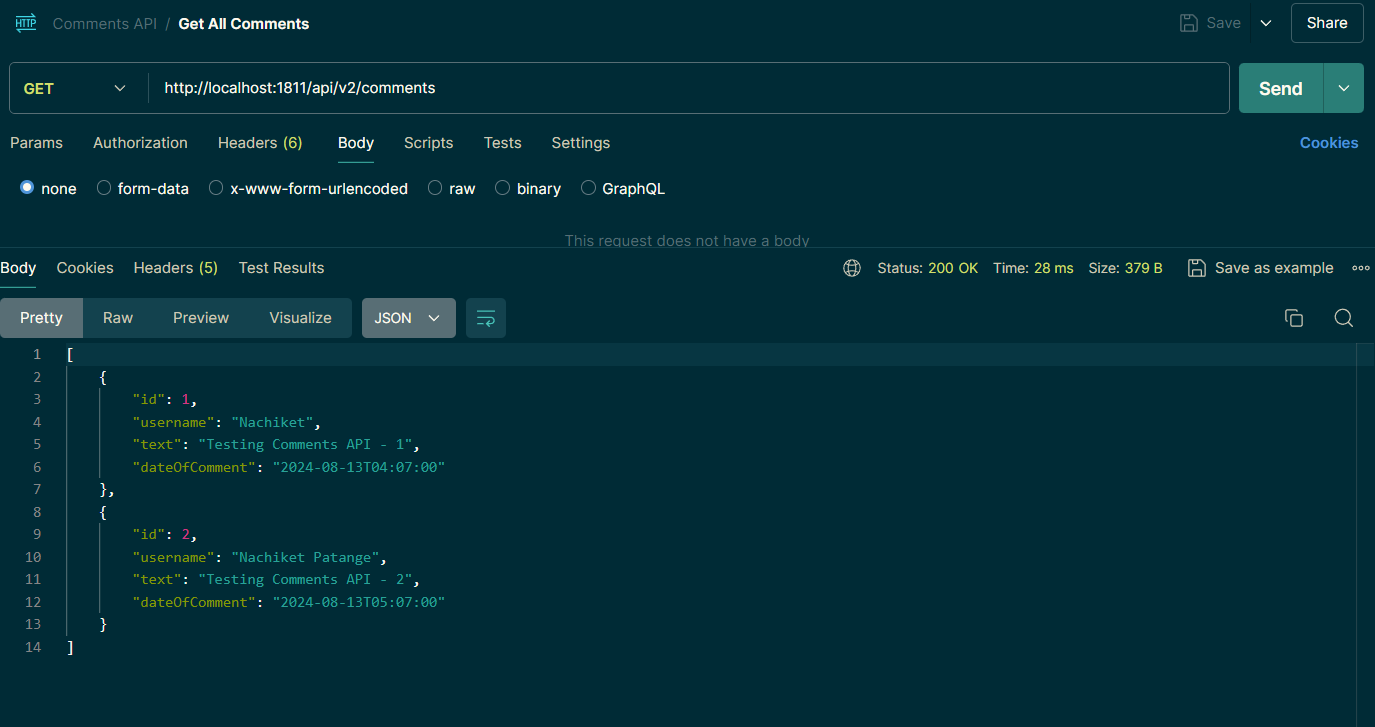
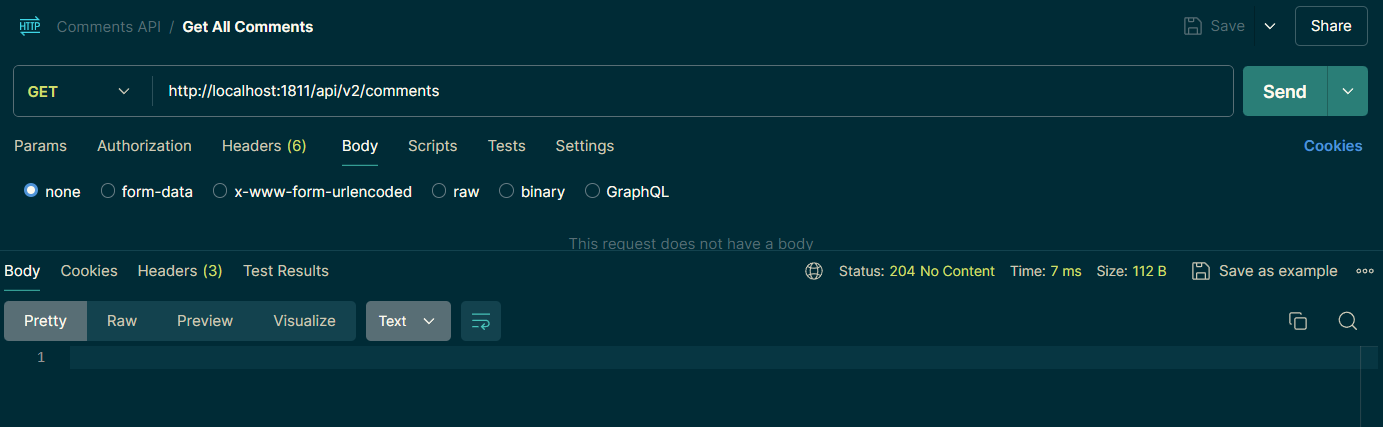
**Test Endpoints**

