Project Development Report: SynApp

Introduction

SynApp aims to connect people based on their interests in a number of topics. To begin with users will have the choice to communicate with (make synapses with) with other users based on their interest in Politics, Sports, Health or Tech.
Future updates will bring further topics and features but we at SynApp believe in developing the app with our users based on their response.

One of the unique features of SynApp is that users can set an expiration time for the post, after which the post cannot be reacted to. We believe this will encourage more positive interactions between users and minimise the negative aspects seen on other platforms. Users will still be able to browse the synapses either by topic or not.

This report provides a comprehensive overview of the Node.js Express application called "SynApp" for the Piazza platform. It details the architecture, database design, services, and API resources, along with references to external resources used during development.

Application Architecture

Technology Stack

- **Backend:** Node.js with Express framework
- **Database:** MongoDB with Mongoose ODM
- **Authentication:** JSON Web Tokens (JWT)
- **Validation:** Joi
- **Security:** bcryptjs for hashing

Key Components

- **Server Initialization: app.js** initializes the Express server and connects to the MongoDB database.
- **Middleware:** Includes body-parser for JSON parsing, custom token verification, and proposed validation middleware.
- **Route Handling:** Defined in **routes** directory, with separate files for **synapses** and **auth**.

Database Design

- **User Model:** Stores username, email, and hashed password.
- **Synapse Model:** Includes fields like title, body, user, comments, react(like/dislike), and topic.
- **Relationships:** References between synapse posts and users via ObjectIDs.

Service Descriptions

1. **User Authentication:**
    - Register: User sign-up with validation.
    - Login: User authentication and token generation.
2. **Synapse Management:**
    - Post Creation/Deletion/Update: CRUD operations on synapse posts.
    - Likes/Dislikes: User interactions with posts.

    - Commenting: Adding comments to posts.

API Resources

Endpoints

- **Auth:**
    - POST **/api/users/register**: User registration
    - POST **/api/users/login**: User login
- **Synapses:**
    - POST **/api/synapses**: Create a synapse
    - GET **/api/synapses**: Retrieve all synapses
    - PATCH **/api/synapses/:synapseId**: Update a synapse
    - DELETE **/api/synapses/:synapseId**: Delete a synapse
    - POST **/api/synapses/react/:synapseId**: React to a synapse
    - POST **/api/synapses/comments/:synapseId**: Comment on a synapse

References and External Resources

- **Node.js and Express:**
    - Official Documentation: Express.js
    - Tutorial: MDN Web Docs
- **MongoDB and Mongoose:**
    - Official Documentation: Mongoose
    - Guide: MongoDB University
- **Authentication and Security:**
    - JWT: jsonwebtoken NPM
    - Password Hashing: bcryptjs NPM
    - Security Best Practices: Node.js Security Checklist
- **Validation:**
    - Joi Validation: Joi NPM

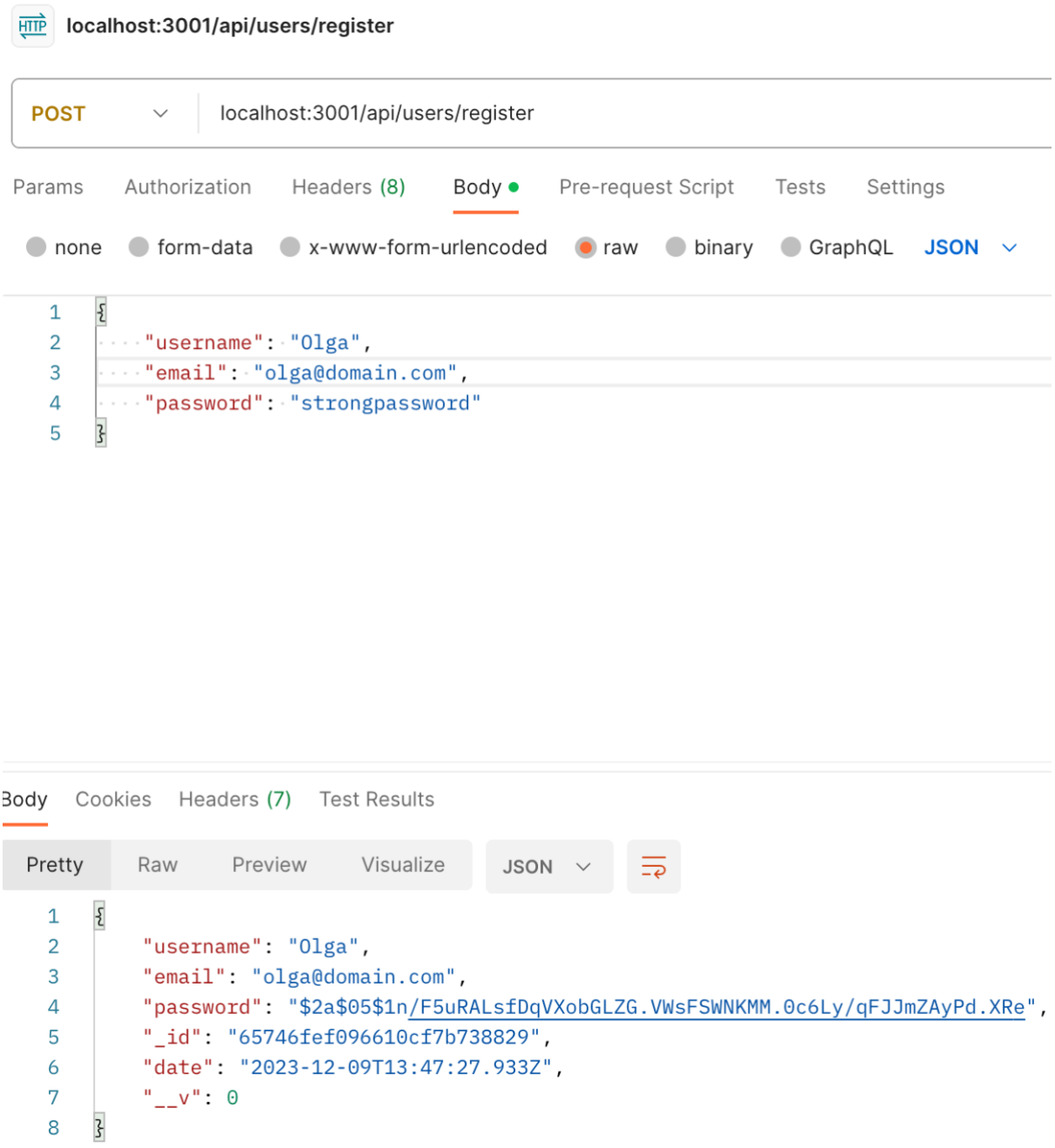
In addition, a number of other online teaching resources and forums were utilised during the development of this app;

- Learn JavaScript
- Learn Intermediate JavaScript
- Learn Node.js
- And of course: stack**overflow**

