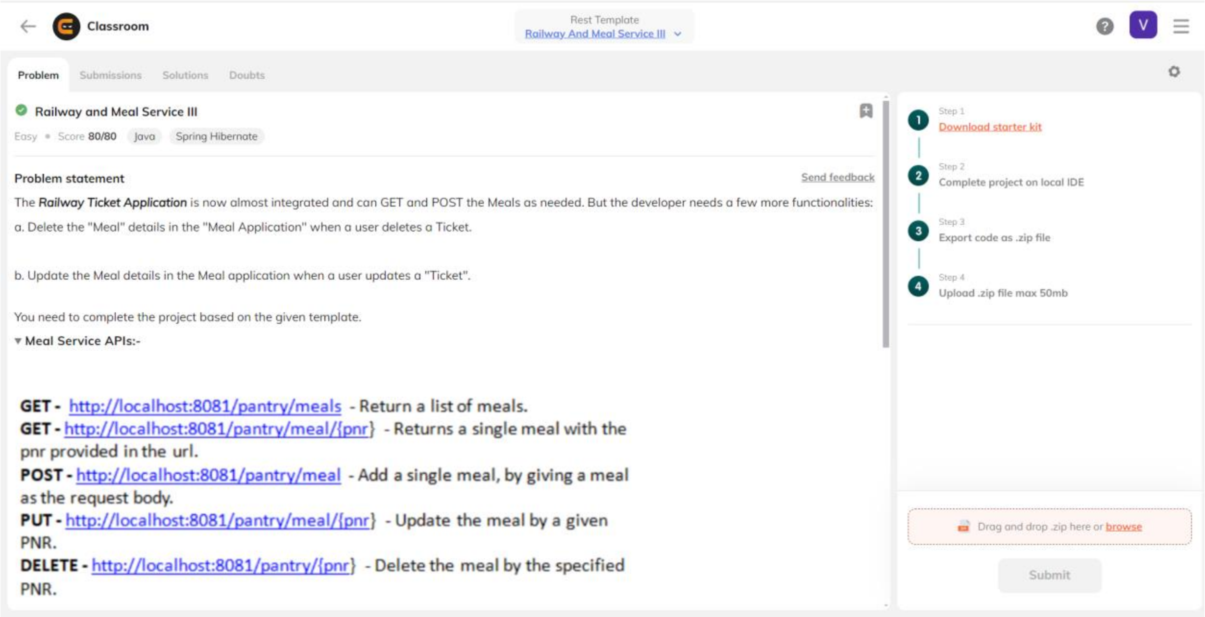
Railway and Meal Service III

Problem Statement



Railway and Meal Service III

Easy • Score 80/80 Java Spring Hibernate

Problem statement

The *Railway Ticket Application* is now almost integrated and can GET and POST the Meals as needed. But the developer needs a few more functionalities:

a. Delete the "Meal" details in the "Meal Application" when a user deletes a Ticket.

b. Update the Meal details in the Meal application when a user updates a "Ticket".

You need to complete the project based on the given template.