1. **Get All Comments:**
   1. **URL:** <http://localhost:1811/api/v2/comments>
   2. **Method:** GET
   3. **Description:** Retrieves a list of all comments
   4. **Test:**
      1. If the database has comments, the response will include a list of comments.



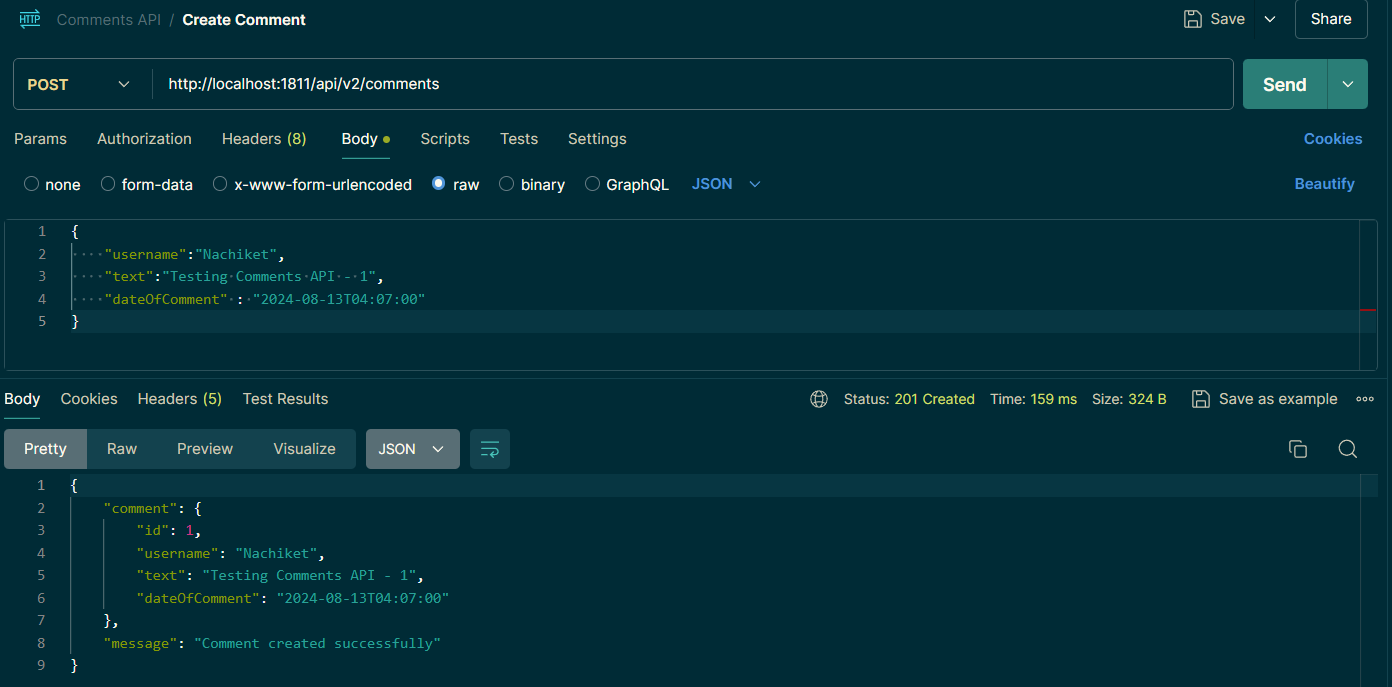
* + 1. If no comments are found, you should see a 204 No Content.



