

USER_MANAGEMENT_API_TEST_REPORT

Echo Tree - User Management Service API Test Report

Base URL: http://localhost:5176

Service: User Management Service

Test Summary

| Category | Endpoint | Status |
|--------------------------|---|--|
| User Management | POST /api/users | ✓ PASS |
| User Management | GET /api/users/:userId | ✓ PASS |
| User Management | PATCH /api/users/:userId | ✓ PASS |
| User Management | DELETE /api/users/:userId | ✓ PASS |
| Authentication | POST /api/sessions | ✓ PASS |
| Authentication | GET /api/sessions/:sessionId | ✓ PASS |
| Authentication | DELETE /api/sessions/:sessionId | ✓ PASS |
| Sessions (Sub-resource) | GET /api/users/:userId/sessions | ✓ PASS |
| Sessions (Sub-resource) | DELETE /api/users/:userId/sessions/:sessionId | ✓ PASS |
| Penalties (Sub-resource) | POST /api/users/:userId/penalties | ! REQUIRES ADMIN |
| Penalties (Sub-resource) | GET /api/users/:userId/penalties | ✓ PASS |
| Penalties (Sub-resource) | PATCH /api/users/:userId/penalties | ! REQUIRES ADMIN |

1. User Registration

a) Use Case: Create a new user

Resource(s): users

HTTP Method: POST

Request URL: http://localhost:3000/api/users

Request Headers:

```
Content-Type: application/json
```

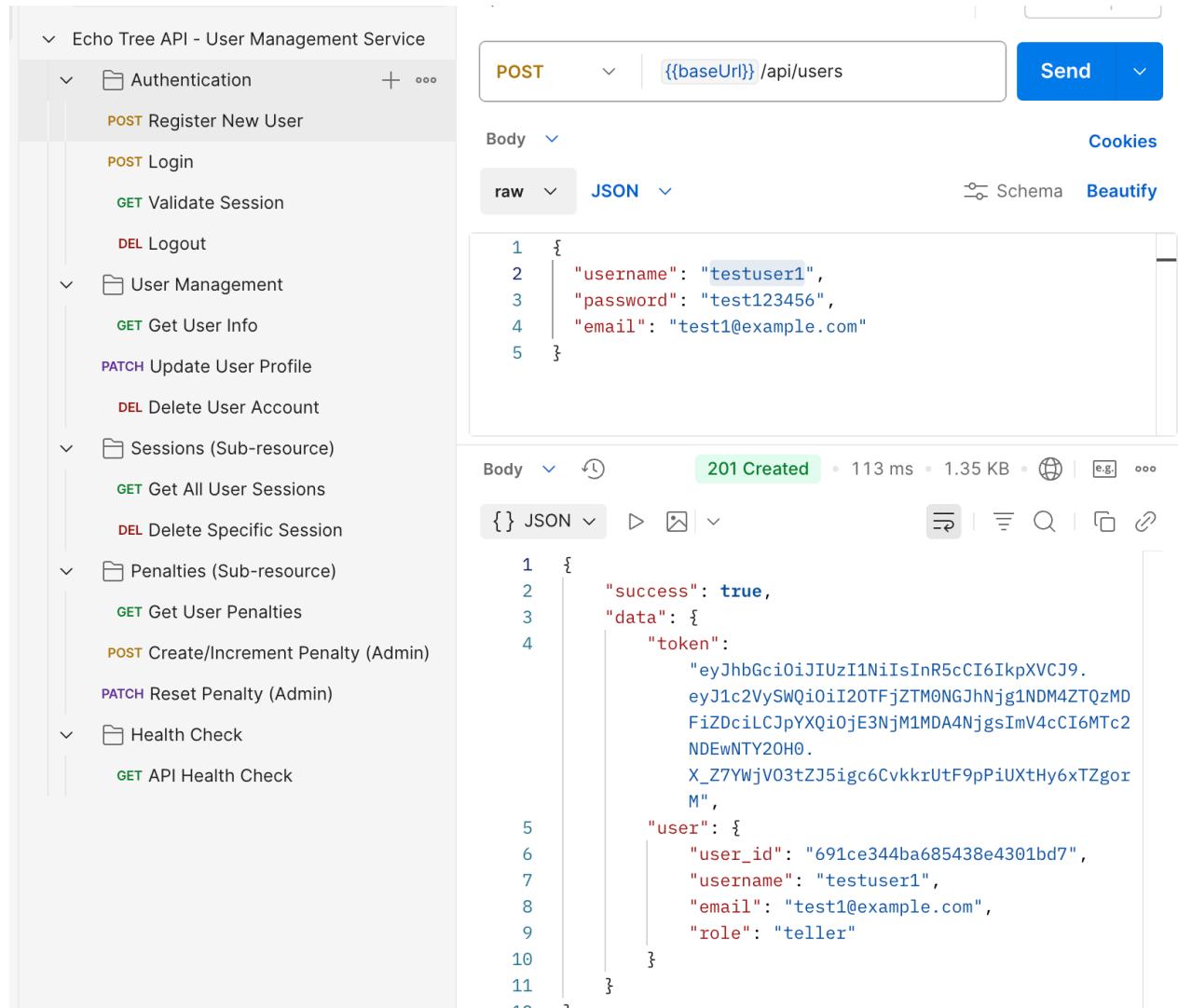
Request Body:

```
{
  "username": "testuser1",
  "password": "test123456",
  "email": "test1@example.com"
}
```

Expected Response (Status: 201 Created):

```
{
  "success": true,
  "data": {
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...",
    "user": {
      "user_id": "691ccc64ba685438e4301b86",
      "username": "testuser1",
      "email": "test1@example.com",
      "role": "teller"
    }
  }
}
```

Test Result: PASS



The screenshot shows a REST API testing interface. On the left, a tree view of the Echo Tree API - User Management Service. Under the Authentication section, there is a POST operation for Register New User. The main panel shows a POST request to {{baseUrl}} /api/users. The Body tab is selected, showing the raw JSON input:

```
1 {  
2   "username": "testuser1",  
3   "password": "test123456",  
4   "email": "test1@example.com"  
5 }
```

The response is a 201 Created status with a response time of 113 ms and a size of 1.35 KB. The response body is displayed in JSON format:

```
1 {  
2   "success": true,  
3   "data": {  
4     "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
5       eyJ1c2VySWQiOiI20TFjZTM0NGJhNjg1NDM4ZTQzMDFiZDciLCJpYXQiOjE3NjM1MDA4NjgsImV4cCI6MTc2  
6       NDEwNTY2OH0.  
7       X_7YWjV03tZJ5igc6CvkkrUtF9pPiUXtHy6xTZgor  
8       M",  
9     "user": {  
10       "user_id": "691ce344ba685438e4301bd7",  
11       "username": "testuser1",  
12       "email": "test1@example.com",  
13       "role": "teller"  
14     }  
15   }  
16 }
```

2. User Login (Create Session)

b) Use Case: User login

Resource(s): sessions

HTTP Method: POST

Request URL: <http://localhost:3000/api/sessions>

Request Headers:

Content-Type: application/json

Request Body:

```
{  
  "username": "testuser1",  
  "password": "test123456"  
}
```

Expected Response (Status: 200 OK):

```
{  
  "success": true,  
  "data": {  
    "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjI20TFjZTM0NGJhNjg1NDM4  
ZTQzMDFiZDciLCJpYXQiOjE3NjM1MDExNjQsImV4cCI6MTc2NDEwNTk2NH0.9e0pydkdWoZJh8  
8Yqe0P31zaLtgovT1ZJ6bbazoQ8_o",  
    "user": {  
      "user_id": "691ce344ba685438e4301bd7",  
      "username": "testuser1",  
      "email": "test1@example.com",  
      "role": "teller",  
      "status": "active"  
    }  
  }  
}
```

