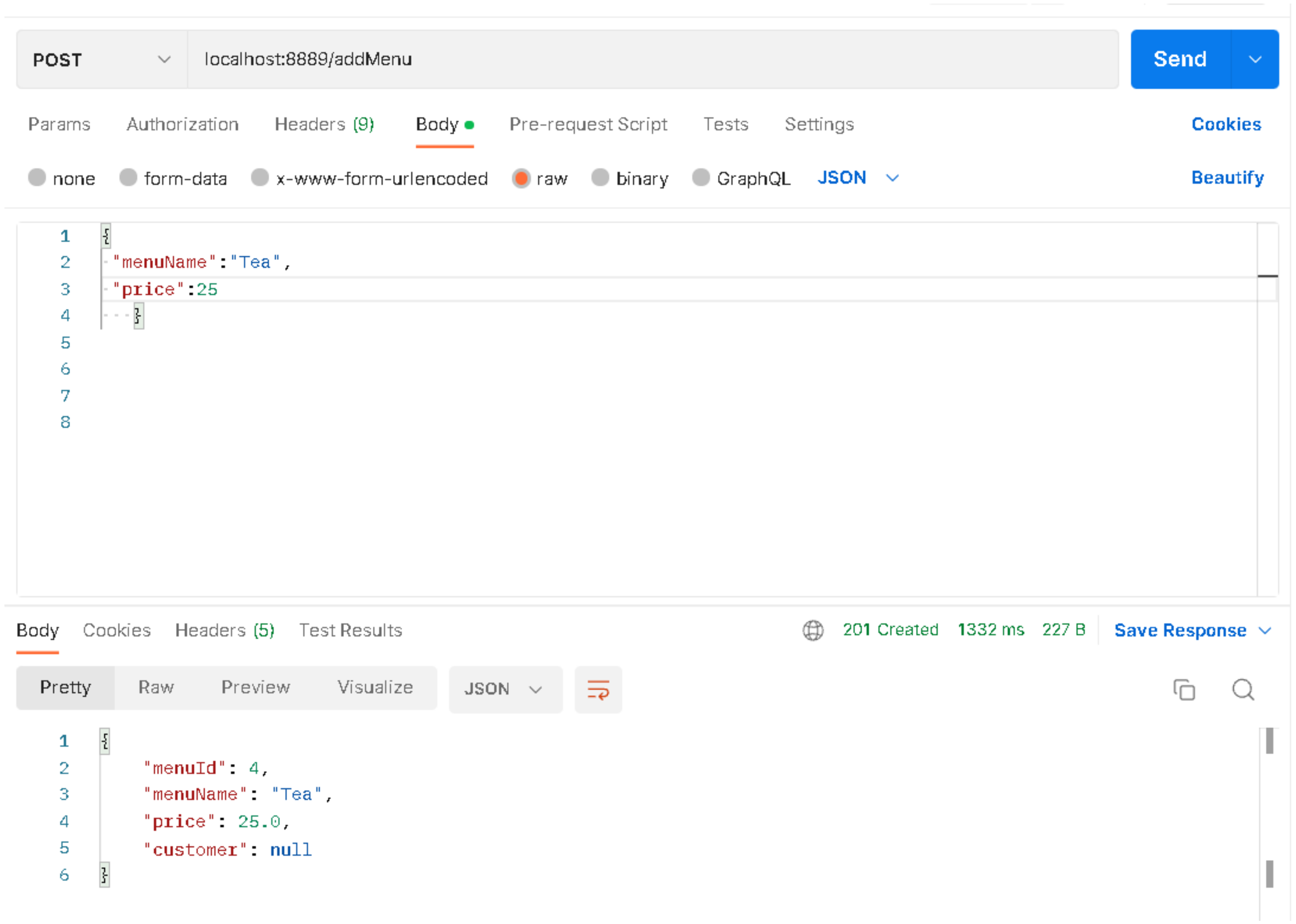


PUT localhost:8889/room/4/customer/4

```
{
"roomType":"AC Room",
"roomPrice":2000
   }
```

200 OK 322 ms 299 B

```
{
    "customerId": 4,
    "customerName": "Anushree",
    "customerContactNo": "9087564567",
    "room": {
        "roomNo": 4,
        "roomType": "AC Room",
        "roomPrice": 2000.0
    }
```