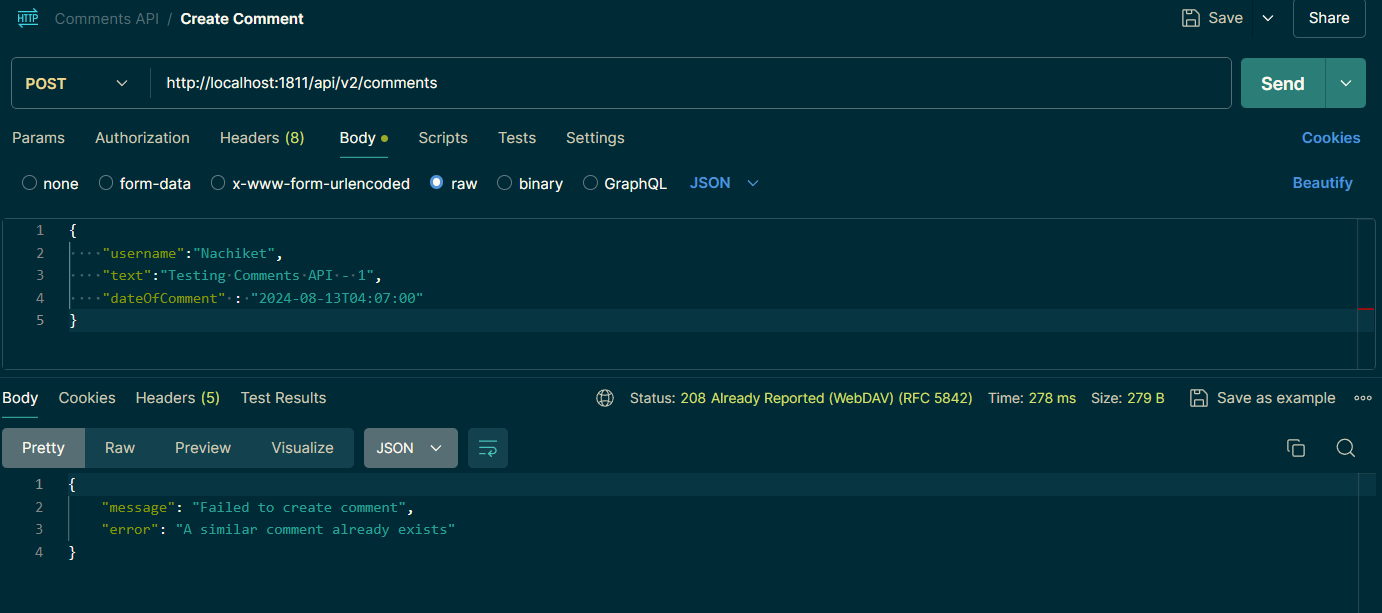
1. **Create a Comment:**
   1. **URL:** <http://localhost:1811/api/v2/comments>
   2. **Method:** POST
   3. **Description:** Creates a new comment.
   4. **Payload:**