Test Result:  **PASS**

The screenshot shows the Postman interface with the following details:

- HTTP Method:** POST
- URL:** {{baseUrl}}/api/sessions
- Body:** JSON (shown in the code editor)
- Code:**

```

1 {
2     "username": "testuser1",
3     "password": "test123456"
4 }
```
- Response Status:** 200 OK
- Response Time:** 107 ms
- Response Size:** 1.72 KB
- Response Content:** (JSON object containing success: true, token: a long JWT token, and user: {user_id: 691ce344ba685438e4301bd7, username: testuser1, email: test1@example.com, role: teller, status: active})

3. Get User Information

c) Use Case: Retrieve user profile

Resource(s): users

HTTP Method: GET

Request URL: {{baseUrl}}/api/users/691ce344ba685438e4301bd7

Request Headers:

Headers (7 hidden)

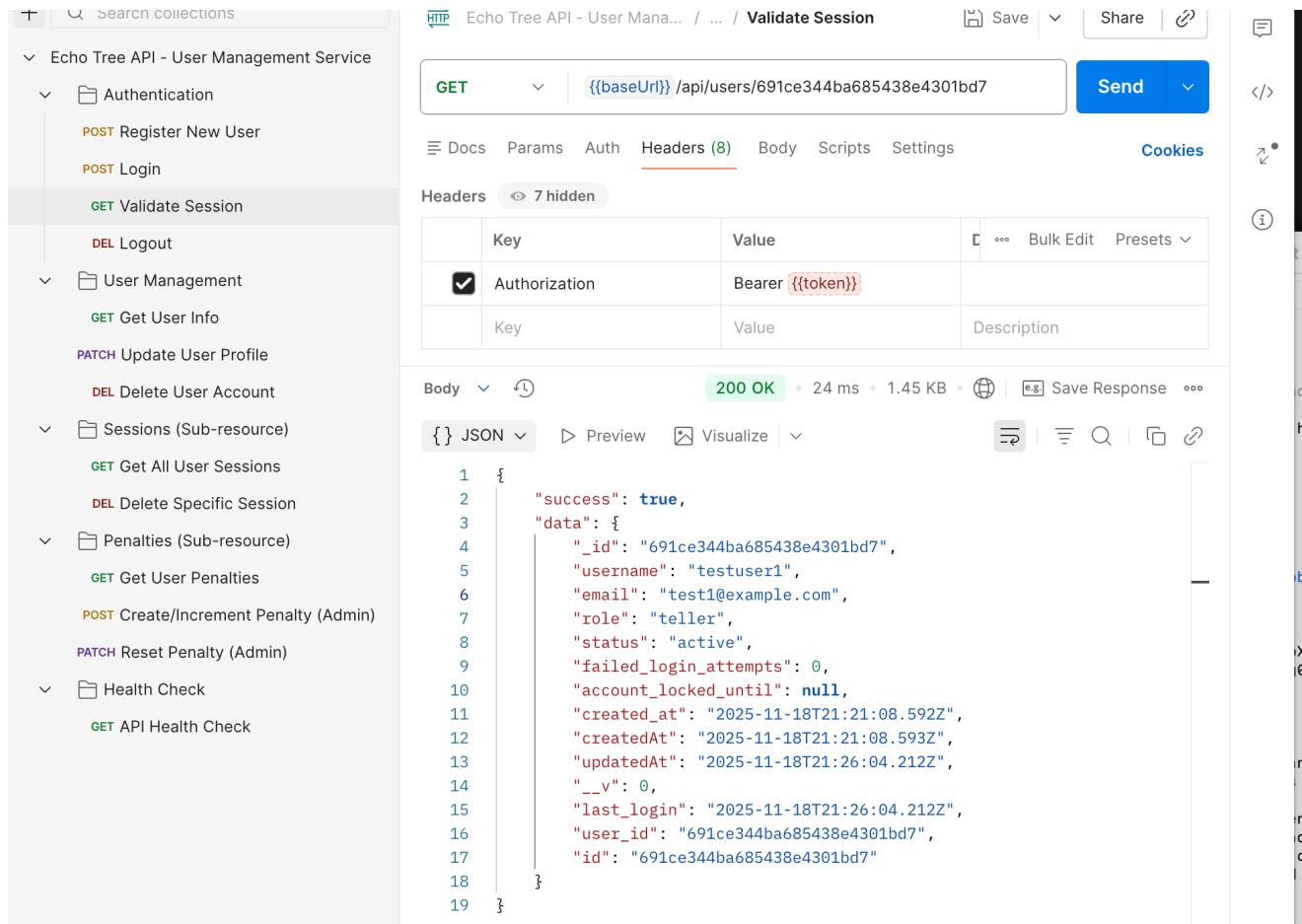
| | Key | Value | ... | Bulk Edit | Presets |
|-------------------------------------|---------------|------------------|-----|-------------|---------|
| <input checked="" type="checkbox"/> | Authorization | Bearer {{token}} | | | |
| | Key | Value | | Description | |

Request Body: None

Expected Response (Status: 200 OK):

```
{  
  
  "success": true,  
  
  "data": {  
  
    "_id": "691ce344ba685438e4301bd7",  
  
    "username": "testuser1",  
  
    "email": "test1@example.com",  
  
    "role": "teller",  
  
    "status": "active",  
  
    "failed_login_attempts": 0,  
  
    "account_locked_until": null,  
  
    "created_at": "2025-11-18T21:21:08.592Z",  
  
    "createdAt": "2025-11-18T21:21:08.593Z",  
  
    "updatedAt": "2025-11-18T21:26:04.212Z",  
  
    "__v": 0,  
  
    "last_login": "2025-11-18T21:26:04.212Z",  
  
    "user_id": "691ce344ba685438e4301bd7",  
  
    "id": "691ce344ba685438e4301bd7"  
  
  }  
  
}
```

Test Result: PASS



The screenshot shows the Postman interface with the following details:

- Collection:** Echo Tree API - User Management Service
- Request Type:** GET
- URL:** {{baseUrl}} /api/users/691ce344ba685438e4301bd7
- Headers:** Authorization: Bearer {{token}}
- Body:** 200 OK (Response status)
- Response Body (JSON):**

```
1 {  
2   "success": true,  
3   "data": {  
4     "_id": "691ce344ba685438e4301bd7",  
5     "username": "testuser1",  
6     "email": "test1@example.com",  
7     "role": "teller",  
8     "status": "active",  
9     "failed_login_attempts": 0,  
10    "account_locked_until": null,  
11    "created_at": "2025-11-18T21:21:08.592Z",  
12    "createdAt": "2025-11-18T21:21:08.593Z",  
13    "updatedAt": "2025-11-18T21:26:04.212Z",  
14    "__v": 0,  
15    "last_login": "2025-11-18T21:26:04.212Z",  
16    "user_id": "691ce344ba685438e4301bd7",  
17    "id": "691ce344ba685438e4301bd7"  
18  }  
19 }
```

4. Update User Profile

d) Use Case: Update user email

Resource(s): users

HTTP Method: PATCH

Request URL: <http://localhost:3000/api/users/691ce344ba685438e4301bd7>

Request Headers:

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...  
Content-Type: application/json
```

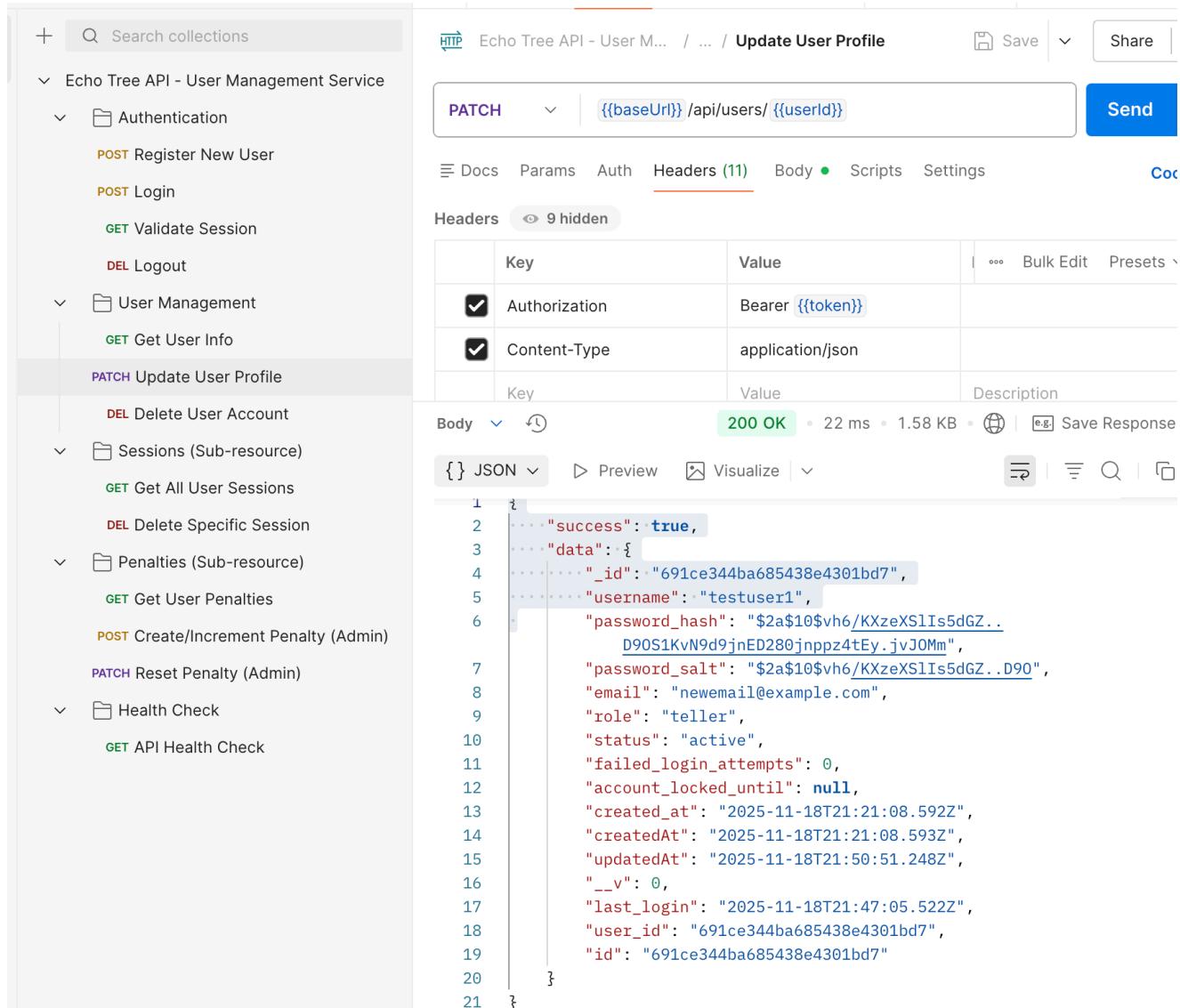
Request Body:

```
{  
  "email": "newemail@example.com"  
}
```

Expected Response (Status: 200 OK):

```
{  
  
  "success": true,  
  
  "data": {  
  
    "_id": "691ce344ba685438e4301bd7",  
  
    "username": "testuser1",  
  
    "password_hash":  
      "$2a$10$vh6/KXzeXS1Is5dGZ..D90S1KvN9d9jnED280jnppz4tEy.jvJ0Mm",  
  
    "password_salt": "$2a$10$vh6/KXzeXS1Is5dGZ..D90",  
  
    "email": "newemail@example.com",  
  
    "role": "teller",  
  
    "status": "active",  
  
    "failed_login_attempts": 0,  
  
    "account_locked_until": null,  
  
    "created_at": "2025-11-18T21:21:08.592Z",  
  
    "createdAt": "2025-11-18T21:21:08.593Z",  
  
    "updatedAt": "2025-11-18T21:50:51.248Z",  
  
    "__v": 0,  
  
    "last_login": "2025-11-18T21:47:05.522Z",  
  
    "user_id": "691ce344ba685438e4301bd7",  
  
    "id": "691ce344ba685438e4301bd7"  
  
  }  
  
}
```

Test Result: PASS



The screenshot shows the Echo Tree API - User Management Service in Postman. The left sidebar lists various endpoints under the "Echo Tree API - User Management Service" collection. The "PATCH Update User Profile" endpoint is selected. The main interface shows a PATCH request to `{{baseUrl}}/api/users/{{userId}}`. The Headers tab is active, showing two checked headers: "Authorization: Bearer {{token}}" and "Content-Type: application/json". The Body tab shows a JSON response with status 200 OK, a response time of 22 ms, and a size of 1.58 KB. The response body is a JSON object with fields like success, data (containing user details), and id.

```
1  ...
2  ...
3  ...
4  ...
5  ...
6  ...
7  ...
8  ...
9  ...
10  ...
11  ...
12  ...
13  ...
14  ...
15  ...
16  ...
17  ...
18  ...
19  ...
20  ...
21  ...
```

5. Get User Sessions (Sub-resource)

e) Use Case: Retrieve all active sessions for a user

Resource(s): users/sessions

HTTP Method: GET

Request URL: `{{baseUrl}}/api/users/691ce344ba685438e4301bd7/sessions`

Request Headers:

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Query Parameters:

- `limit` : Number of results (default: 20)
- `offset` : Pagination offset (default: 0)

Request Body: None

Expected Response (Status: 200 OK):

```
{  
  "success": true,  
  "data": [  
    {  
      "_id": "691ce959ba685438e4301bf1",  
      "user_id": "691ce344ba685438e4301bd7",  
      "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOjI20TFjZTM0NGJhNjg1NDM4  
ZTQzMDFiZDciLCJpYXQiOjE3NjM1MDI0MjUsImV4cCI6MTc2NDEwNzIyNX0.bf0h-  
0Mq_FhlK7o31EgMbHoCEXGnqbeYv04vGYh820o",  
      "expires_at": "2025-11-25T21:47:05.522Z",  
      "ip_address": "::1",  
      "user_agent": "PostmanRuntime/7.49.1",  
      "created_at": "2025-11-18T21:47:05.523Z",  
      "createdAt": "2025-11-18T21:47:05.523Z",  
      "updatedAt": "2025-11-18T21:47:05.523Z",  
      "__v": 0  
ZTQzMDFiZDciLCJpYXQiOjE3NjM1MDIzNDQsImV4cCI6MTc2NDEwNzE0NH0.kkLureDkTEpKOC  
zTERQC8ZFyA9gMIWQrSsMDy08oUwI",  

```