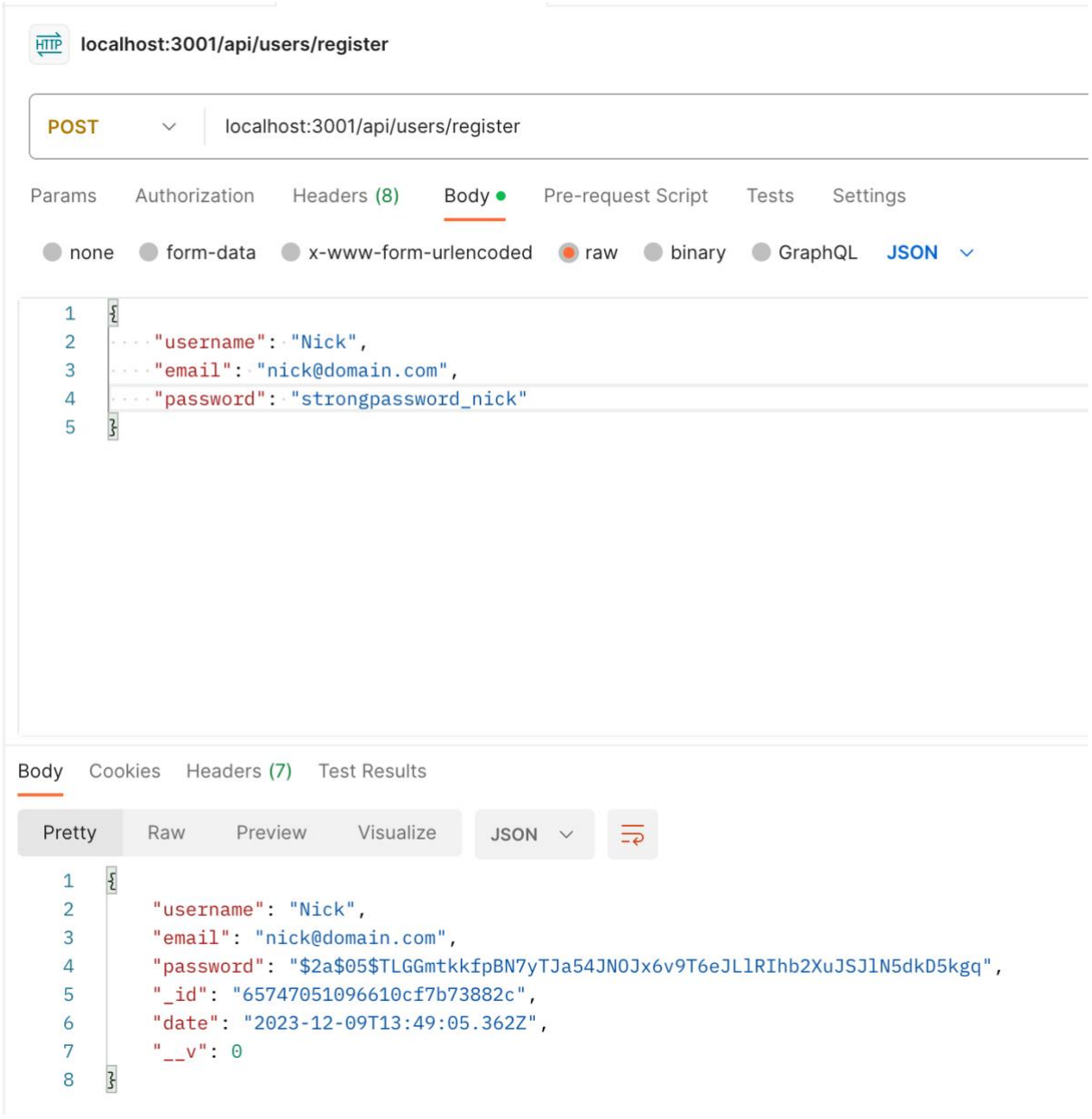
Test Cases

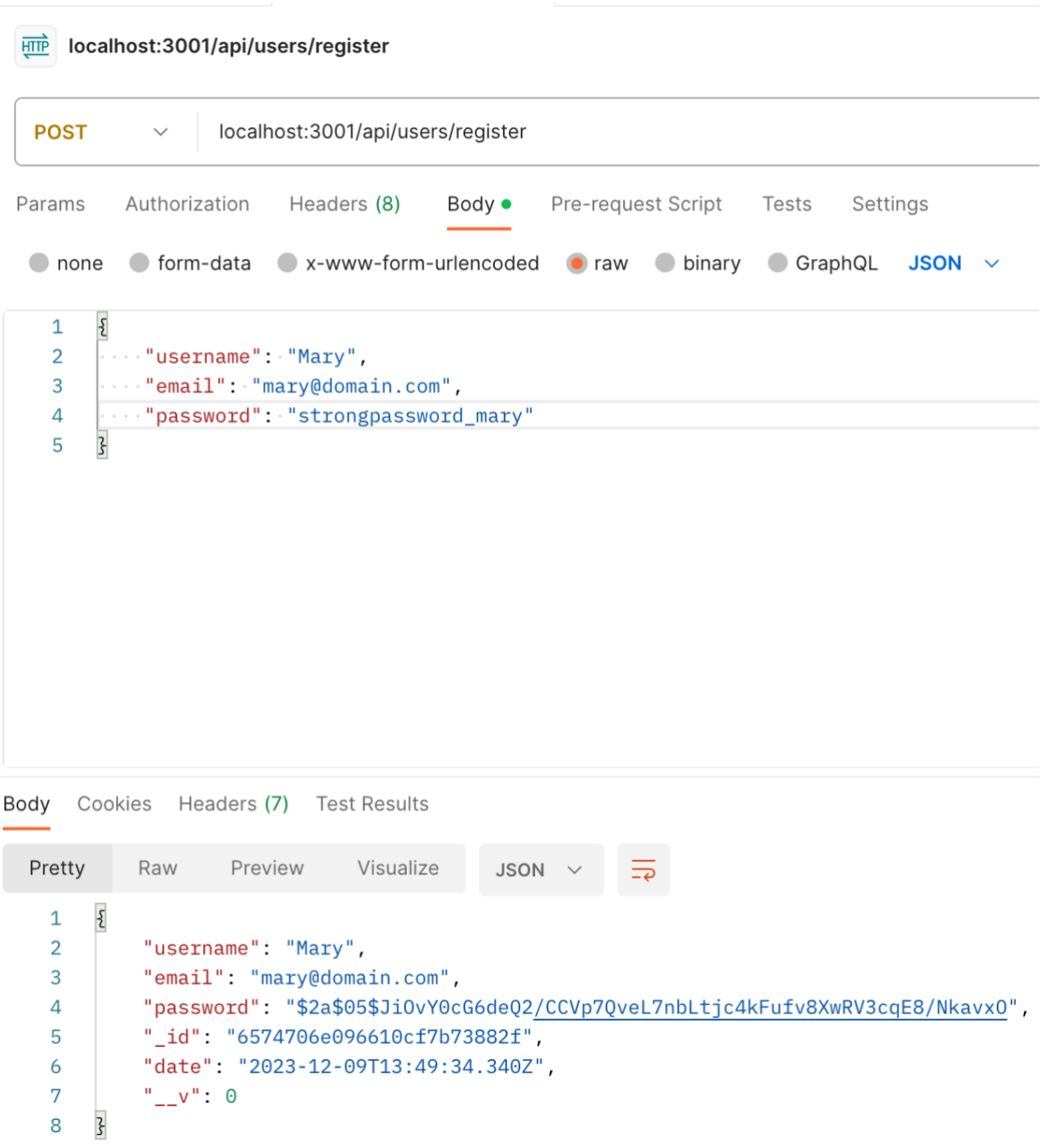
To test for the scenarios required in the course work, calls were made using Postman and screenshots are provided below for each testcase.

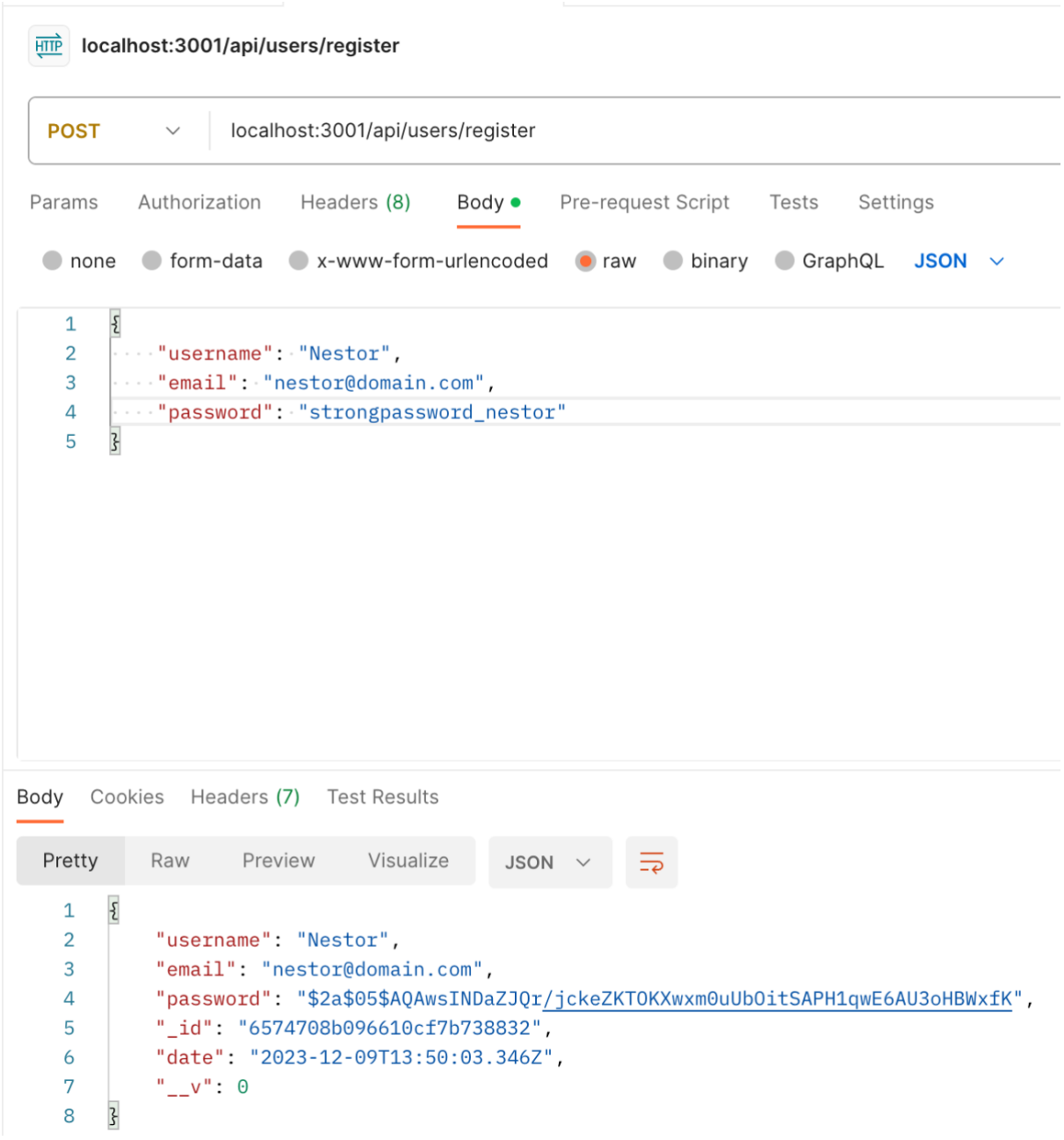
TC 1. Olga, Nick, Mary, and Nestor register and are ready to access the **Piazza** API.