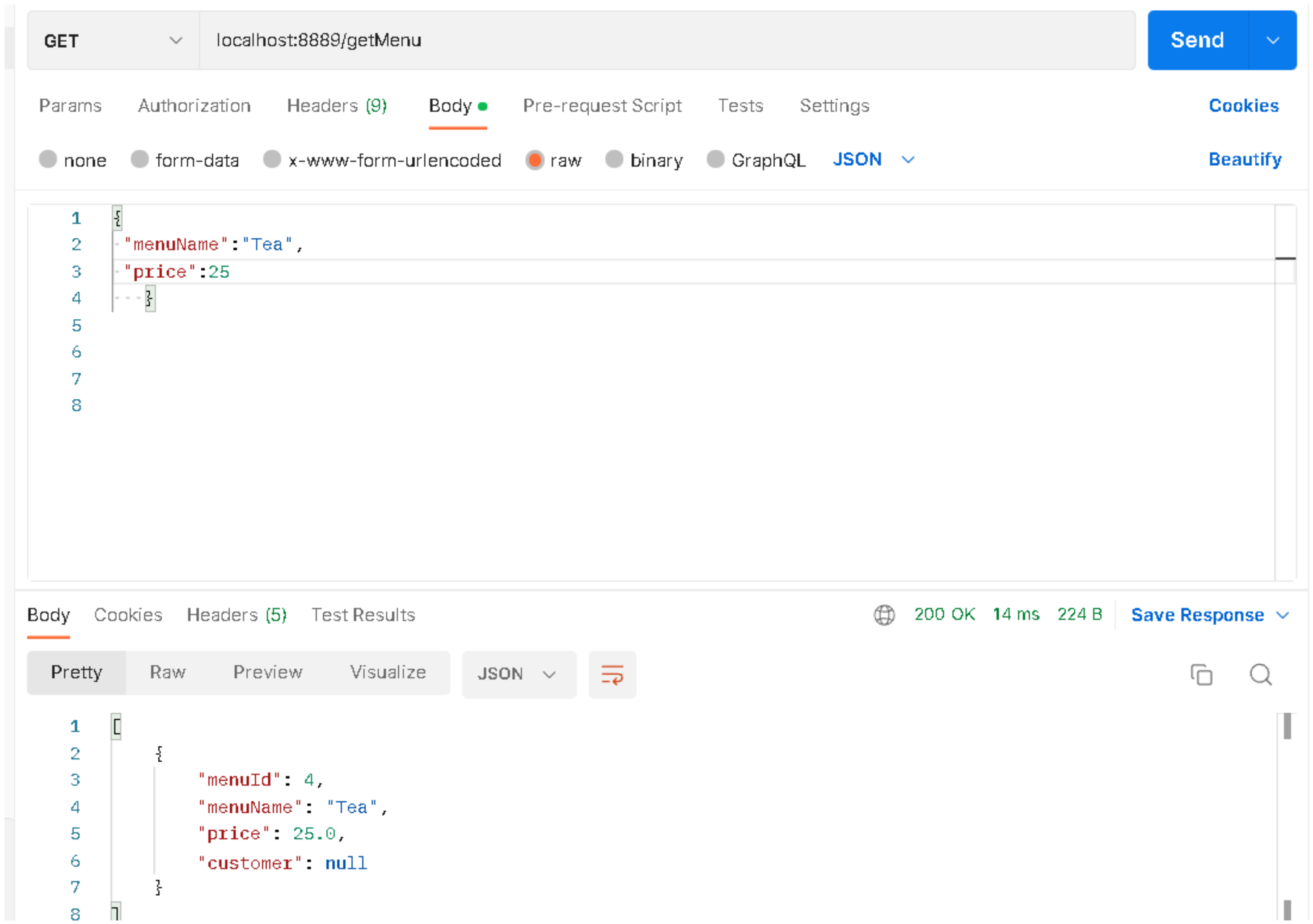


```
localhost:8889/addMenu
```

```
{
"menuName":"Tea",
"price":25
}
```

```
201 Created   1332 ms   227 B
```

```
{
    "menuId": 4,
    "menuName": "Tea",
    "price": 25.0,
    "customer": null
}
```