```
"ip_address": "::1",
"user_agent": "PostmanRuntime/7.49.1",
"created_at": "2025-11-18T21:45:44.288Z",
"createdAt": "2025-11-18T21:45:44.289Z",
"updatedAt": "2025-11-18T21:45:44.289Z",
"__v": 0
},
{
"_id": "691ce46cba685438e4301bdc",
"user_id": "691ce344ba685438e4301bd7",
"token":
"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI20TFjZTM0NGJhNjg1NDM4
ZTQzMDFiZDciLCJpYXQiOjE3NjM1MDExNjQsImV4cCI6MTc2NDEwNTk2NH0.9e0pydkdWoZJh8
8YqeOP31zaLtgovT1ZJ6bbazoQ8_o",
"expires_at": "2025-11-25T21:26:04.215Z",
"ip_address": "::1",
"user_agent": "PostmanRuntime/7.49.1",
"created_at": "2025-11-18T21:26:04.215Z",
"createdAt": "2025-11-18T21:26:04.216Z",
"updatedAt": "2025-11-18T21:26:04.216Z",
"__v": 0
},
],
"pagination": {
"total": 3,
"limit": 20,
```

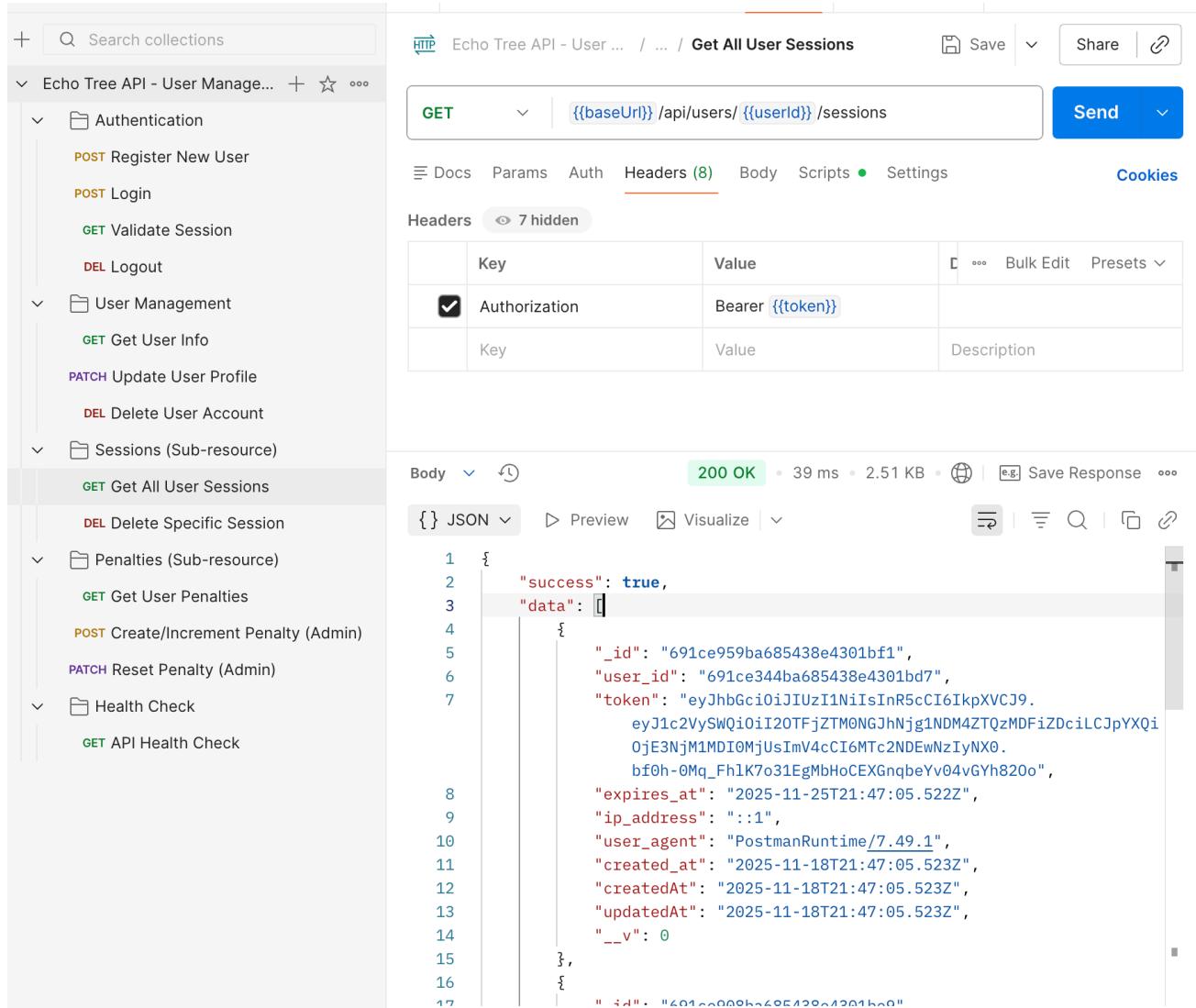
```
"offset": 0
```

```
}
```

```
}
```

Test Result: PASS

Notes:



The screenshot shows the Postman interface with the following details:

- Request URL:** {{baseUrl}}/api/users/{{userId}}/sessions
- Headers:** Authorization: Bearer {{token}}
- Response:** 200 OK (39 ms, 2.51 KB)
 - Body (JSON):**