localhost:3001/api/users/register
POST
localhost:3001/api/users/register
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"username": "Olga",
"email": "olga@domain.com",
"password": "strongpassword"
}

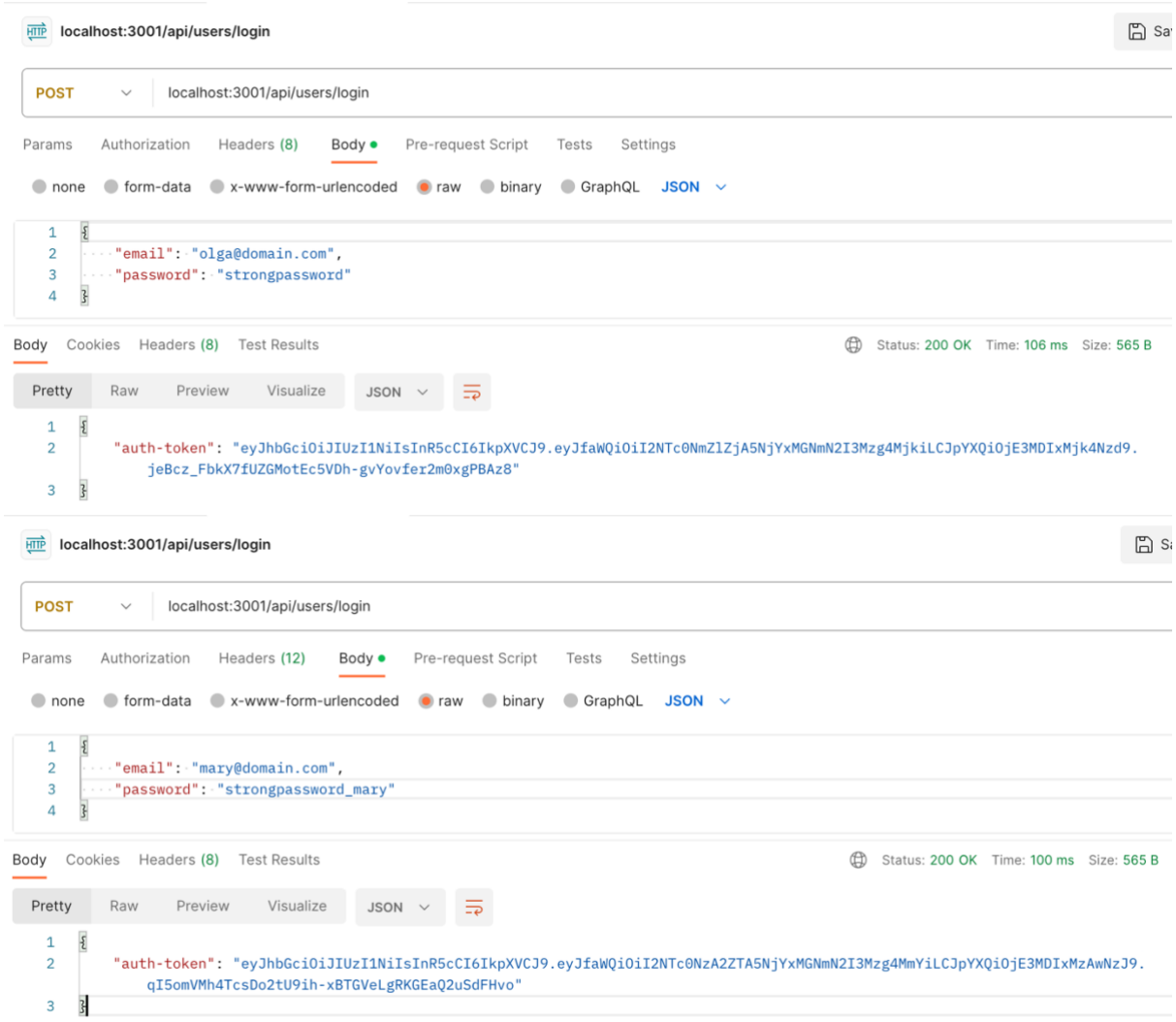
Body
Cookies
Headers (7)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"username": "Olga",
"email": "olga@domain.com",
"password": "$2a$05$1n/F5uRALsfDqVXobGLZG.VWsFSWNKMM.0c6Ly/qFJJmZAyPd.XRe",
"_id": "65746fef096610cf7b738829",
"date": "2023-12-09T13:47:27.933Z",
"__v": 0
}


localhost:3001/api/users/register
POST
localhost:3001/api/users/register
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"username": "Nick",
"email": "nick@domain.com",
"password": "strongpassword_nick"
}
Body
Cookies
Headers (7)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"username": "Nick",
"email": "nick@domain.com",
"password": "$2a$05$TLGGmtkkfpBN7yTJa54JNOJx6v9T6eJLlRIhb2XuJSJlN5dkD5kgq",
"_id": "65747051096610cf7b73882c",
"date": "2023-12-09T13:49:05.362Z",
"__v": 0
}