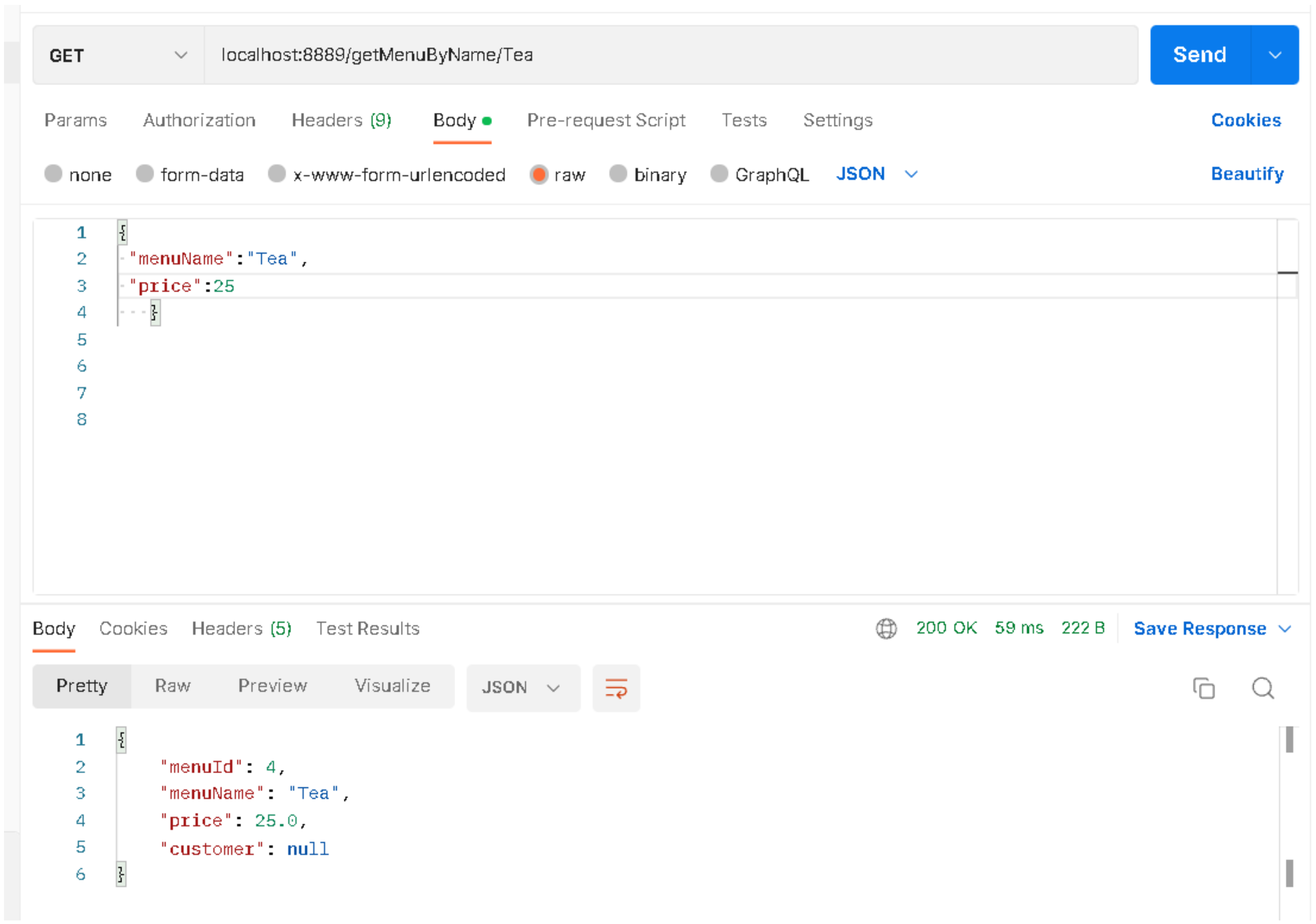
```
localhost:8889/getMenu
```

Body: [✓ raw] JSON

```
{
"menuName":"Tea",
"price":25
  }
```

200 OK 14 ms 224 B

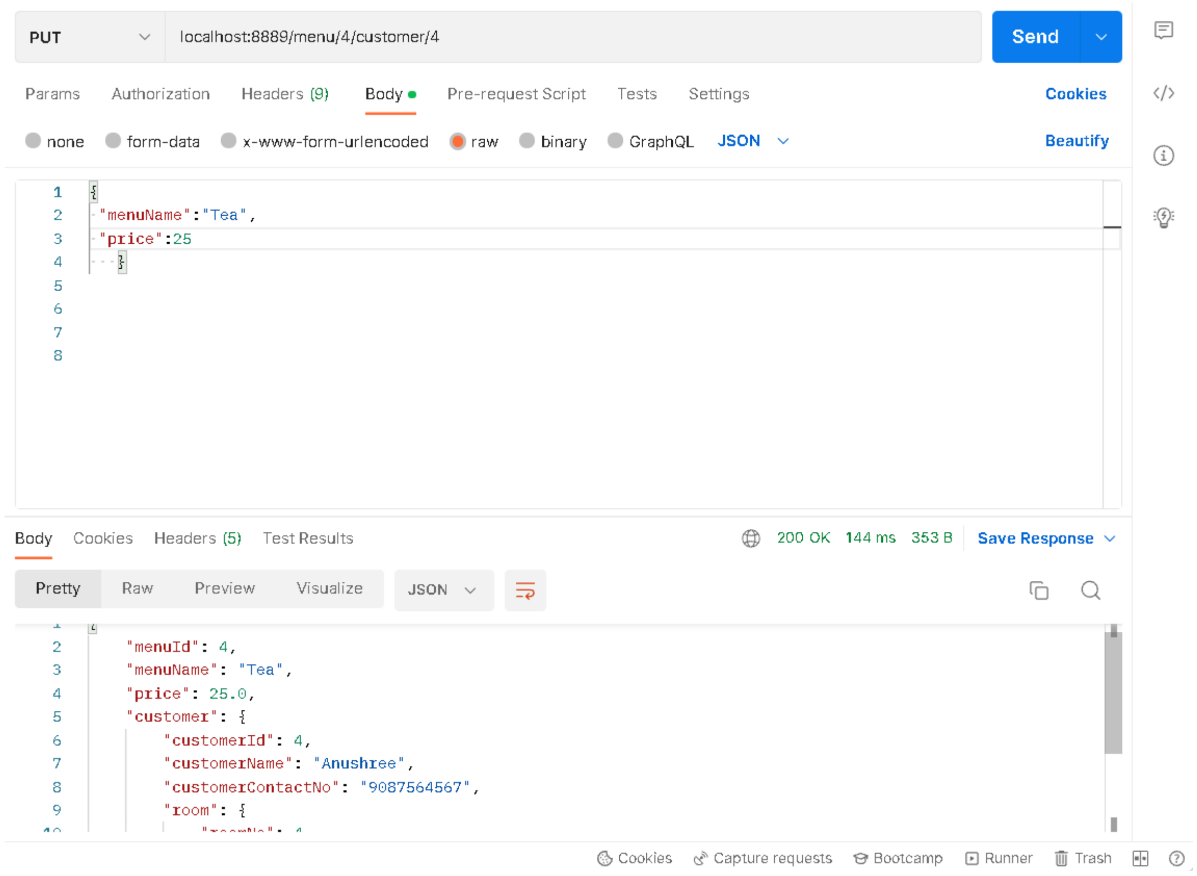
```
[
    {
        "menuId": 4,
        "menuName": "Tea",
        "price": 25.0,
        "customer": null
    }
]
```


