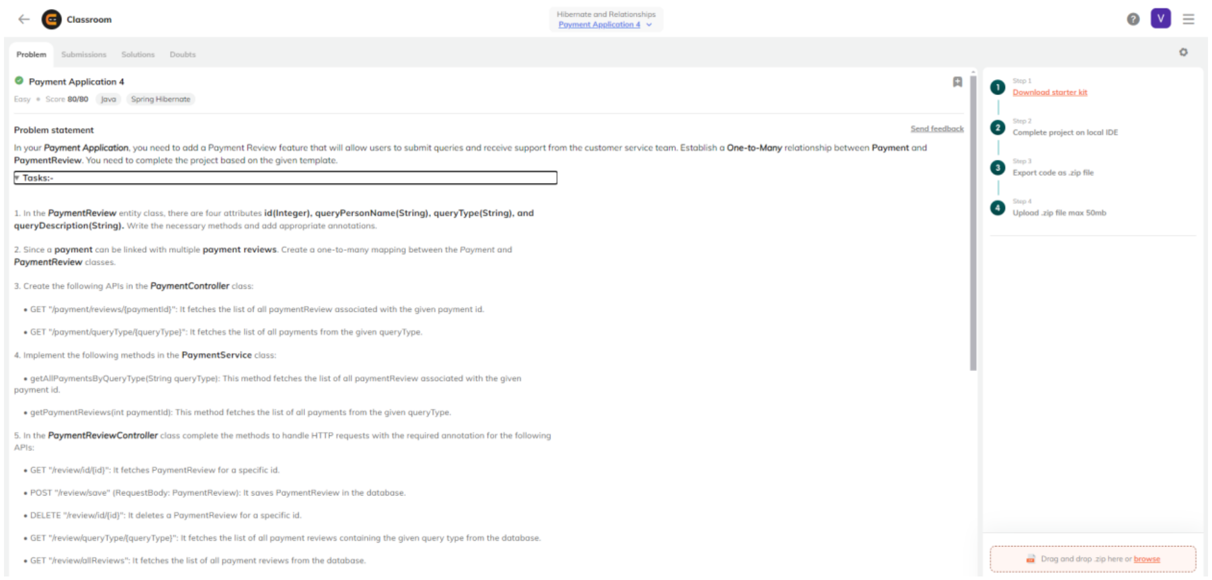
Payment Application IV

Problem Statement:



Problem statement

In your **Payment Application**, you need to add a Payment Review feature that will allow users to submit queries and receive support from the customer service team. Establish a **One-to-Many** relationship between **Payment** and **PaymentReview**. You need to complete the project based on the given template.

▾ **Tasks:-**

1. In the **PaymentReview** entity class, there are four attributes **id(Integer), queryPersonName(String), queryType(String), and queryDescription(String).** Write the necessary methods and add appropriate annotations.

2. Since a **payment** can be linked with multiple **payment reviews**. Create a one-to-many mapping between the Payment and **PaymentReview** classes.

3. Create the following APIs in the **PaymentController** class:

   • GET "/payment/reviews/{paymentId}": It fetches the list of all paymentReview associated with the given payment id.

   • GET "/payment/queryType/{queryType}": It fetches the list of all payments from the given queryType.

4. Implement the following methods in the **PaymentService** class:

   • getAllPaymentsByQueryType(String queryType): This method fetches the list of all paymentReview associated with the given payment id.

   • getPaymentReviews(int paymentId): This method fetches the list of all payments from the given queryType.

5. In the **PaymentReviewController** class complete the methods to handle HTTP requests with the required annotation for the following APIs:

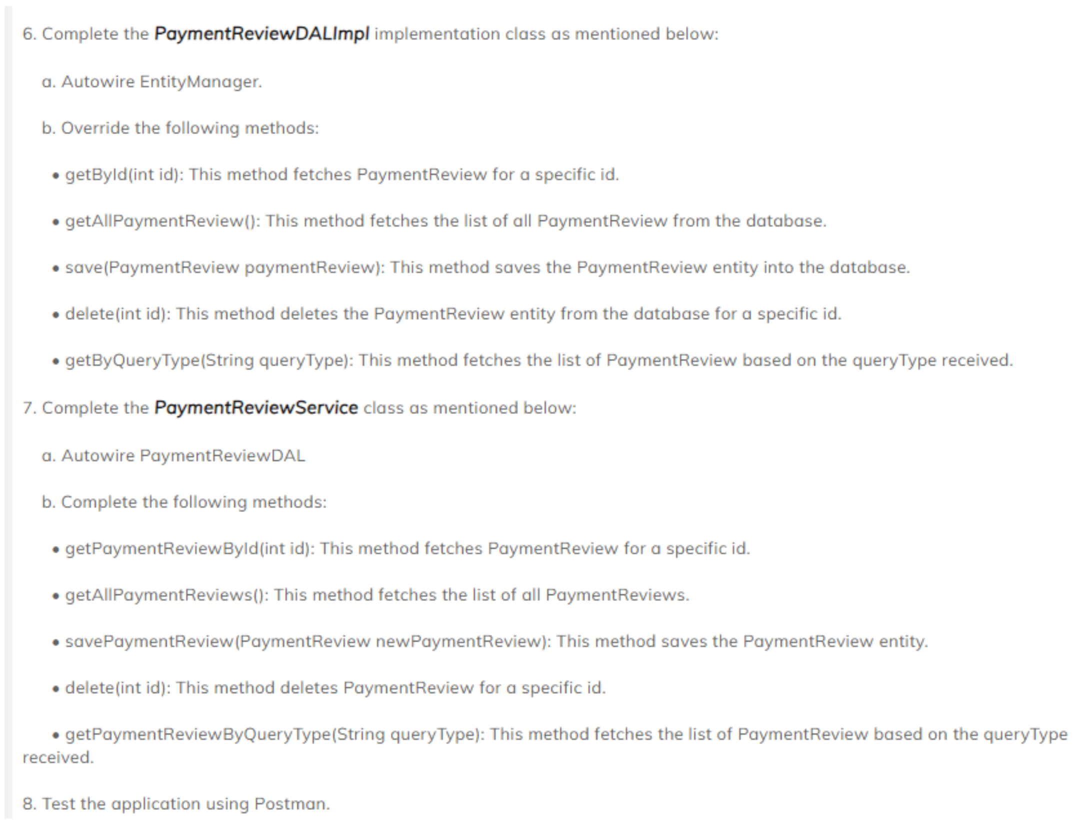
   • GET "/review/id/{id}": It fetches PaymentReview for a specific id.

   • POST "/review/save" (RequestBody: PaymentReview): It saves PaymentReview in the database.

   • DELETE "/review/id/{id}": It deletes a PaymentReview for a specific id.

   • GET "/review/queryType/{queryType}": It fetches the list of all payment reviews containing the given query type from the database.

   • GET "/review/allReviews": It fetches the list of all payment reviews from the database.



6. Complete the **PaymentReviewDALImpl** implementation class as mentioned below:

   a. Autowire EntityManager.

   b. Override the following methods:

   • getById(int id): This method fetches PaymentReview for a specific id.

   • getAllPaymentReview(): This method fetches the list of all PaymentReview from the database.

   • save(PaymentReview paymentReview): This method saves the PaymentReview entity into the database.

   • delete(int id): This method deletes the PaymentReview entity from the database for a specific id.

   • getByQueryType(String queryType): This method fetches the list of PaymentReview based on the queryType received.

7. Complete the **PaymentReviewService** class as mentioned below:

   a. Autowire PaymentReviewDAL

   b. Complete the following methods:

   • getPaymentReviewById(int id): This method fetches PaymentReview for a specific id.

   • getAllPaymentReviews(): This method fetches the list of all PaymentReviews.