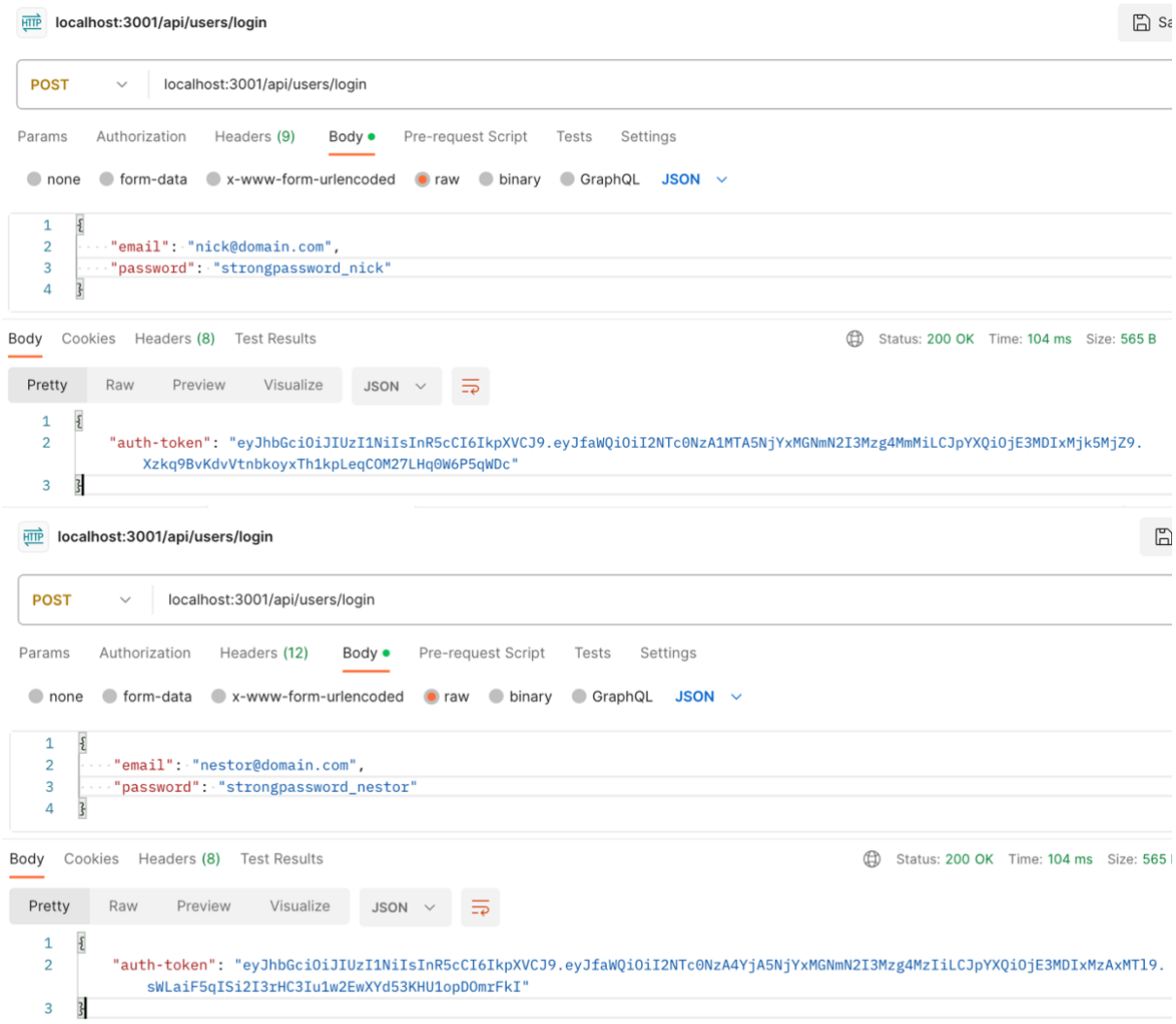

localhost:3001/api/users/register
POST
localhost:3001/api/users/register
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"username": "Mary",
"email": "mary@domain.com",
"password": "strongpassword_mary"
}
Body
Cookies
Headers (7)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"username": "Mary",
"email": "mary@domain.com",
"password": "$2a$05$JiOvY0cG6deQ2/CCVp7QveL7nbLtjc4kFufv8XwRV3cqE8/NkavxO",
"_id": "6574706e096610cf7b73882f",
"date": "2023-12-09T13:49:34.340Z",
"__v": 0
}



TC 2. Olga, Nick, Mary, and Nestor use the oAuth v2 authorisation service to register and get their tokens.

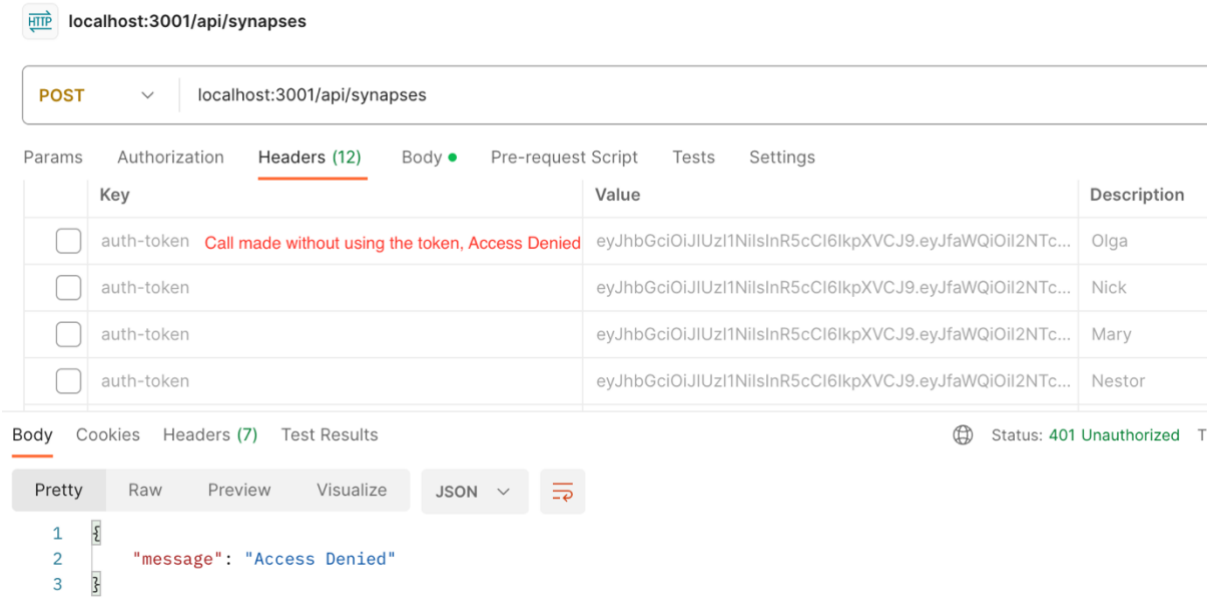

localhost:3001/api/users/login
POST
localhost:3001/api/users/login
Params
Authorization
Headers (8)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"email": "olga@domain.com",
"password": "strongpassword"
}
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 106 ms
Size: 565 B
Pretty
Raw
Preview
Visualize
JSON
{
"auth-token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NmZlZjA5NjYxMGNmN2I3Mzg4MjkiLCJpYXQiOjE3MDIxMjk4Nzd9.
jeBcz_FbkX7fUZGMotEc5VDh-gvYovfer2m0xgPBAz8"
}

localhost:3001/api/users/login
POST
localhost:3001/api/users/login
Params
Authorization
Headers (12)
Body
Pre-request Script
Tests
Settings
none
form-data
x-www-form-urlencoded
raw
binary
GraphQL
JSON
{
"email": "mary@domain.com",
"password": "strongpassword_mary"
}
Body
Cookies
Headers (8)
Test Results
Status: 200 OK
Time: 100 ms
Size: 565 B
Pretty
Raw
Preview
Visualize
JSON
{
"auth-token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA2ZTA5NjYxMGNmN2I3Mzg4MmYiLCJpYXQiOjE3MDIxMzAwNzJ9.
qI5omVMh4TcsDo2tU9ih-xBTGVeLgRKGEaQ2uSdFHvo"
}