```
1 {  
2   "success": true,  
3   "data": [  
4     {  
5       "_id": "691ce959ba685438e4301bf1",  
6       "user_id": "691ce344ba685438e4301bd7",  
7       "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
8         eyJ1c2VySWQiOii20TFjZTMONGJhNjg1NDM4ZTQzMDFiZDciLCJpYXQi  
9         OjE3NjM1MDI0MjUsImV4cCI6MTc2NDEwNzIyNX0.  
10        bf0h-0Mq_Fh1K7o31EgMbHoCEXGnqbeYv04vGYh820o",  
11        "expires_at": "2025-11-25T21:47:05.522Z",  
12        "ip_address": "::1",  
13        "user_agent": "PostmanRuntime/7.49.1",  
14        "created_at": "2025-11-18T21:47:05.523Z",  
15        "createdAt": "2025-11-18T21:47:05.523Z",  
16        "updatedAt": "2025-11-18T21:47:05.523Z",  
17        "__v": 0  
18     },  
19     {  
20       "id": "691ce959ba685438e4301bf1",  
21       "user_id": "691ce344ba685438e4301bd7",  
22       "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.  
23         eyJ1c2VySWQiOii20TFjZTMONGJhNjg1NDM4ZTQzMDFiZDciLCJpYXQi  
24         OjE3NjM1MDI0MjUsImV4cCI6MTc2NDEwNzIyNX0.  
25         bf0h-0Mq_Fh1K7o31EgMbHoCEXGnqbeYv04vGYh820o",  
26         "expires_at": "2025-11-25T21:47:05.522Z",  
27         "ip_address": "::1",  
28         "user_agent": "PostmanRuntime/7.49.1",  
29         "created_at": "2025-11-18T21:47:05.523Z",  
30         "createdAt": "2025-11-18T21:47:05.523Z",  
31         "updatedAt": "2025-11-18T21:47:05.523Z",  
32         "__v": 0  
33     }  
34   ]  
35 }
```

6. Validate Session

f) Use Case: Check if a session is valid

Resource(s): sessions

HTTP Method: GET

Request URL: {{baseUrl}}/api/users/691ce344ba685438e4301bd7

Request Headers:

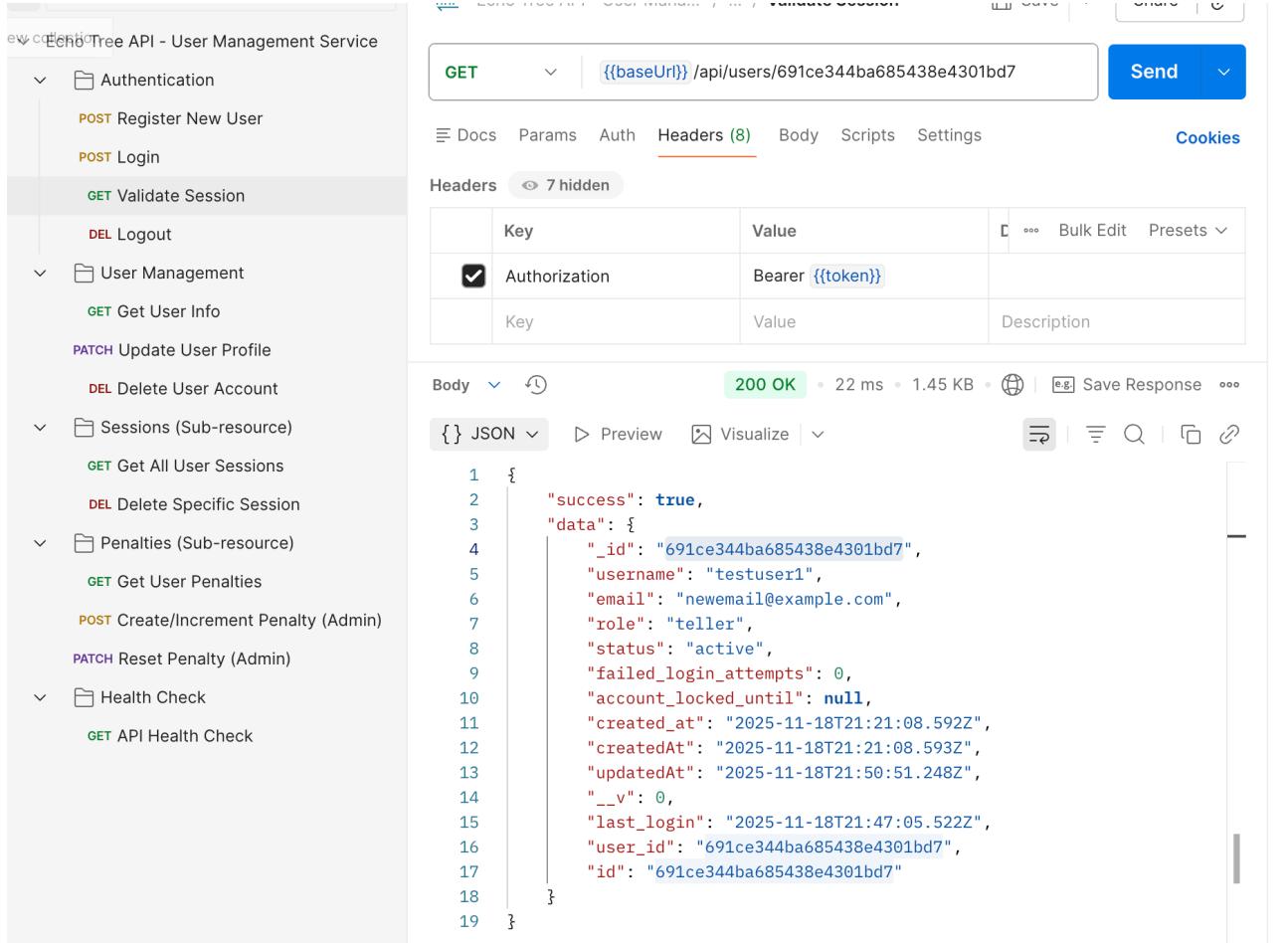
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Request Body: None

Expected Response (Status: 200 OK):

```
{  
  "success": true,  
  "data": {  
    "_id": "691ce344ba685438e4301bd7",  
    "username": "testuser1",  
    "email": "newemail@example.com",  
    "role": "teller",  
    "status": "active",  
    "failed_login_attempts": 0,  
    "account_locked_until": null,  
    "created_at": "2025-11-18T21:21:08.592Z",  
    "createdAt": "2025-11-18T21:21:08.593Z",  
    "updatedAt": "2025-11-18T21:50:51.248Z",  
    "__v": 0,  
    "last_login": "2025-11-18T21:47:05.522Z",  
    "user_id": "691ce344ba685438e4301bd7",  
    "id": "691ce344ba685438e4301bd7"  
  }  
}
```

Test Result: PASS



The screenshot shows the EchoTree API - User Management Service interface. On the left, there's a sidebar with a tree view of API endpoints. The 'Sessions (Sub-resource)' section is expanded, showing 'Get All User Sessions' and 'Delete Specific Session'. The 'Logout' endpoint under 'Authentication' is highlighted. On the right, a detailed request configuration window is open. It shows a GET request to {{baseUrl}}/api/users/691ce344ba685438e4301bd7. The 'Headers' tab is selected, containing an Authorization header with the value 'Bearer {{token}}'. The 'Body' tab shows a JSON response with status 200 OK, 22 ms, 1.45 KB, and a sample response object:

```
1 {
2   "success": true,
3   "data": {
4     "_id": "691ce344ba685438e4301bd7",
5     "username": "testuser1",
6     "email": "newemail@example.com",
7     "role": "teller",
8     "status": "active",
9     "failed_login_attempts": 0,
10    "account_locked_until": null,
11    "created_at": "2025-11-18T21:21:08.592Z",
12    "createdAt": "2025-11-18T21:21:08.593Z",
13    "updatedAt": "2025-11-18T21:50:51.248Z",
14    "__v": 0,
15    "last_login": "2025-11-18T21:47:05.522Z",
16    "user_id": "691ce344ba685438e4301bd7",
17    "id": "691ce344ba685438e4301bd7"
18  }
19 }
```

7. Logout (Delete Session)

g) Use Case: Logout user by deleting session

Resource(s): sessions

HTTP Method: DELETE

Request URL: {{baseUrl}}/api/sessions/691ce344ba685438e4301bd7

Request Headers:

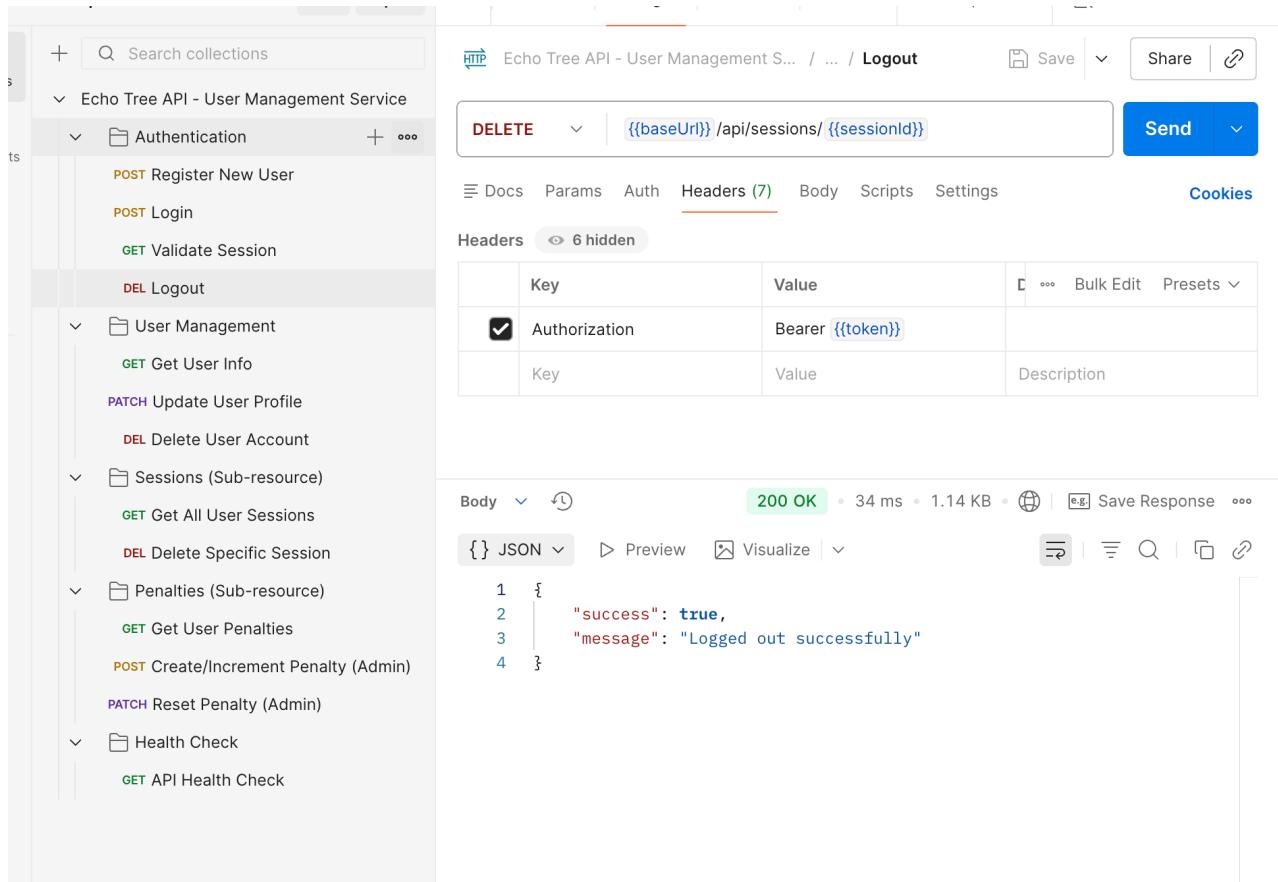
```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Request Body: None