GET localhost:8889/getMenuByName/Tea

```
{
"menuName":"Tea",
"price":25
    }
```

200 OK 59 ms 222 B

```
{
    "menuId": 4,
    "menuName": "Tea",
    "price": 25.0,
    "customer": null
}
```

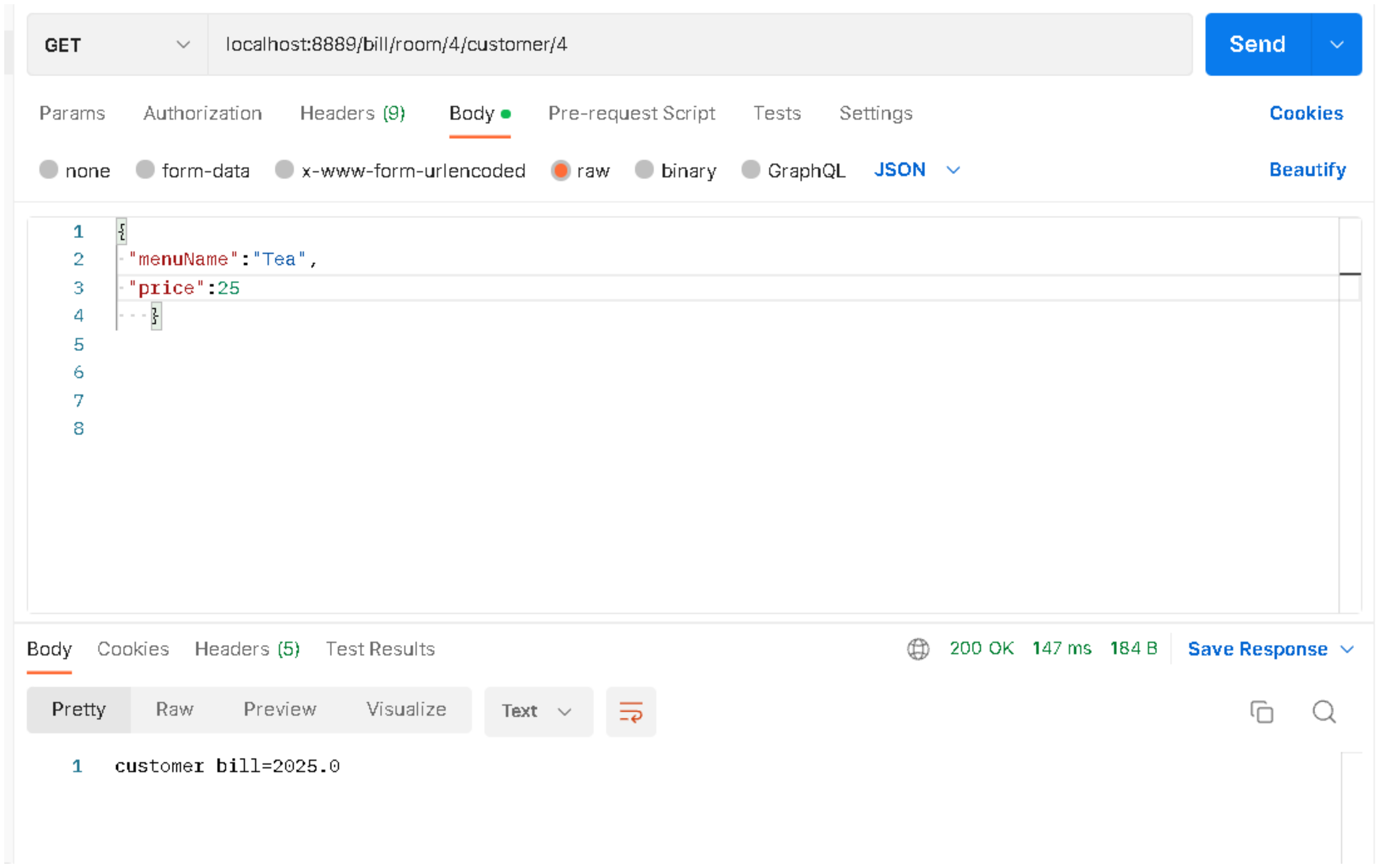


localhost:8889/menu/4/customer/4

```
{
"menuName":"Tea",
"price":25
   }
```

200 OK 144 ms 353 B

```
{
    "menuId": 4,
    "menuName": "Tea",
    "price": 25.0,
    "customer": {
        "customerId": 4,
        "customerName": "Anushree",
        "customerContactNo": "9087564567",
        "room": {
```