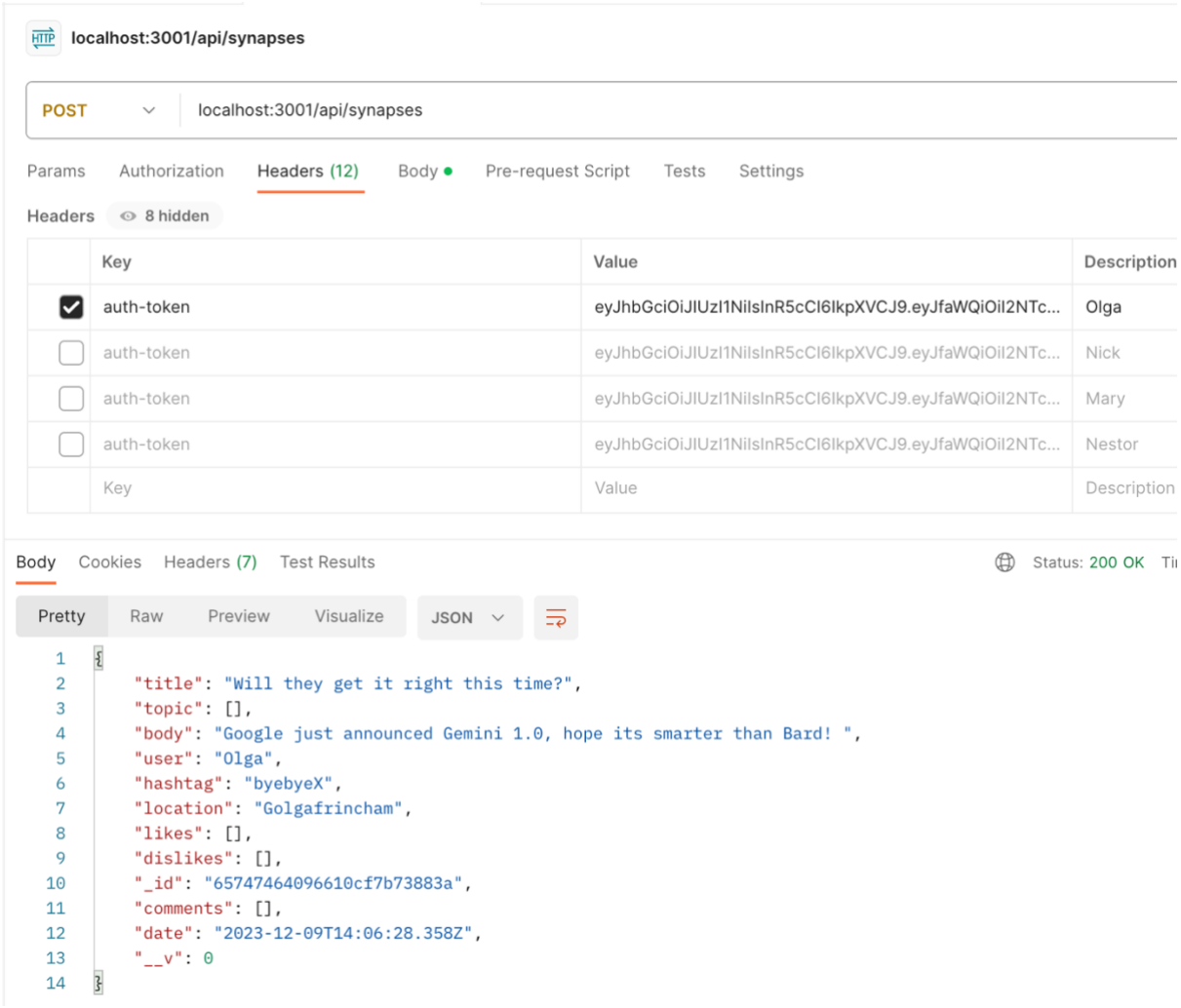

localhost:3001/api/users/login
POST
localhost:3001/api/users/login
Params Authorization Headers (9) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
{
"email": "nick@domain.com",
"password": "strongpassword_nick"
}
Body Cookies Headers (8) Test Results
Status: 200 OK Time: 104 ms Size: 565 B
Pretty Raw Preview Visualize JSON
{
"auth-token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA1MTA5NjYxMGNmN2I3Mzg4MmMiLCJpYXQiOjE3MDIxMjk5MjZ9.
Xzkq9BvKdvVtnbkoyxTh1kpLeqCOM27LHq0W6P5qWDc"
}

localhost:3001/api/users/login
POST
localhost:3001/api/users/login
Params Authorization Headers (12) Body Pre-request Script Tests Settings
none form-data x-www-form-urlencoded raw binary GraphQL JSON
{
"email": "nestor@domain.com",
"password": "strongpassword_nestor"
}
Body Cookies Headers (8) Test Results
Status: 200 OK Time: 104 ms Size: 565
Pretty Raw Preview Visualize JSON
{
"auth-token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA4YjA5NjYxMGNmN2I3Mzg4MzIiLCJpYXQiOjE3MDIxMzAxMTl9.
sWLaiF5qISi2I3rHC3Iu1w2EwXYd53KHU1opDOmrFkI"
}

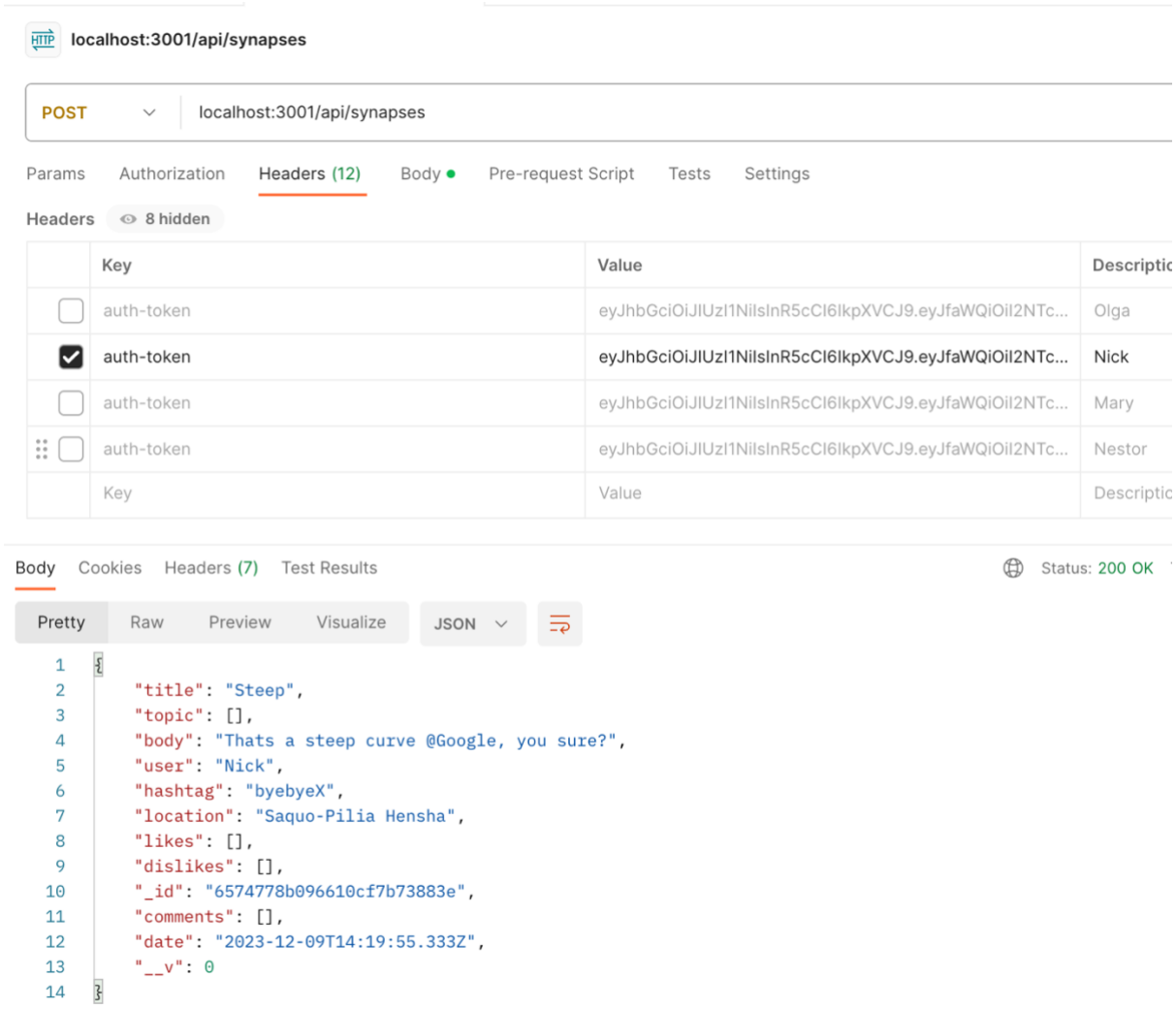
TC 3. Olga makes a call to the API without using her token. This call should be unsuccessful as the user is unauthorised.