****

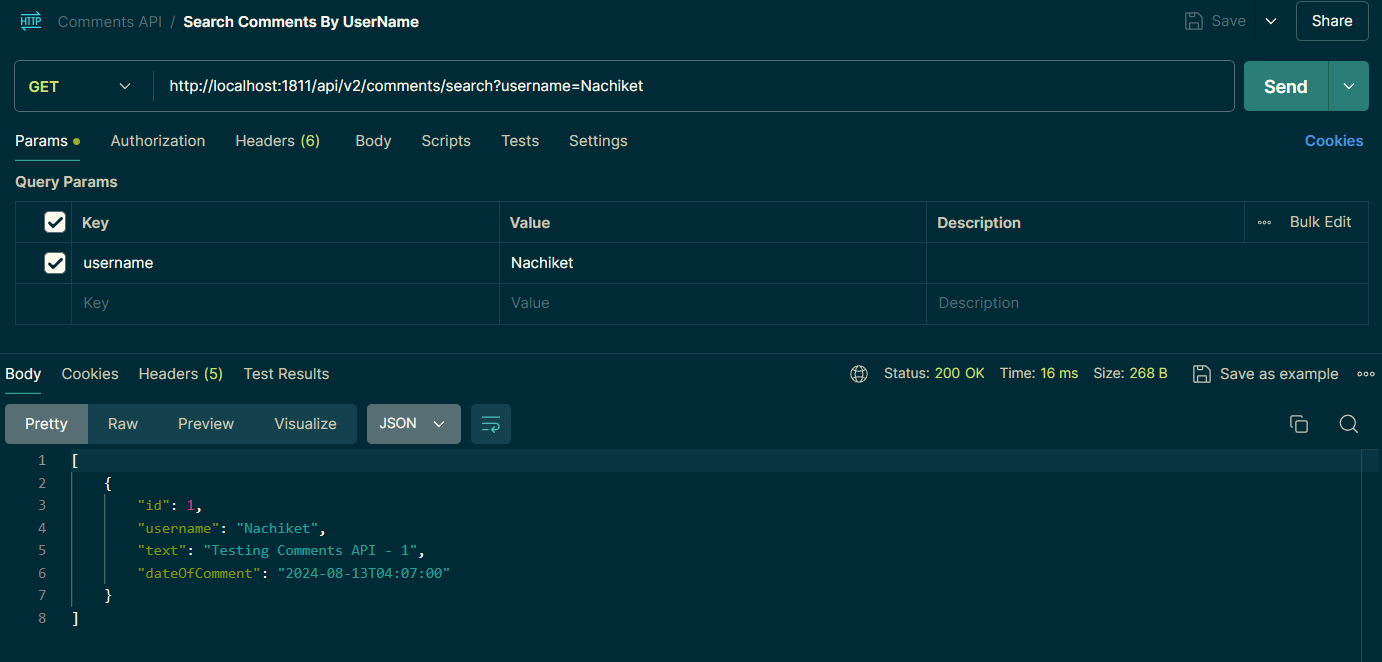
* 1. **Test:**
     1. Send a valid request body and ensure the comment is created.
     2. If successful, a 201 Created status is returned along with the created comment in the response body.