▾ Meal Service APIs:-

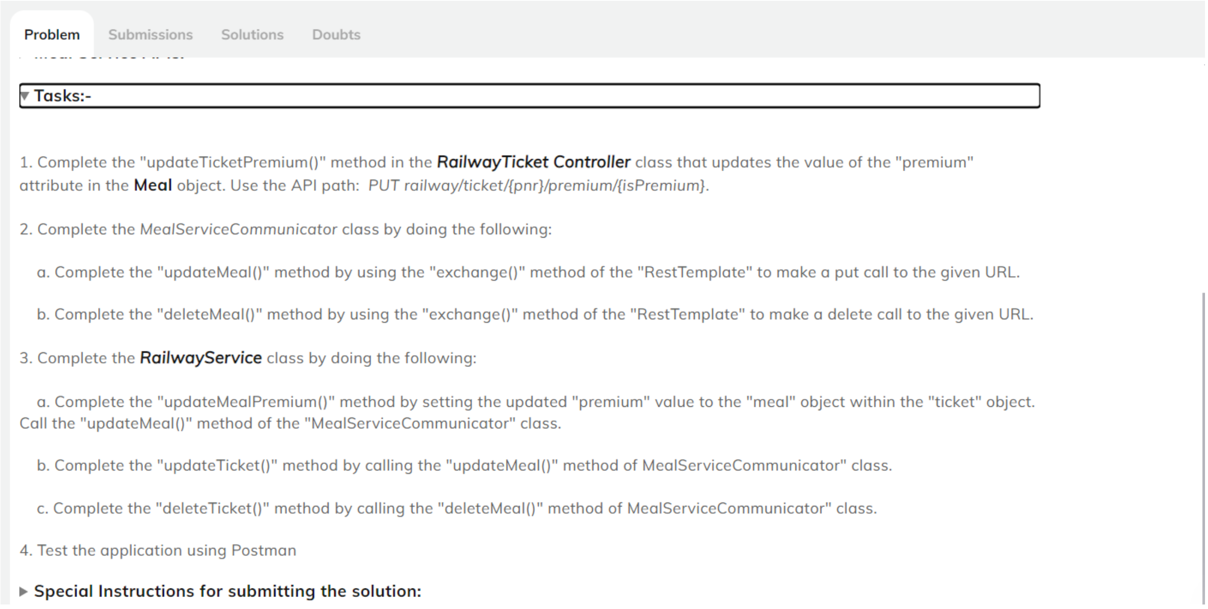
GET - http://localhost:8081/pantry/meals - Return a list of meals.
GET - http://localhost:8081/pantry/meal/{pnr} - Returns a single meal with the pnr provided in the url.
POST - http://localhost:8081/pantry/meal - Add a single meal, by giving a meal as the request body.
PUT - http://localhost:8081/pantry/meal/{pnr} - Update the meal by a given PNR.
DELETE - http://localhost:8081/pantry/{pnr} - Delete the meal by the specified PNR.

Step 1
Download starter kit

Step 2
Complete project on local IDE

Step 3
Export code as .zip file

Step 4
Upload .zip file max 50mb



▾ Tasks:-

1. Complete the "updateTicketPremium()" method in the **RailwayTicket Controller** class that updates the value of the "premium" attribute in the **Meal** object. Use the API path: *PUT railway/ticket/{pnr}/premium/{isPremium}.*

2. Complete the *MealServiceCommunicator* class by doing the following:

a. Complete the "updateMeal()" method by using the "exchange()" method of the "RestTemplate" to make a put call to the given URL.

b. Complete the "deleteMeal()" method by using the "exchange()" method of the "RestTemplate" to make a delete call to the given URL.

3. Complete the ***RailwayService*** class by doing the following:

a. Complete the "updateMealPremium()" method by setting the updated "premium" value to the "meal" object within the "ticket" object. Call the "updateMeal()" method of the "MealServiceCommunicator" class.

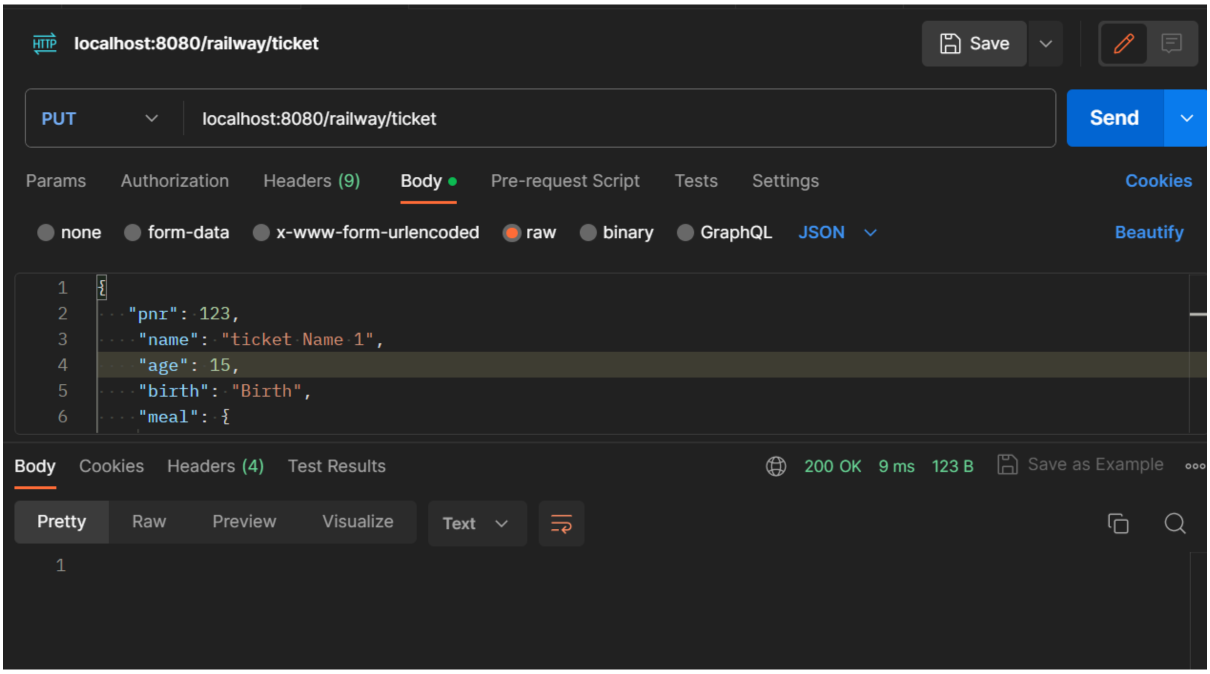
b. Complete the "updateTicket()" method by calling the "updateMeal()" method of MealServiceCommunicator" class.