Expected Response (Status: 200 OK):

```
{
  "success": true,
  "message": "Logged out successfully"
}
```

Test Result: PASS



The screenshot shows the Postman interface with the following details:

- Collection:** Echo Tree API - User Management Service
- Request Type:** DELETE
- URL:** {{baseUrl}}/api/sessions/{{sessionId}}
- Headers:** Authorization: Bearer {{token}}
- Response Status:** 200 OK
- Response Body (JSON):**

```
1 {  
2   "success": true,  
3   "message": "Logged out successfully"  
4 }
```

8. Delete Specific User Session (Sub-resource)

h) Use Case: Delete a specific session for a user

Resource(s): users/sessions

HTTP Method: DELETE

Request URL:

{{baseUrl}}/api/users/691ce344ba685438e4301bd7/sessions/691ce908ba685438e4301be9

Request Headers:

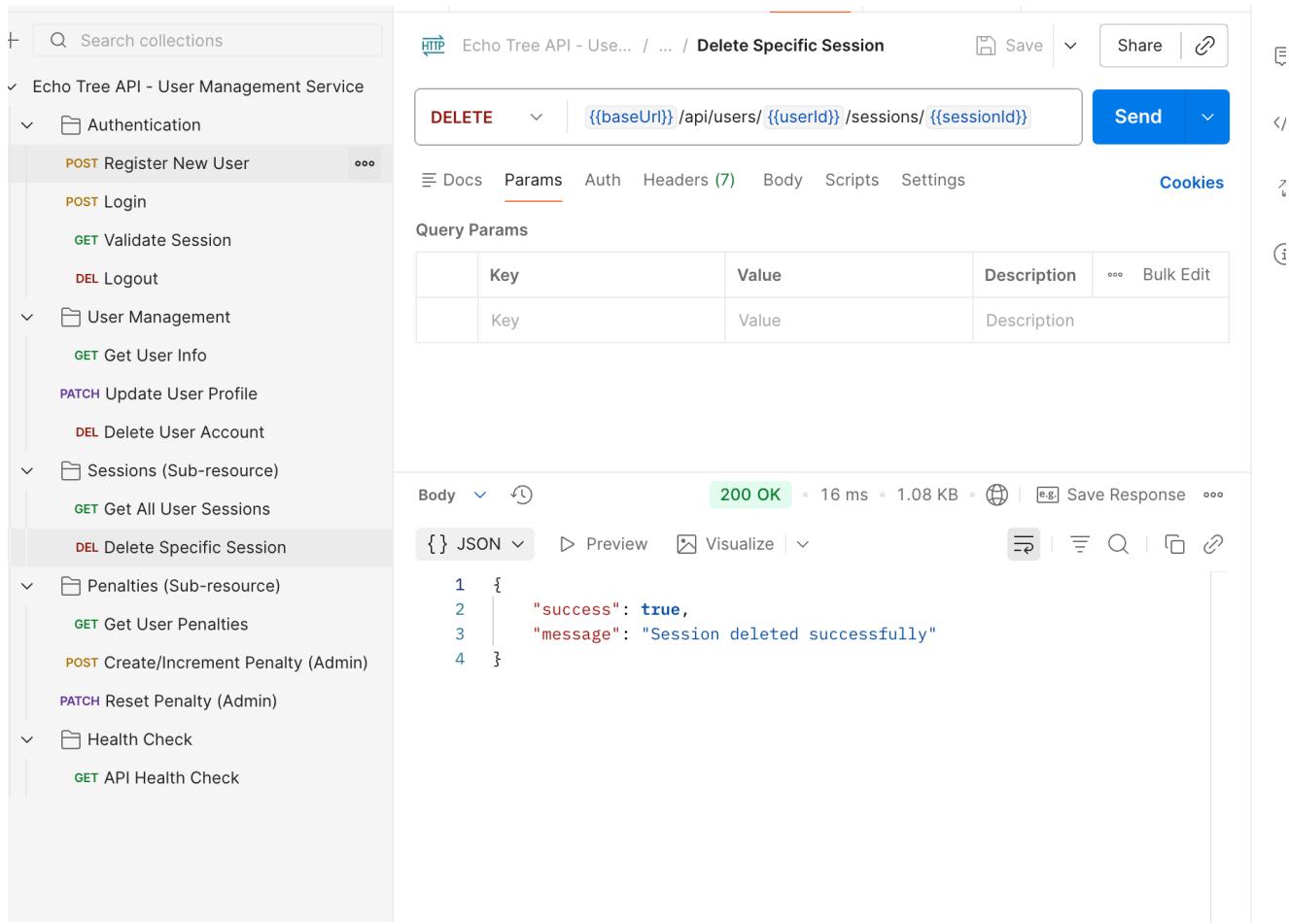
```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Request Body: None

Expected Response (Status: 200 OK):

```
{  
  "success": true,  
  "message": "Session deleted successfully"  
}
```

Test Result: PASS



The screenshot shows the Echo Tree API - User Management Service in Postman. The left sidebar lists various endpoints under the 'Echo Tree API - User Management Service' collection. The 'Sessions (Sub-resource)' section is expanded, and the 'Delete Specific Session' endpoint is selected. The main interface shows a 'DELETE' request with the URL `{baseUrl}/api/users/{userId}/sessions/{sessionId}`. The 'Params' tab is active, showing a table for 'Query Params' with one row: 'Key' (Value) and 'Value' (Description). Below the request area, the response is displayed as a '200 OK' status with a response time of 16 ms and a size of 1.08 KB. The response body is JSON, showing a success message:

```
1 {  
2   "success": true,  
3   "message": "Session deleted successfully"  
4 }
```

9. Delete User Account

i) Use Case: Delete user account permanently

Resource(s): users

HTTP Method: DELETE

Request URL: `{baseUrl}/api/users/691ce344ba685438e4301bd7`

Request Headers:

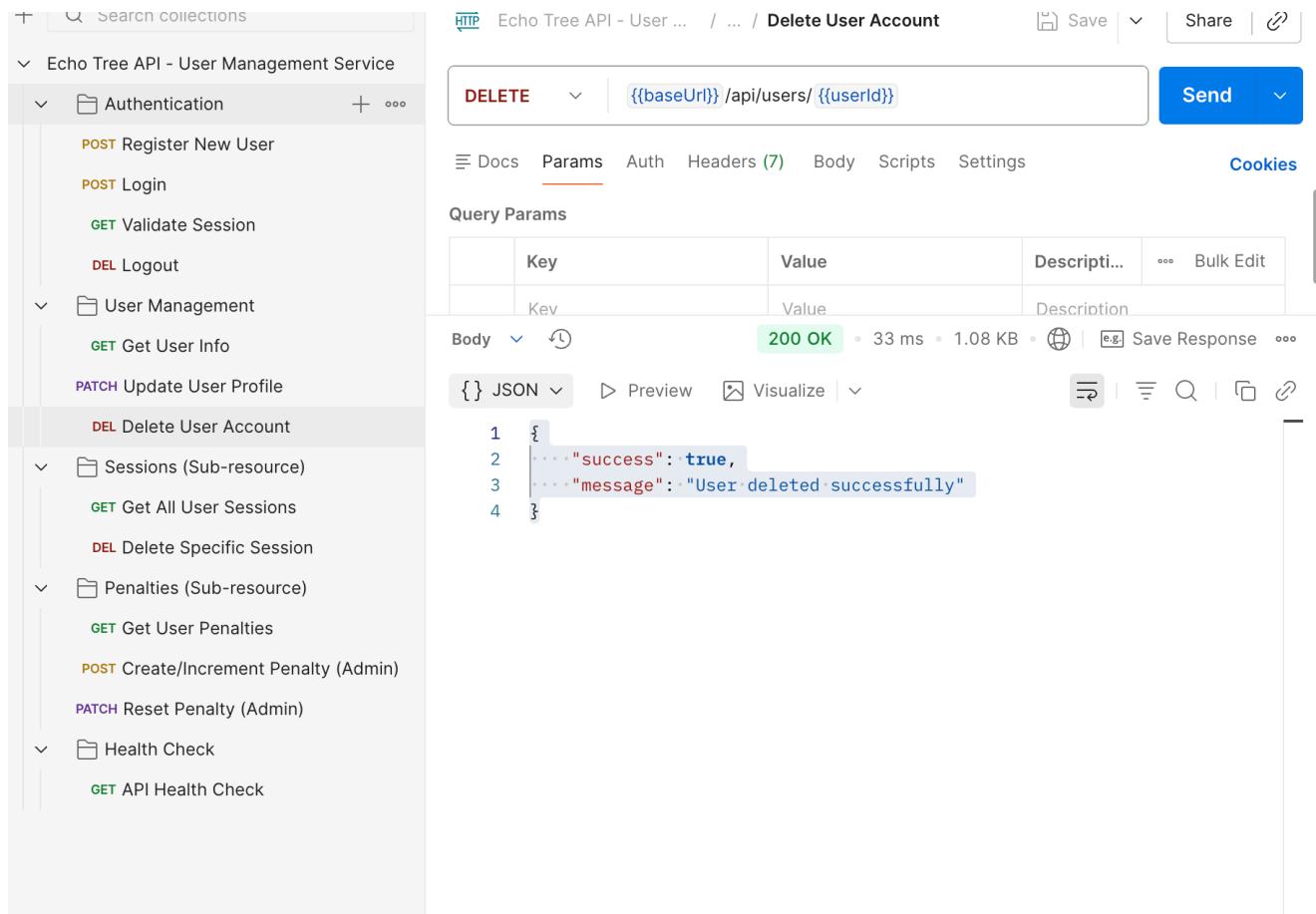
```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Request Body: None

Expected Response (Status: 200 OK):

```
{  
  "success": true,  
  "message": "User deleted successfully"  
}
```

Test Result: PASS



The screenshot shows the Echo Tree API - User Management Service in Postman. The left sidebar lists various API endpoints under categories like Authentication, User Management, Sessions, Penalties, and Health Check. The main area shows a DELETE request to `{{baseUrl}}/api/users/{{userId}}`. The response status is 200 OK, with a response time of 33 ms and a size of 1.08 KB. The response body is a JSON object:

```
1 {  
2   "success": true,  
3   "message": "User deleted successfully"  
4 }
```

10. Get User Penalties (Sub-resource)

j) Use Case: Retrieve healer penalty information

Resource(s): users/penalties

HTTP Method: GET

Request URL: `{{baseUrl}}/api/users/{{userId}}/penalties`

Request Headers:

```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

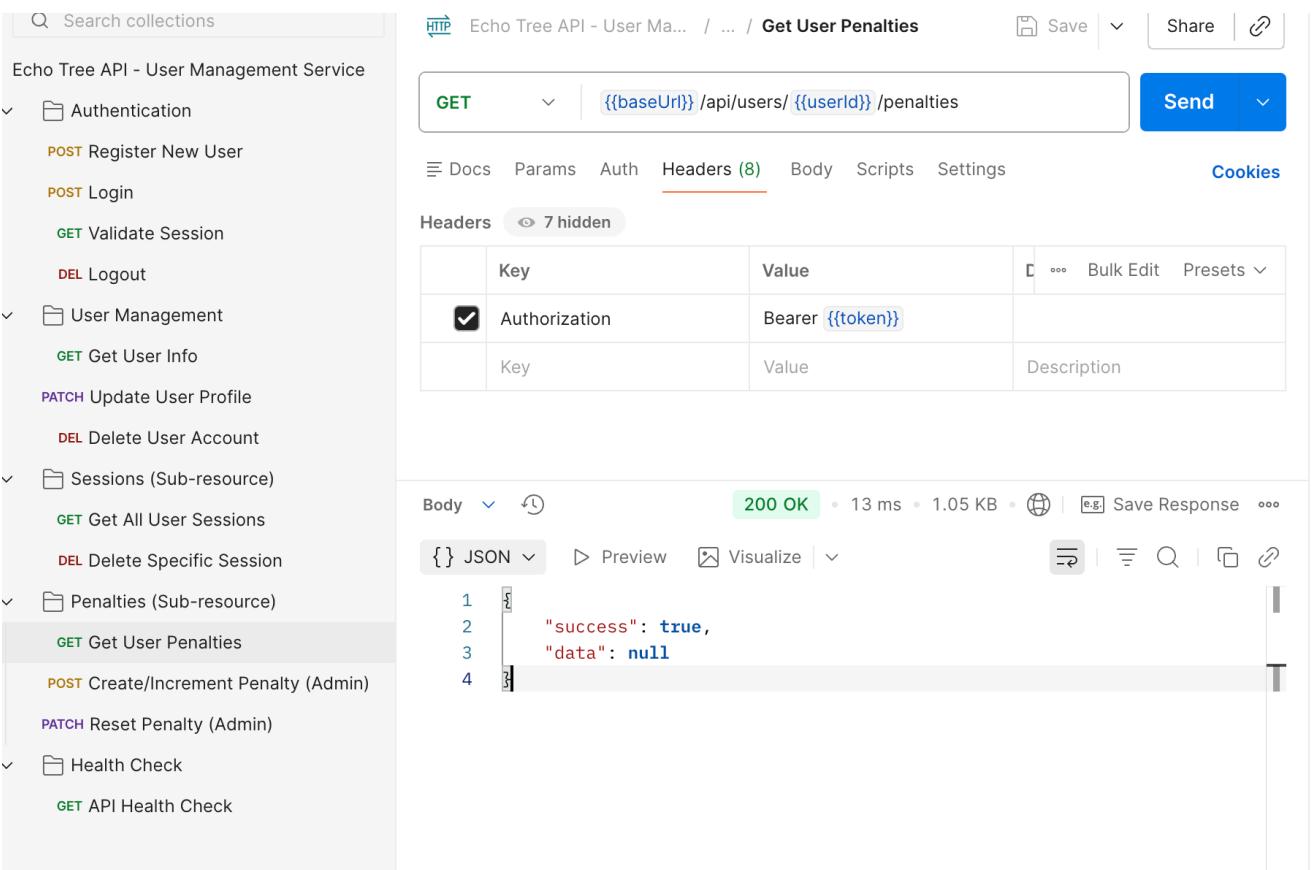
Request Body: None

Expected Response (Status: 200 OK):