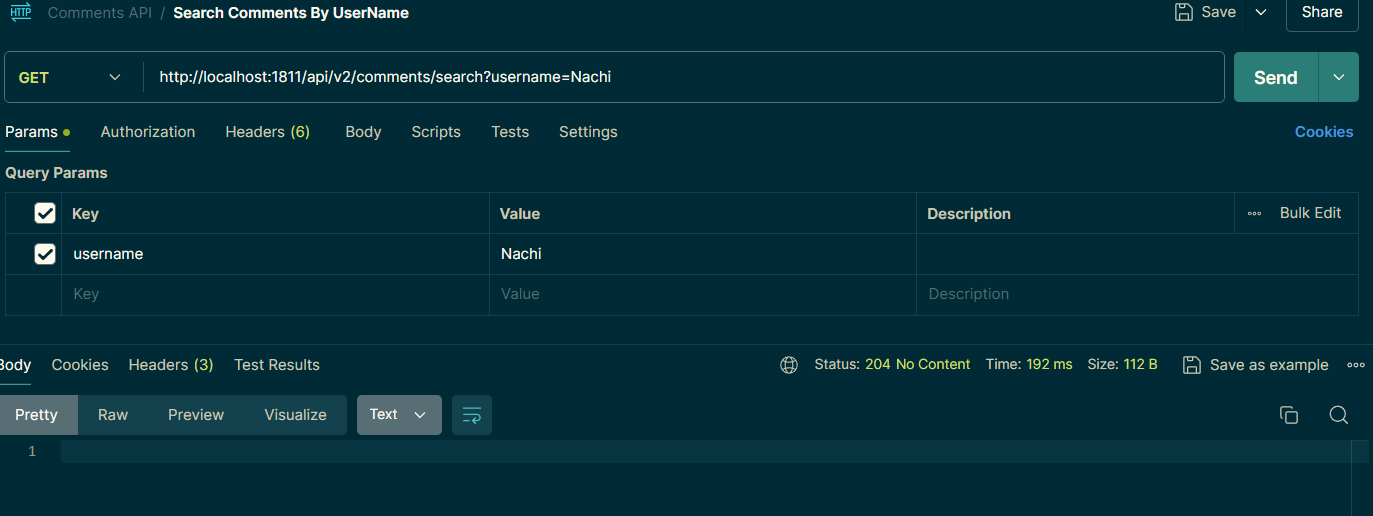
* + 1. If the request body is invalid or comment is already present, you should receive a relevant error message.



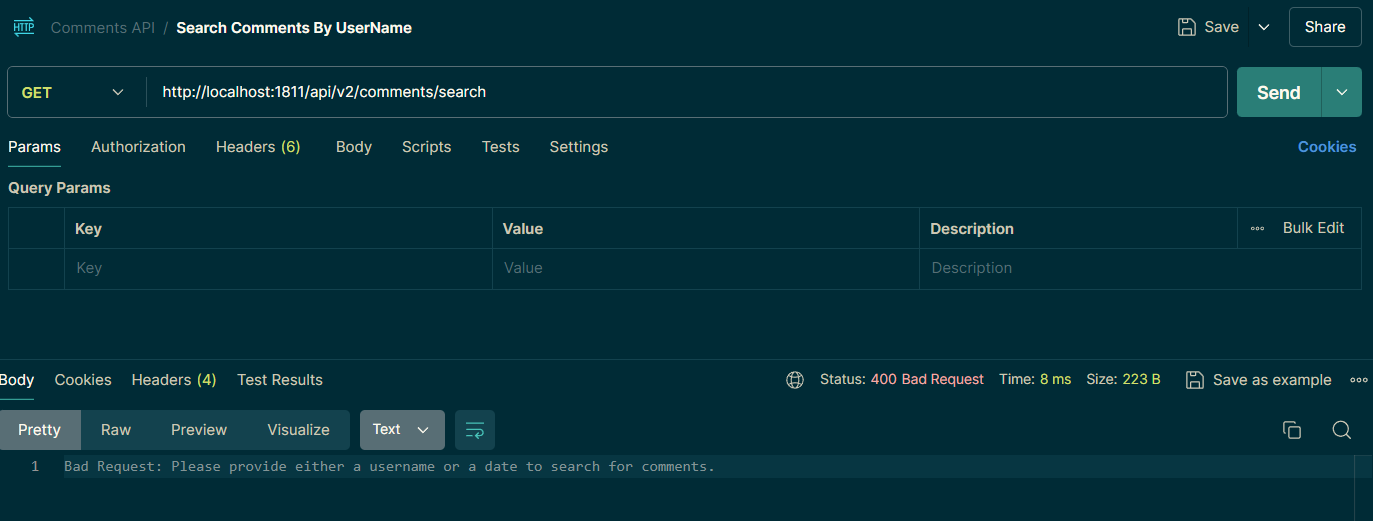
1. **Search Comments by Username or Date:**
   1. **URL:** <http://localhost:1811/api/v2/comments/search?username=Nachiket>
   2. **Method:** GET
   3. **Query Parameters:**
      1. username: Search by the username of the commenter.
      2. date: Search by the date the comment was made (format: YYYY-MM-DD).
   4. **Description:** Searches for comments based on the provided username or date.
   5. **Test:**
      1. Provide a valid username or date and ensure the appropriate comments are returned.