c. Complete the "deleteTicket()" method by calling the "deleteMeal()" method of MealServiceCommunicator" class.

4. Test the application using Postman

▸ **Special Instructions for submitting the solution:**

Output

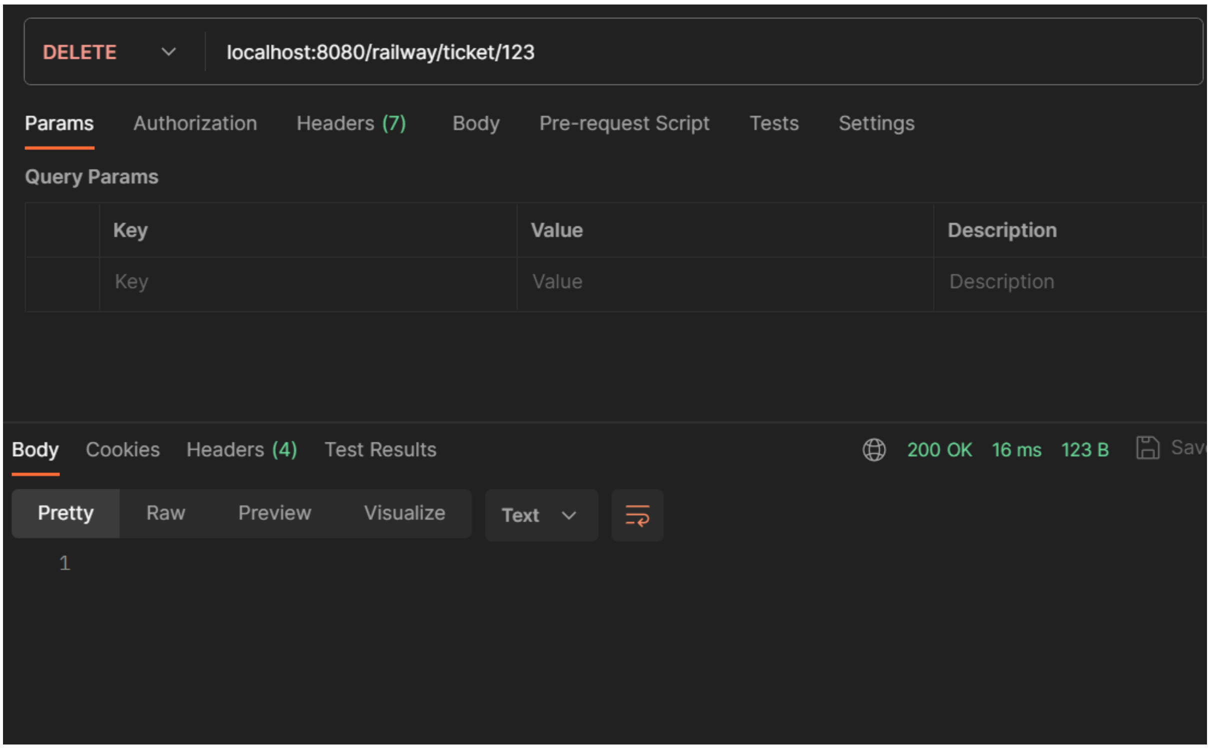


PUT localhost:8080/railway/ticket

```
{
  "pnr": 123,
   "name": "ticket Name 1",
   "age": 15,
   "birth": "Birth",
   "meal": {
```

200 OK 9 ms 123 B

```
1
```



DELETE localhost:8080/railway/ticket/123

Query Params

Key	Value	Description
Key	Value	Description

200 OK 16 ms 123 B

```
1
```