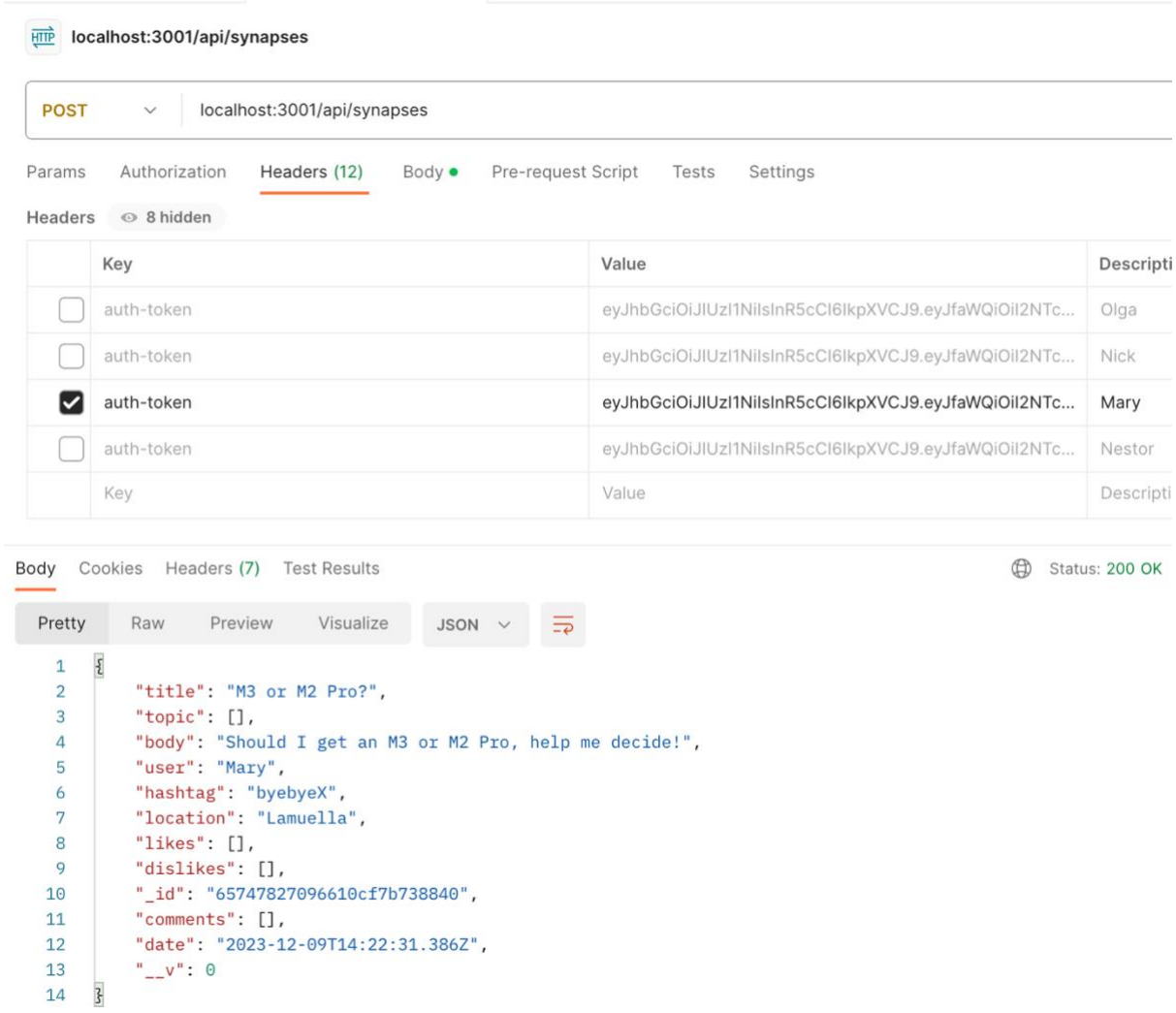
TC 4. Olga posts a message in the Tech topic with an expiration time (e.g. 5 minutes) using her token. After the end of the expiration time, the message will not accept any further user interactions (likes, dislikes, or comments).



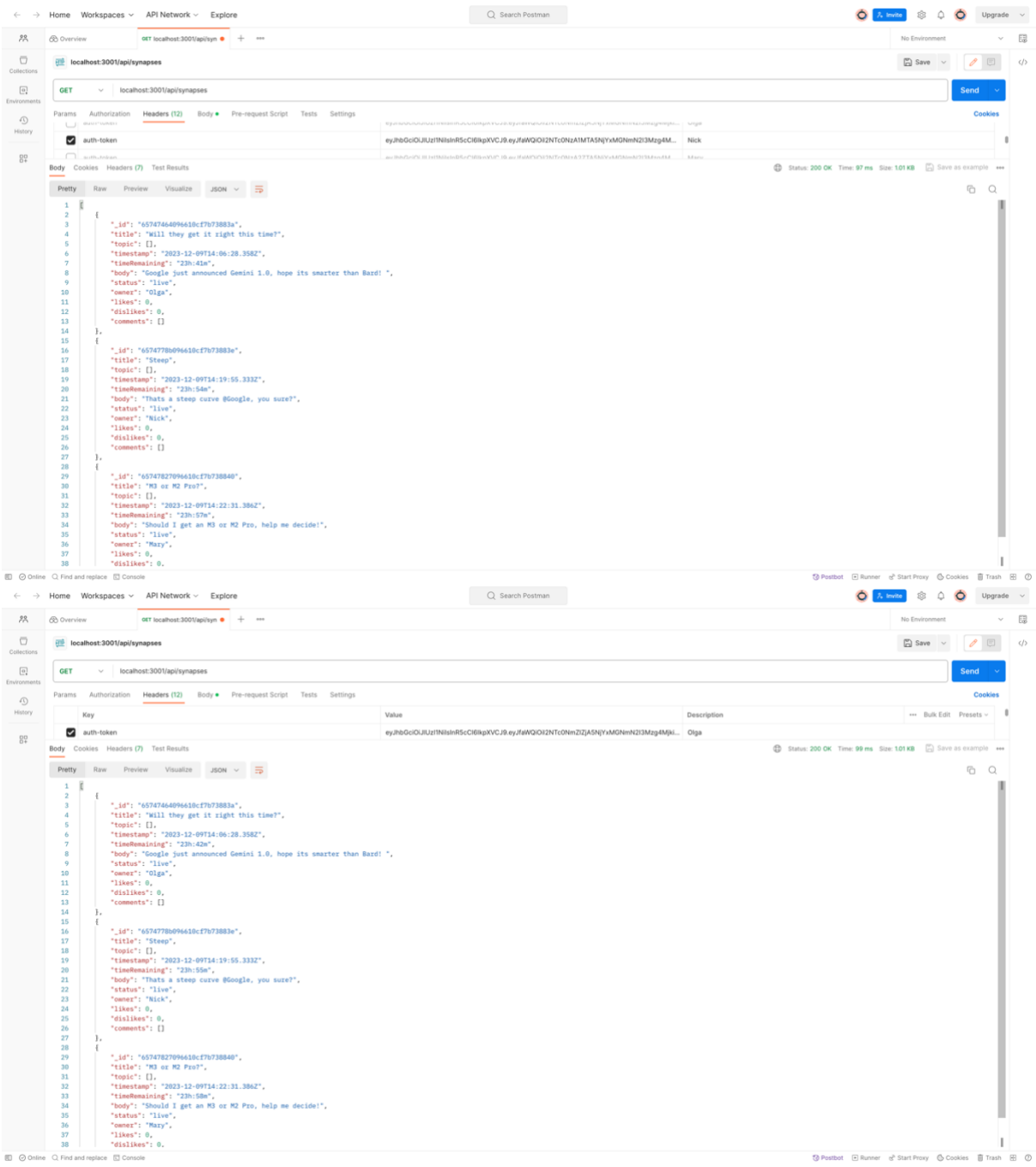
TC5. Nick posts a message in the Tech topic with an expiration time using his token.



TC6. Mary posts a message in the Tech topic with an expiration time using her token.