* + 1. If the username or date is not present in Database, you will see a 204 No Content.

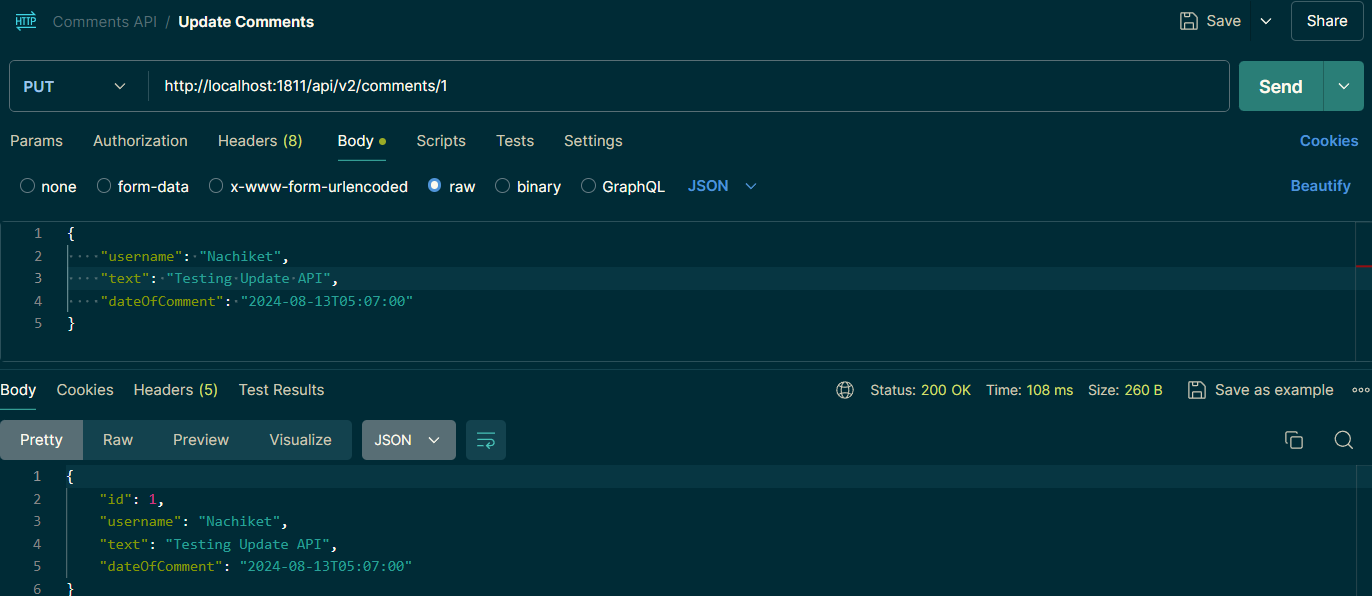


* + 1. If the username or date is not provided, an error will be thrown.

