

Railway Application 3

Problem Statement:

← Classroom

REST API - PUT and DELETE
Railway Application III

?

V

Problem Submissions Solutions Doubts

Railway Application III

Easy • Score 80/80 • java Advanced java

Problem statement

The developer working on the **Railway Ticket Booking** system, now wants you to create some custom exception classes and then throw relevant error messages. He has provided you with the updated project link. You need to complete the project based on the given template.

▼ Tasks:-

1. Complete the **TicketNotFoundException** custom exception class which handles the exceptions while fetching, updating and deleting a ticket by pnr.

2. Complete the **TicketAlreadyExistsException** custom exception class which handles exceptions while adding a ticket with a given pnr.

3. Complete the service layer methods with exception implementation.

4. Test the application using Postman.

► Special Instructions for submitting the solution:-

► Note:-

▼ Sample Output:-

Send feedback

1 Step 1
Download starter kit

2 Step 2
Complete project on local IDE

3 Step 3
Export code as .zip file

4 Step 4
Upload .zip file max 50mb

Classroom

Problem statement

The developer working on the **Railway Ticket Booking** system, now wants you to create some custom exception classes and then throw relevant error messages. He has provided you with the updated project link. You need to complete the project based on the given template.

▼ Tasks:-

Output

The screenshot shows a REST client interface with the URL `localhost:8080/railway/ticket` and a `GET` method. The request is sent to `localhost:8080/railway/ticket/7`. The response status is `404 Not Found` with a response time of `17 ms` and a size of `276 B`. The response body is displayed in JSON format:

```
{
  "timestamp": "2023-08-28T13:33:04.693+00:00",
  "status": 404,
  "error": "Not Found",
  "path": "/railway/ticket/7"
}
```

The screenshot shows a REST client interface with the URL `localhost:8080/railway/ticket` and a `POST` method. The request body is a JSON object:

```
{
  "pnr": 1,
  "name": "Ticket Name",
  "age": 5,
  "birth": "Birth"
}
```

The response status is `409 Conflict` with a response time of `219 ms` and a size of `272 B`. The response body is displayed in JSON format:

```
{
  "timestamp": "2023-08-28T13:32:01.233+00:00",
  "status": 409,
  "error": "Conflict",
  "path": "/railway/ticket"
}
```