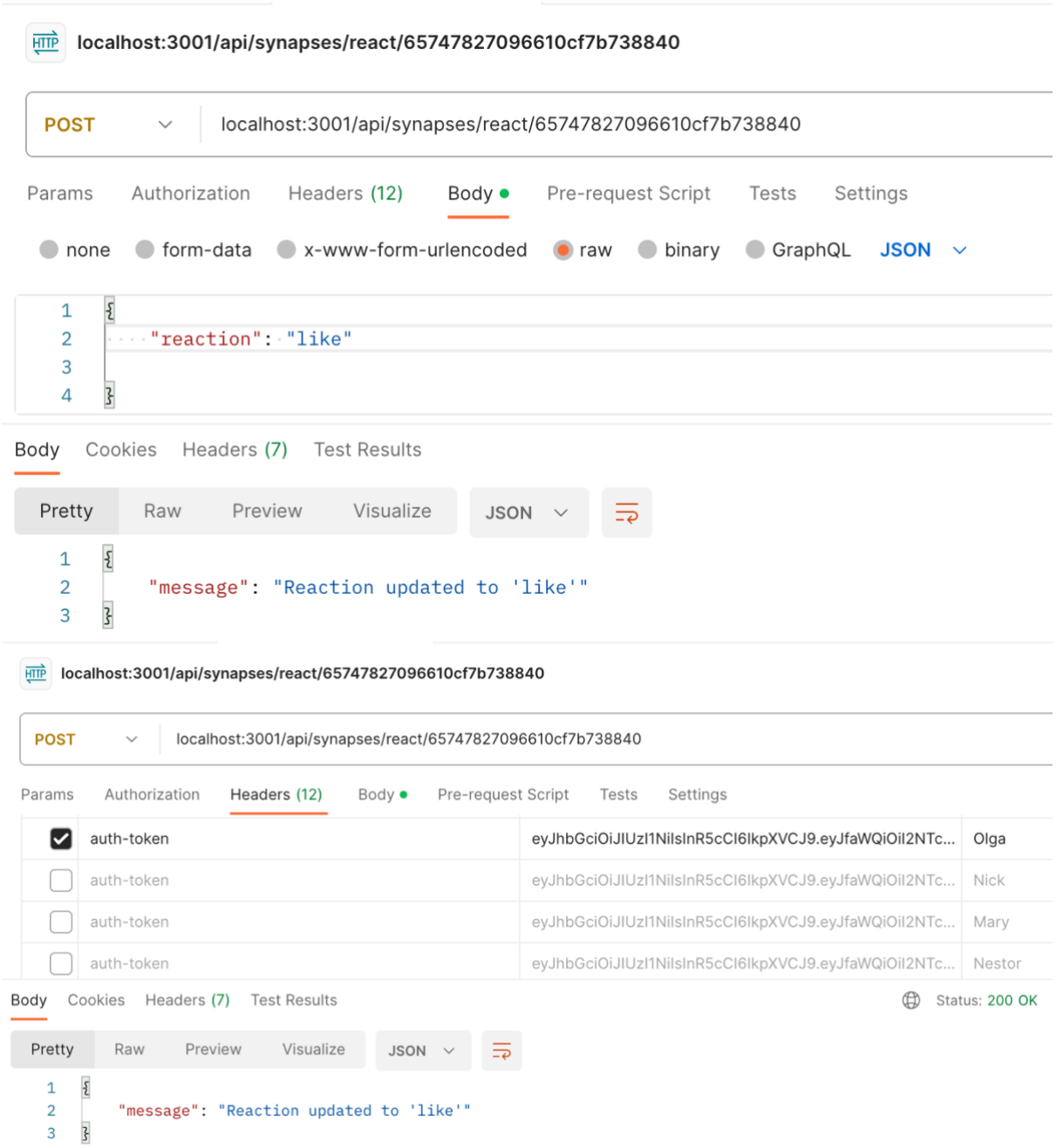


TC7 Nick and Olga browse all the available posts in the Tech topic; three posts should be available with zero likes, zero dislikes and no comments.



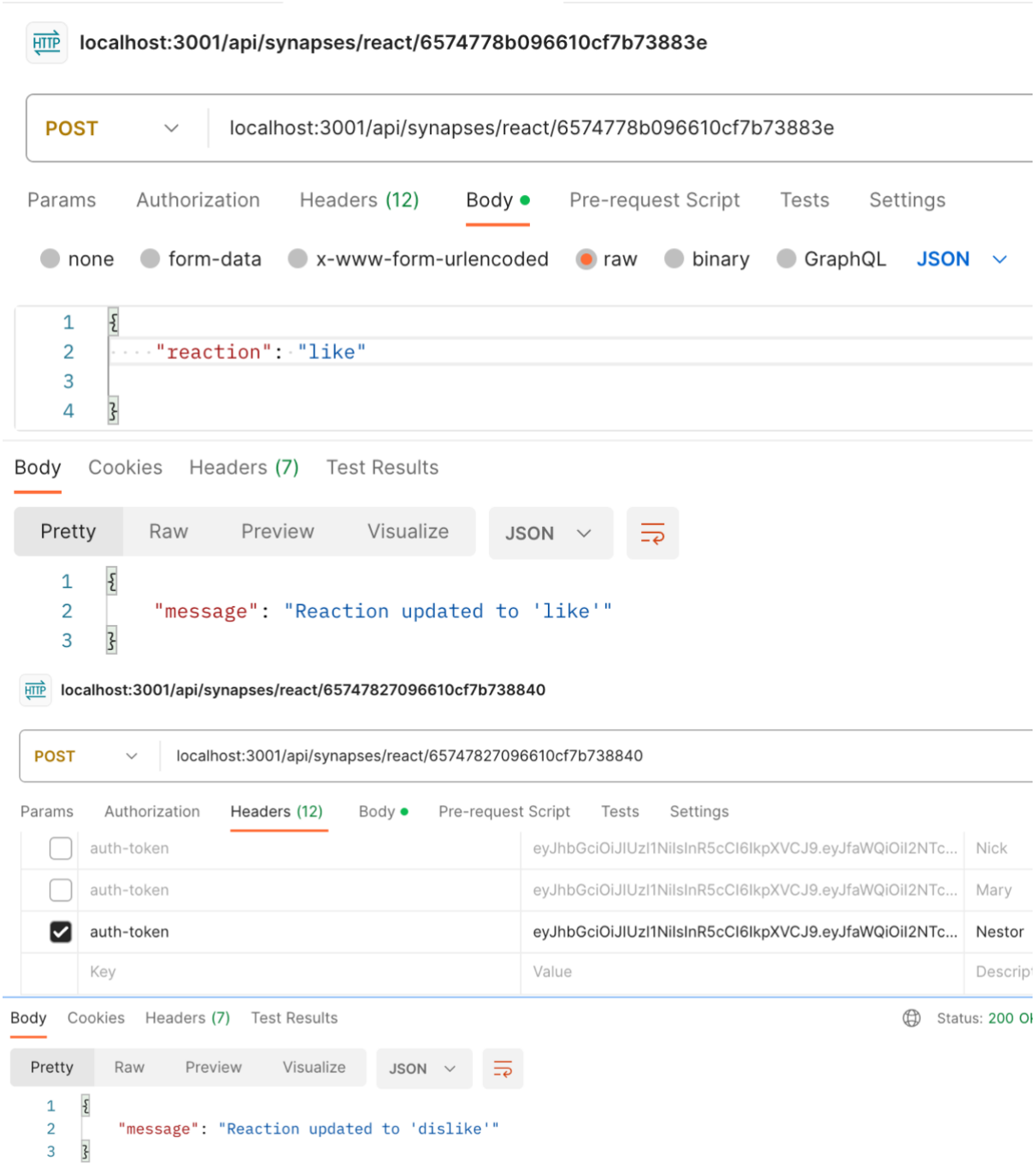
TC8 Nick and Olga “like” Mary’s post on the Tech topic.

(One screenshot each provided for the call made and the token used)