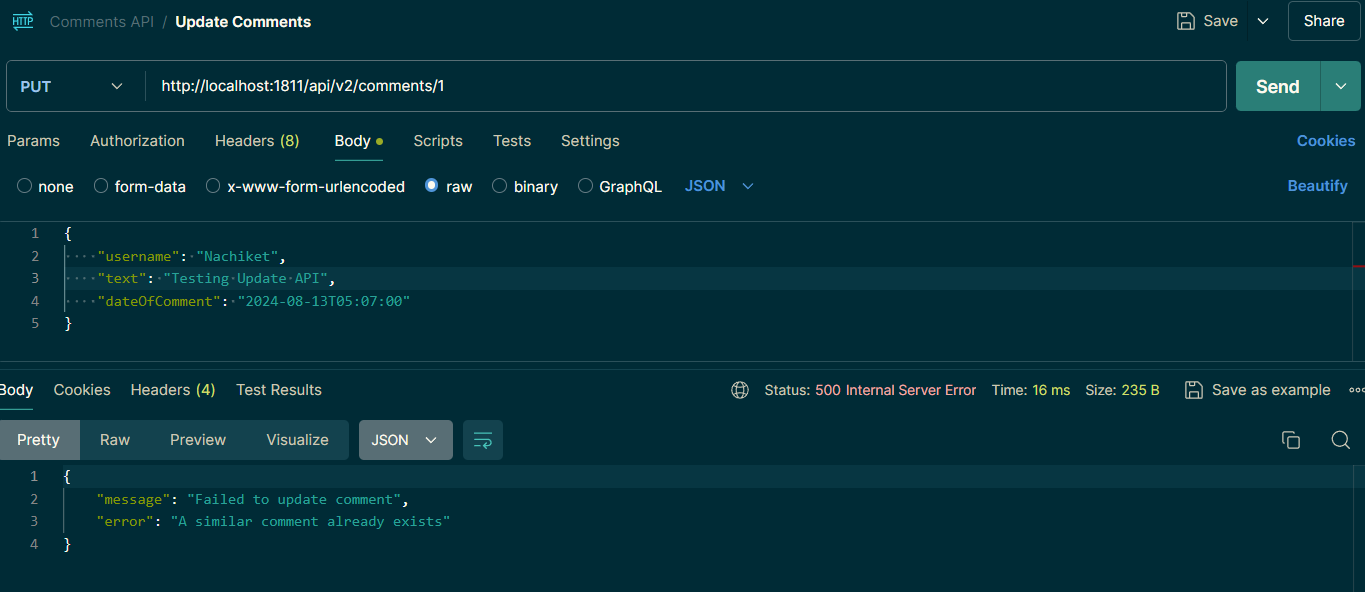


1. **Testing the Update Operation**
   1. **URL:** <http://localhost:1811/api/v2/comments/1>
   2. **Method:** PUT
   3. **Description:** Updates the comments based on the ID provided.
   4. **Test:**
      1. If a comment with the given ID is present in the database, it will be updated with the data provided in the JSON.