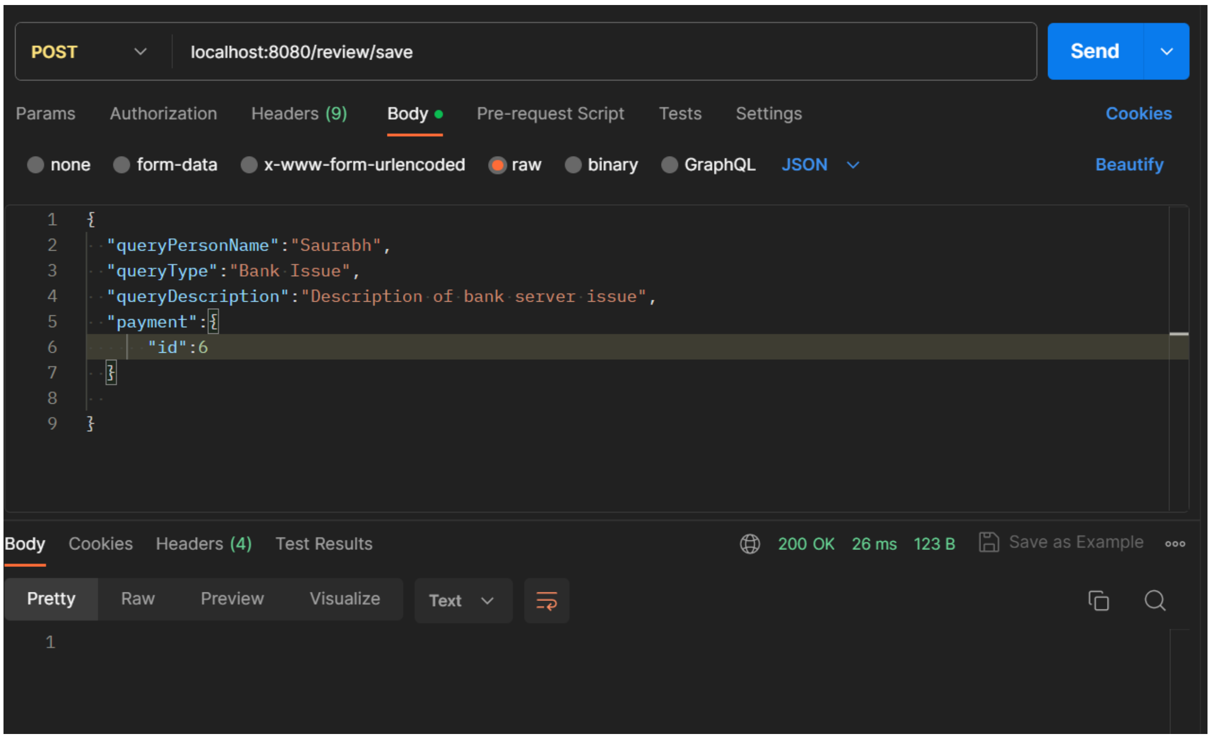
   • savePaymentReview(PaymentReview newPaymentReview): This method saves the PaymentReview entity.

   • delete(int id): This method deletes PaymentReview for a specific id.

   • getPaymentReviewByQueryType(String queryType): This method fetches the list of PaymentReview based on the queryType received.

8. Test the application using Postman.

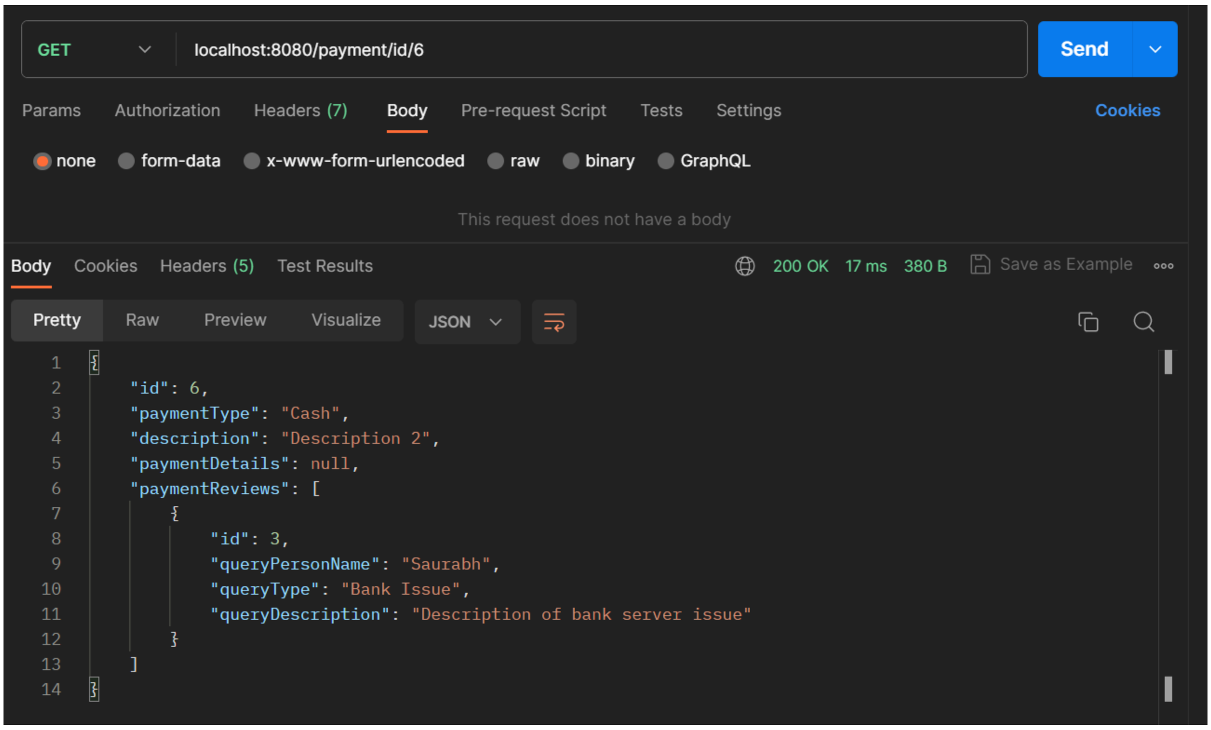
Output:



POST localhost:8080/review/save

```
{
  "queryPersonName":"Saurabh",
  "queryType":"Bank Issue",
  "queryDescription":"Description of bank server issue",
  "payment":{
      "id":6
  }
  
}
```

200 OK 26 ms 123 B

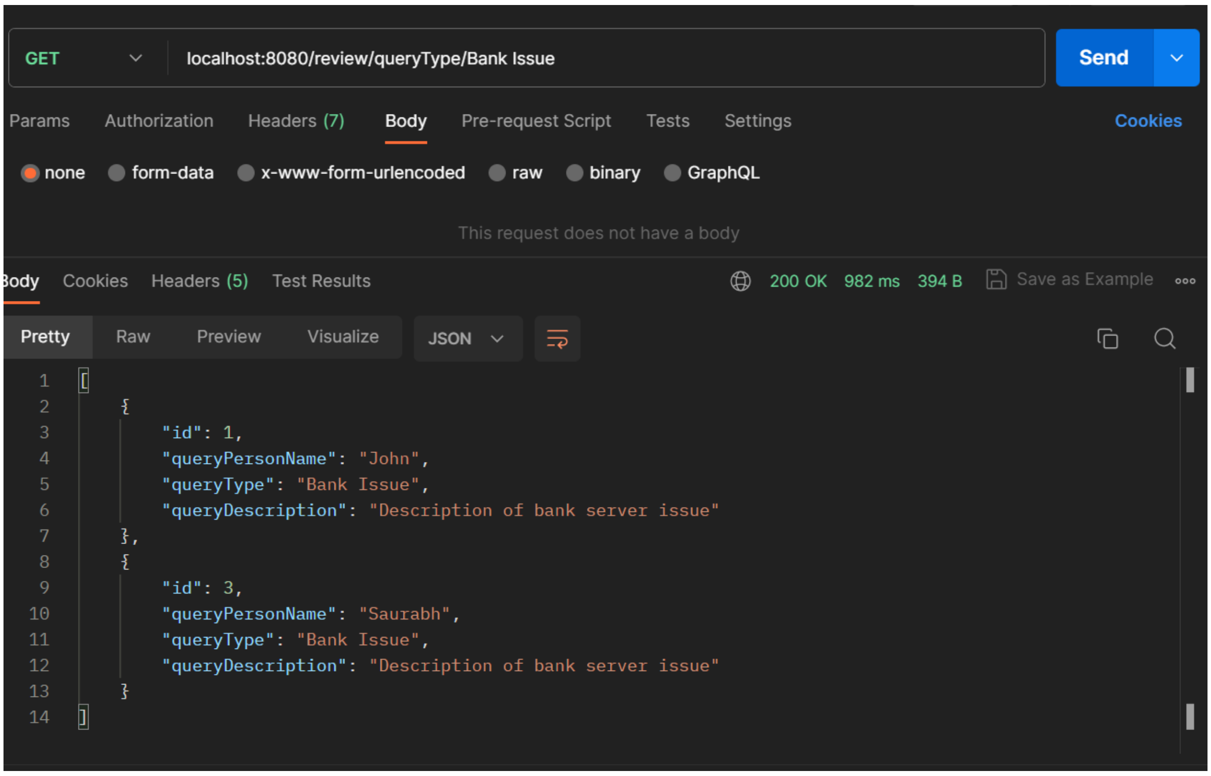


GET localhost:8080/payment/id/6

This request does not have a body

200 OK 17 ms 380 B

```
{
    "id": 6,
    "paymentType": "Cash",
    "description": "Description 2",
    "paymentDetails": null,
    "paymentReviews": [
        {
            "id": 3,
            "queryPersonName": "Saurabh",
            "queryType": "Bank Issue",
            "queryDescription": "Description of bank server issue"
        }
    ]
}
```



```
localhost:8080/review/queryType/Bank Issue
```

This request does not have a body

200 OK 982 ms 394 B

```
[
    {
        "id": 1,
        "queryPersonName": "John",
        "queryType": "Bank Issue",
        "queryDescription": "Description of bank server issue"
    },
    {
        "id": 3,
        "queryPersonName": "Saurabh",
        "queryType": "Bank Issue",
        "queryDescription": "Description of bank server issue"
    }
]
```