GET localhost:8889/bill/room/4/customer/4

Send

Params Authorization Headers (9) Body Pre-request Script Tests Settings Cookies

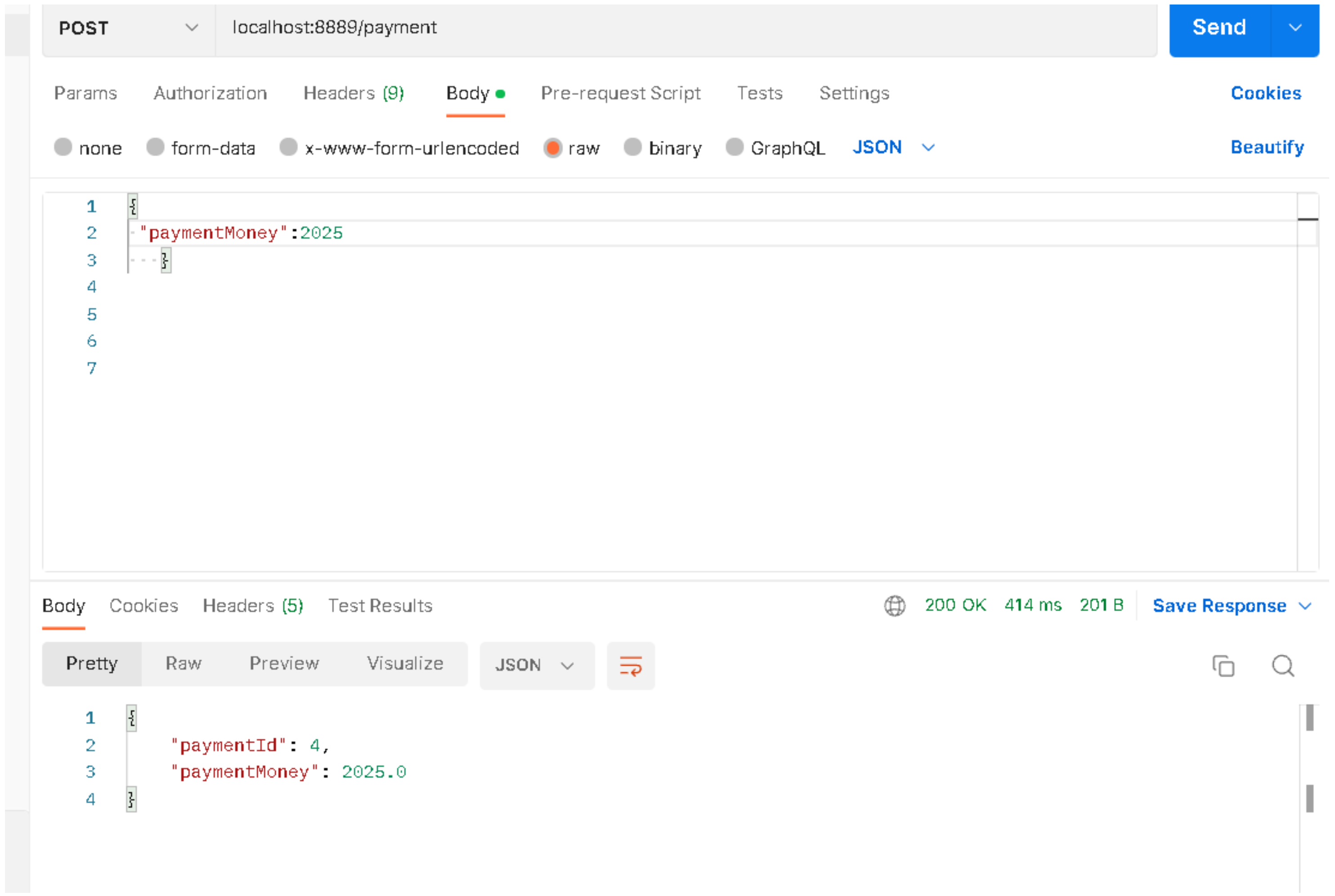
none form-data x-www-form-urlencoded [✓ raw] binary GraphQL JSON Beautify

```
{
"menuName":"Tea",
"price":25
}
```