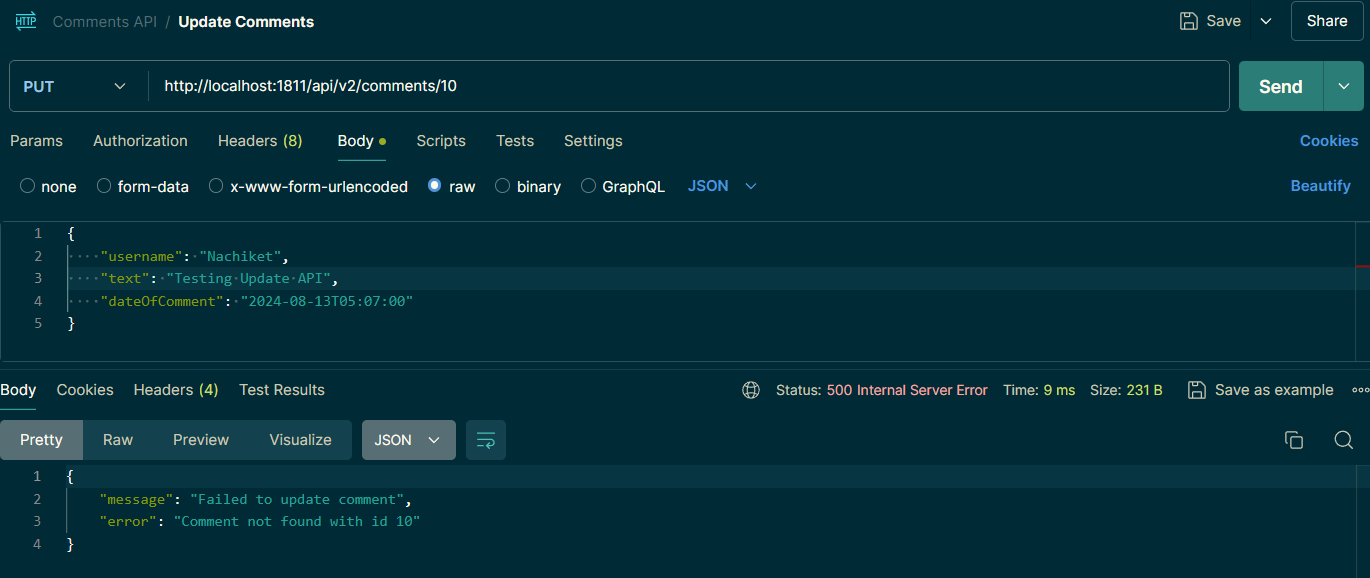




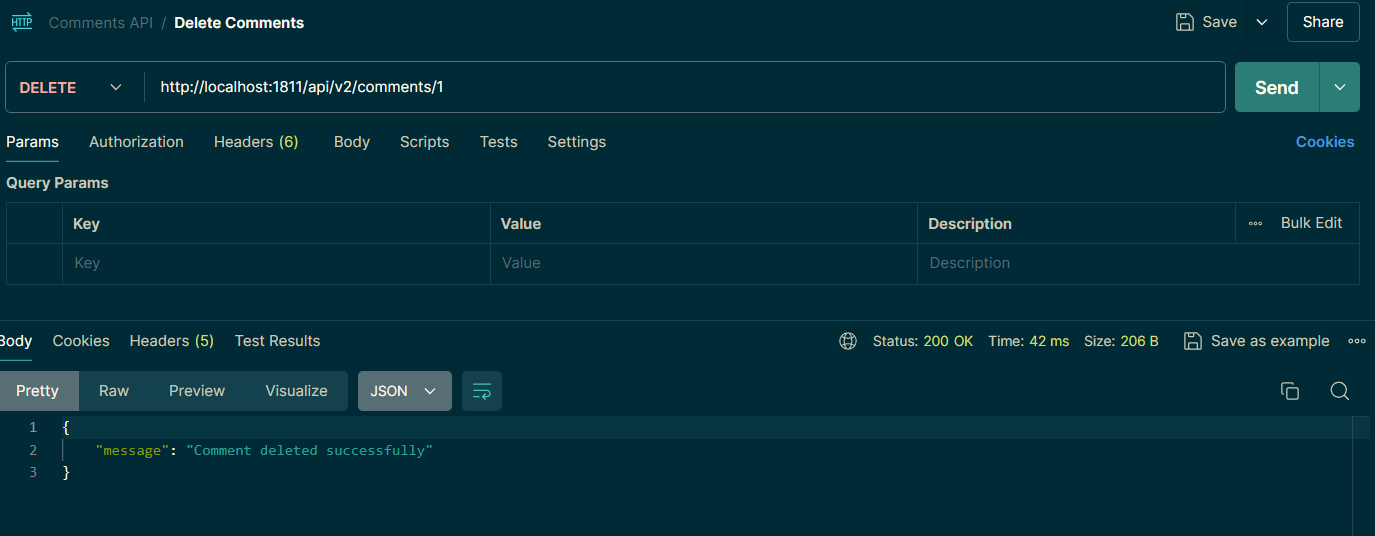
* + 1. If a comment with the given ID is present in the database and the provided JSON matches the data already in the database, the user will be informed with the message, 'A similar comment already exists.’



* + 1. If a comment is not present or if the service encounters an issue while updating, an error will be thrown.



1. **Testing the Delete Operation**
   1. **URL:** <http://localhost:1811/api/v2/comments/7>
   2. **Method:** DELETE
   3. **Description:** DELETES the comments based on the ID provided.
   4. **Test:**
      1. If a comment with the given ID is present in the database, the data will be deleted.



* + 1. If a comment with the given ID is not present, the service will throw an error.