```
{  
  
  "success": true,  
  
  "data": null
```

}

Test Result: PASS



The screenshot shows the Echo Tree API - User Management Service interface. On the left, there's a sidebar with a tree view of API endpoints under 'Echo Tree API - User Management Service'. The 'Penalties (Sub-resource)' node is expanded, and the 'GET Get User Penalties' endpoint is selected. The main panel shows the 'Get User Penalties' endpoint details. The method is 'GET', the URL is '{{baseUrl}}/api/users/{{userId}}/penalties', and the 'Headers' tab is selected, showing an 'Authorization' header with the value 'Bearer {{token}}'. Below this, the response is shown as a 200 OK status with a response body containing JSON:


```

1 {
2   "success": true,
3   "data": null
4 }
```

11. Create/Increment Penalty (Sub-resource - Admin Only)

k) Use Case: Add penalty to healer for harmful content

Resource(s): users/penalties

HTTP Method: POST

Request URL: {{baseUrl}}/api/users/{{userId}}/penalties

Request Headers:

```

Authorization: Bearer <ADMIN_TOKEN>
Content-Type: application/json
```

Request Body:

```
{
  "reason": "Inappropriate response detected",
  "reply_id": "691cd456ba685438e4301d12"
}
```

Expected Response (Status: 201 Created):

```
{  
  "success": true,  
  "data": {  
    "user_id": "691cd9dfba685438e4301bb9",  
    "harmful_count": 1,  
    "alert_sent": false,  
    "healer_status_removed": false,  
    "lastViolationAt": "2025-11-18T22:15:12.037Z",  
    "_id": "691ceff0ba685438e4301c19",  
    "updatedAt": "2025-11-18T22:15:12.040Z",  
    "createdAt": "2025-11-18T22:15:12.040Z",  
    "updatedAt": "2025-11-18T22:15:12.040Z",  
    "__v": 0,  
    "penalty_id": "691ceff0ba685438e4301c19",  
    "id": "691ceff0ba685438e4301c19"  
  }  
}
```

Test Result: ! REQUIRES ADMIN ROLE

The screenshot shows the Postman interface with the following details:

- Left Panel (Workspace):** Shows the Echo Tree API - User Management Service collection with various endpoints like Register New User, Login, Validate Session, Logout, Get User Info, Update User Profile, Delete User Account, Get All User Sessions, Delete Specific Session, Get User Penalties, Create/Increment Penalty (Admin), Reset Penalty (Admin), and API Health Check.
- Request URL:** {{baseUrl}} /api/users/ {{userId}} /penalties
- Method:** POST
- Headers:** Authorization: Bearer {{adminToken}}, Content-Type: application/json
- Body:** 201 Created (JSON response shown below)
- Response Body (JSON):**

```
1 {  
2   "success": true,  
3   "data": {  
4     "user_id": "691cd9dfba685438e4301bb9",  
5     "harmful_count": 1,  
6     "alert_sent": false,  
7     "healer_status_removed": false,  
8     "lastViolation_at": "2025-11-18T22:15:12.037Z",  
9     "_id": "691ceff0ba685438e4301c19",  
10    "updated_at": "2025-11-18T22:15:12.040Z",  
11    "createdAt": "2025-11-18T22:15:12.040Z",  
12    "updatedAt": "2025-11-18T22:15:12.040Z",  
13    "v": 0,  
14    "penalty_id": "691ceff0ba685438e4301c19",  
15    "id": "691ceff0ba685438e4301c19"  
16  }  
17 }
```

12. Reset Penalty (Sub-resource - Admin Only)

I) Use Case: Reset healer penalties (forgiveness/appeal)

Resource(s): users/penalties

HTTP Method: PATCH

Request URL: {{baseUrl}} /api/users/ {{userId}} /penalties

Request Headers:

Authorization: Bearer <ADMIN_TOKEN>

Content-Type: application/json

Request Body: None (or optional reason)

Expected Response (Status: 200 OK):

{

"success": true,

```
"data": {  
    "_id": "691ceff0ba685438e4301c19",  
    "user_id": "691cd9dfba685438e4301bb9",  
    "harmful_count": 0,  
    "alert_sent": false,  
    "healer_status_removed": false,  
    "lastViolationAt": null,  
    "updated_at": "2025-11-18T22:16:02.613Z",  
    "createdAt": "2025-11-18T22:15:12.040Z",  
    "updatedAt": "2025-11-18T22:16:02.613Z",  
    "__v": 0,  
    "penalty_id": "691ceff0ba685438e4301c19",  
    "id": "691ceff0ba685438e4301c19"  
}  
}
```

Test Result: ! REQUIRES ADMIN ROLE

Workspace New Import < PATCH Res | GET Get U | GET Get A | DEL Delete | > + - No environment

+ Search collections

✓ Echo Tree API - User Management Service

 Authentication

 POST Register New User

 POST Login

 GET Validate Session

 DEL Logout

 User Management

 GET Get User Info

 PATCH Update User Profile

 DEL Delete User Account

 Sessions (Sub-resource)

 GET Get All User Sessions

 DEL Delete Specific Session

 Penalties (Sub-resource)

 GET Get User Penalties

 POST Create/Increment Penalty (Admin)

 PATCH Reset Penalty (Admin)

 Health Check

 GET API Health Check

HTTP Echo Tree API - User ... / ... / Reset Penalty (Admin) Save Share

PATCH {{baseUrl}} /api/users/ {{userId}} /penalties Send

Docs Params Auth Headers (9) Body Scripts Settings Cookies

Query Params

| Key | Value | Description | Bulk Edit |
|-----|-------|-------------|-----------|
| Key | Value | Description | |

Body 200 OK 22 ms 1.4 KB Save Response

[{}] JSON Preview Visualize

```
1 {  
2   "success": true,  
3   "data": {  
4     "_id": "691ceff0ba685438e4301c19",  
5     "user_id": "691cd9dfba685438e4301bb9",  
6     "harmful_count": 0,  
7     "alert_sent": false,  
8     "healer_status_removed": false,  
9     "lastViolation_at": null,  
10    "updated_at": "2025-11-18T22:16:02.613Z",  
11    "createdAt": "2025-11-18T22:15:12.040Z",  
12    "updatedAt": "2025-11-18T22:16:02.613Z",  
13    "__v": 0,  
14    "penalty_id": "691ceff0ba685438e4301c19",  
15    "id": "691ceff0ba685438e4301c19"  
16  }  
17  2
```