Body Cookies Headers (5) Test Results 200 OK 147 ms 184 B Save Response

Pretty Raw Preview Visualize Text

```
customer bill=2025.0
```

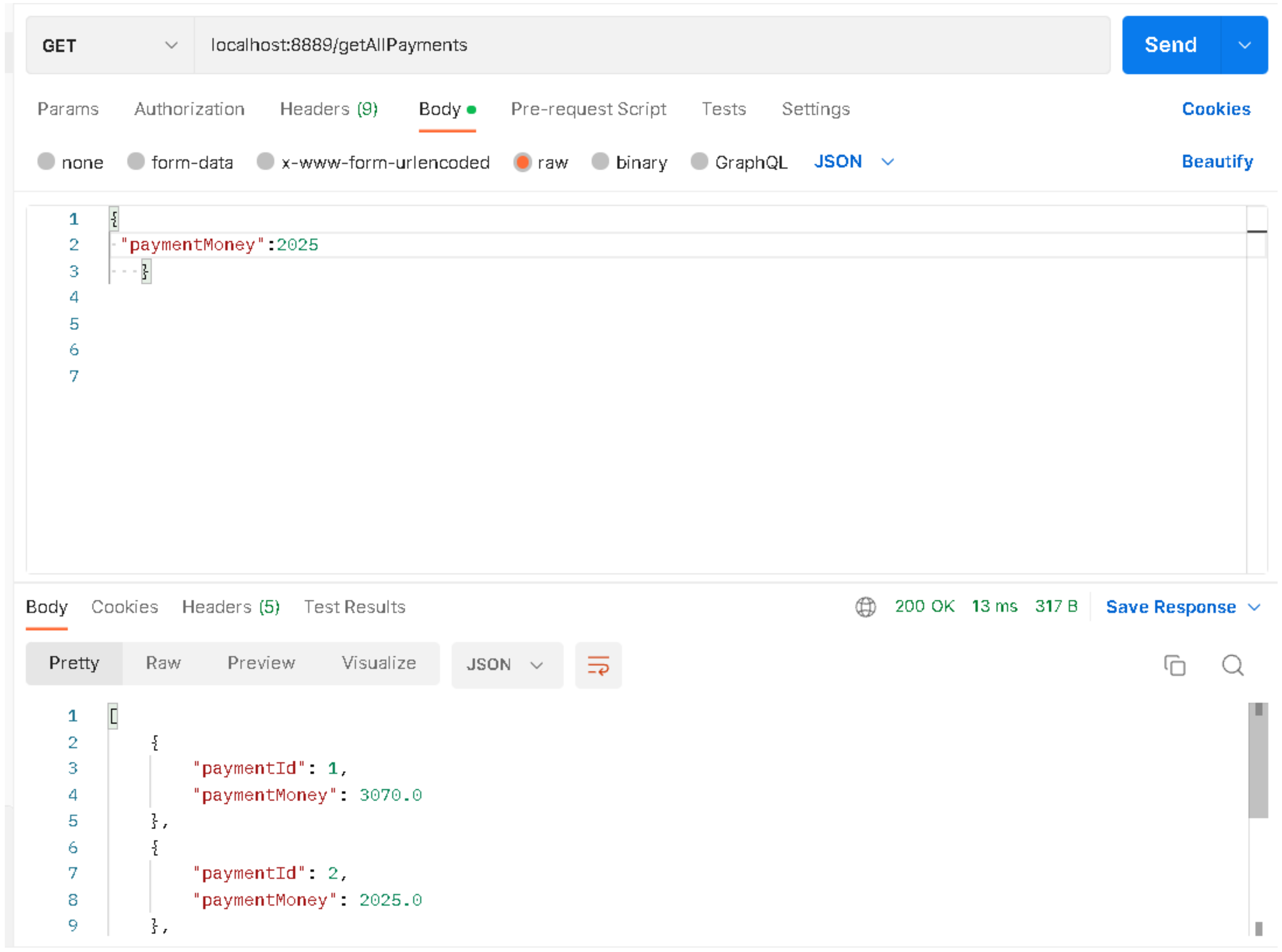


localhost:8889/payment

```
{
"paymentMoney":2025
  }
```

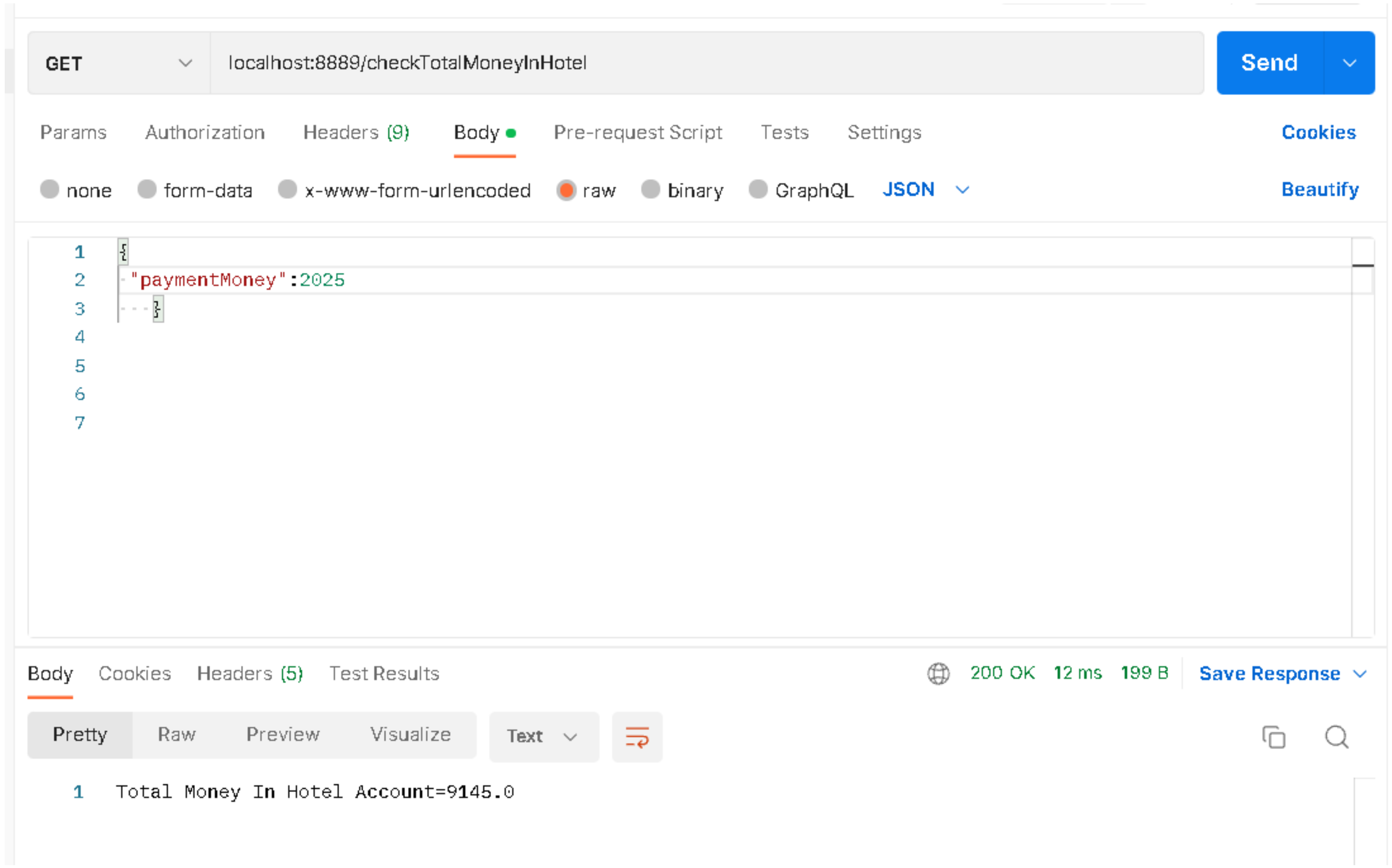
200 OK 414 ms 201 B

```
{
    "paymentId": 4,
    "paymentMoney": 2025.0
}
```



```
{
"paymentMoney":2025
}
```

```
[
    {
        "paymentId": 1,
        "paymentMoney": 3070.0
    },
    {
        "paymentId": 2,
        "paymentMoney": 2025.0
    },
```

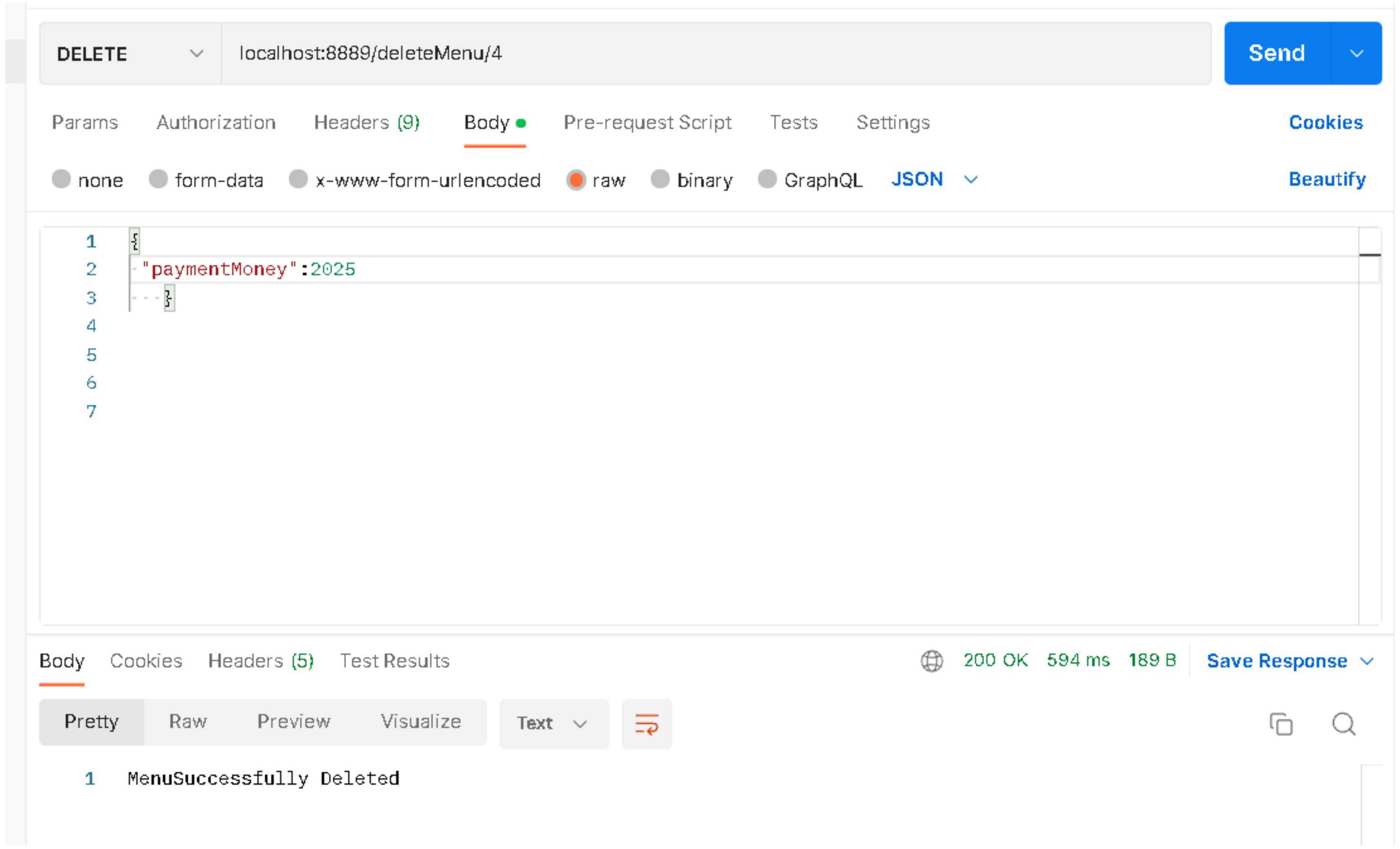


localhost:8889/checkTotalMoneyInHotel

```
{
 "paymentMoney":2025
   }
```

200 OK 12 ms 199 B

```
Total Money In Hotel Account=9145.0
```

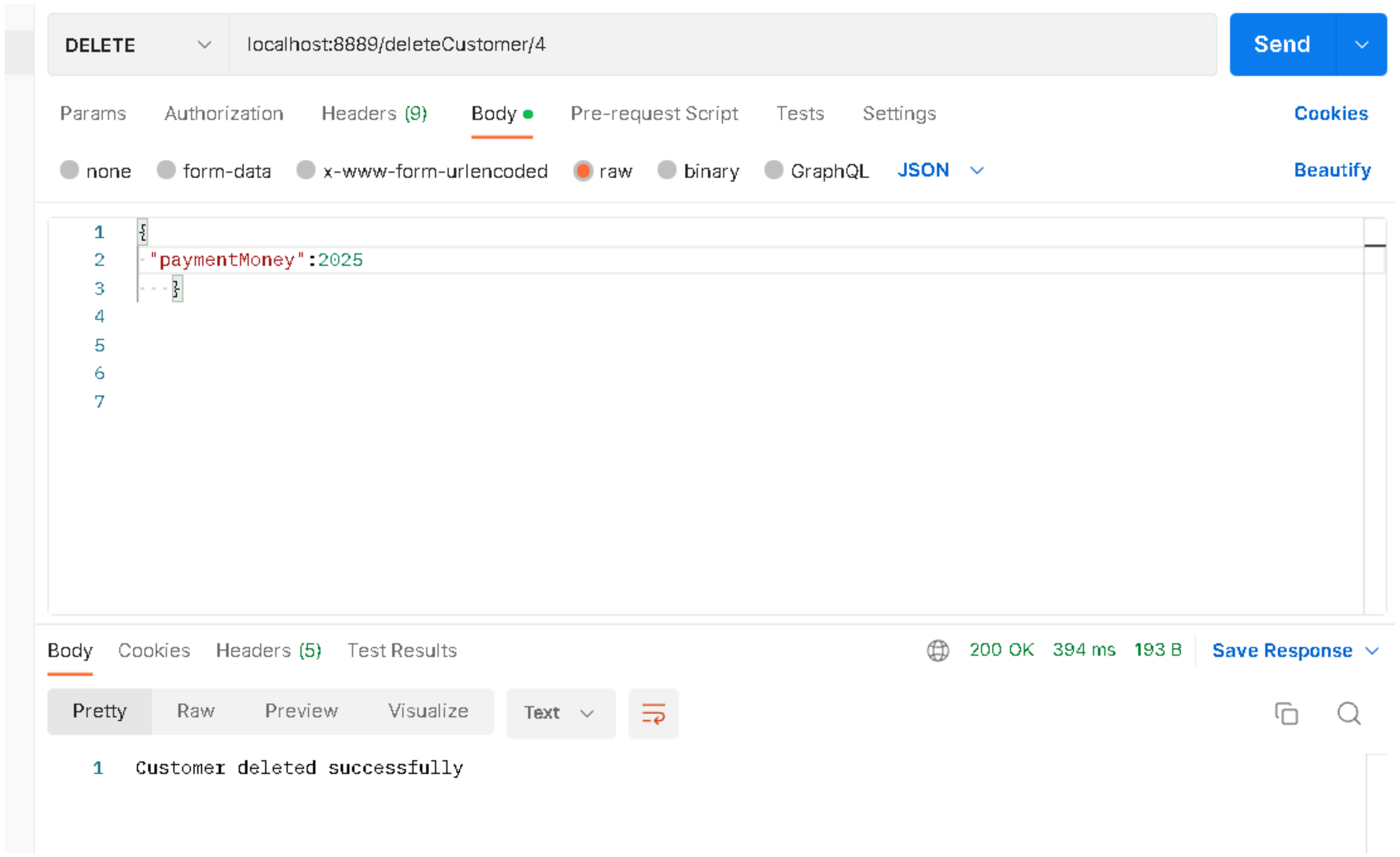


localhost:8889/deleteMenu/4

```
{
"paymentMoney":2025
  }
```

200 OK 594 ms 189 B

```
MenuSuccessfully Deleted
```



localhost:8889/deleteCustomer/4

```
{
"paymentMoney":2025
    }
```

200 OK 394 ms 193 B

```
Customer deleted successfully
```

Advantages and Disadvantages

- ❑ Duplication of the Customer data is avoided.
- ❑ As customer id is already stored in the database therefore human mistakes can be avoided.
- ❑Customer Id is required to fetch the customer details.

Conclusion

- ❑ Effectiveness, efficiency, and reliability are the key aspects that make this web-based Hotel Reservation System very useful for industrial and several other businesses.
- ❑ The proposed project is very flexible to handle new modules and features as per user requirements in future.