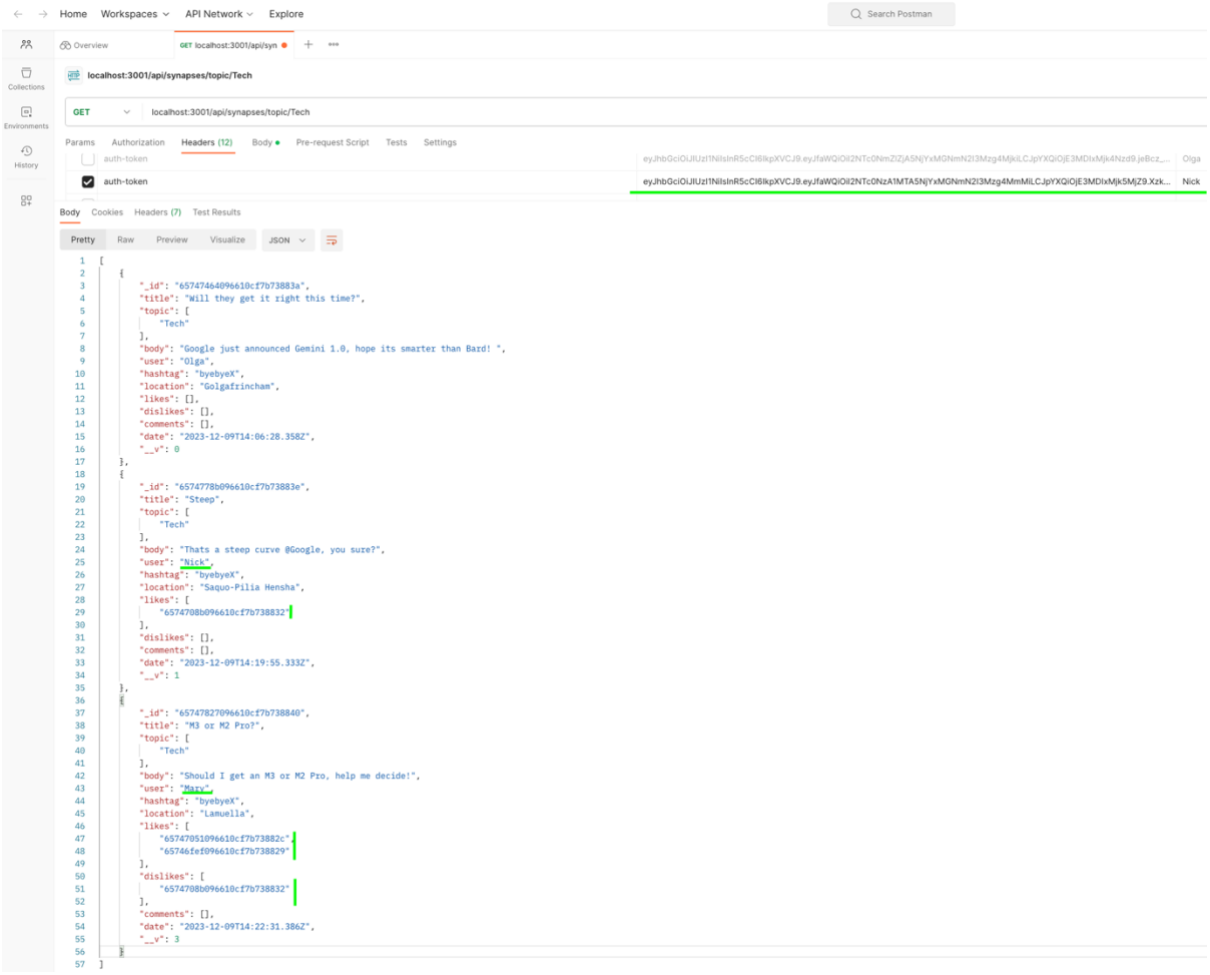


TC9 Nestor “likes” Nick’s post and “dislikes” Mary’s on the Tech topic.

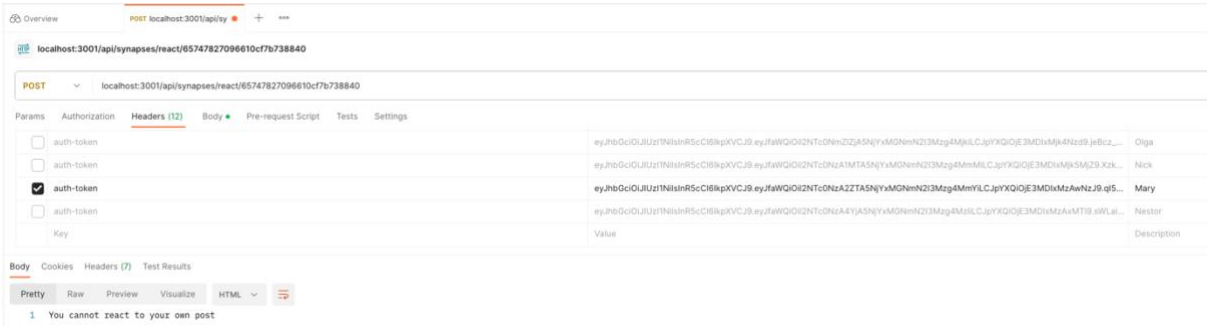
(one of each provided for to save space)



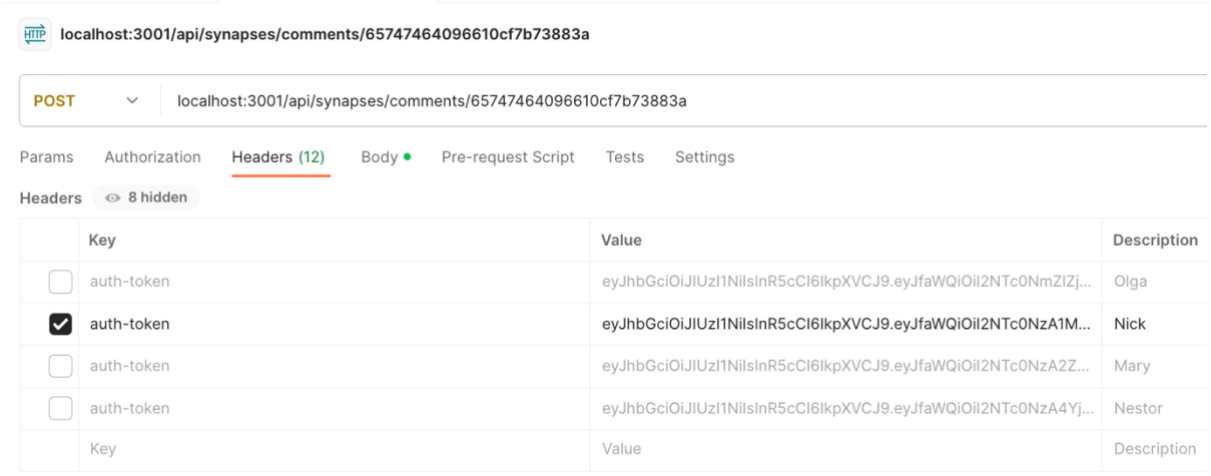
TC10 Nick browses all the available posts on the Tech topic; at this stage, he can see the number of likes and dislikes for each post (Mary has two likes and one dislike, and Nick has one like). There are no comments made yet.

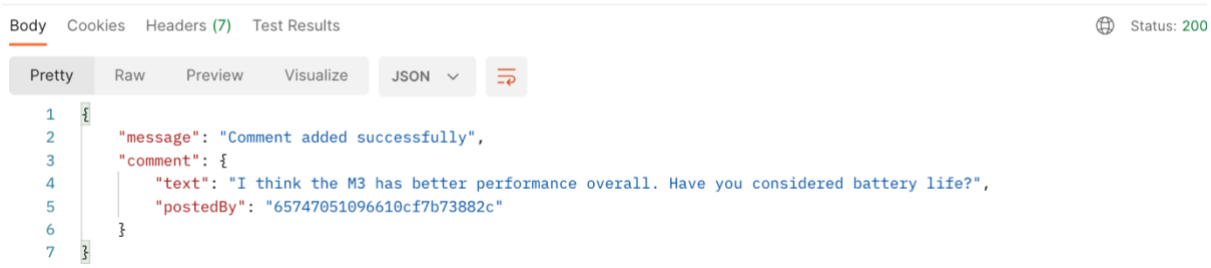


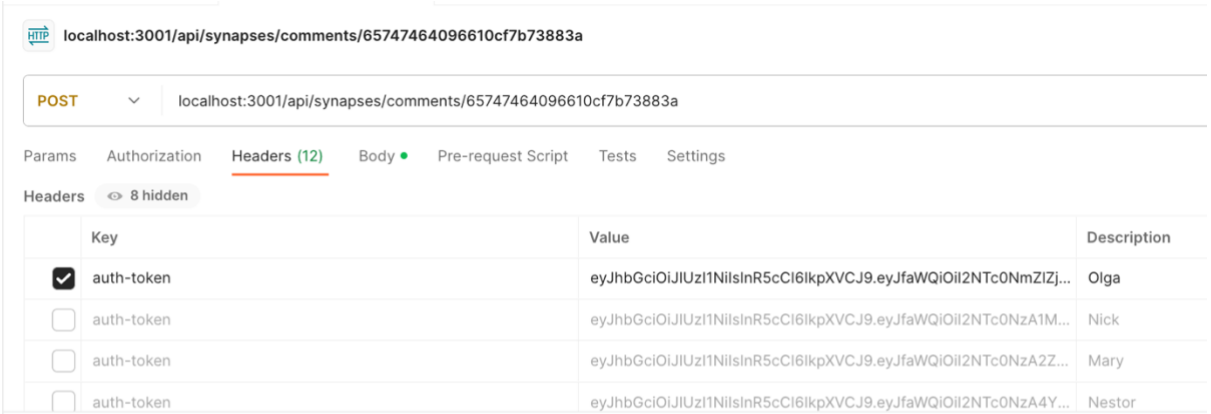
TC11 Mary likes her post on the Tech topic. This call should be unsuccessful; in **Piazza,** a post owner cannot like their messages.

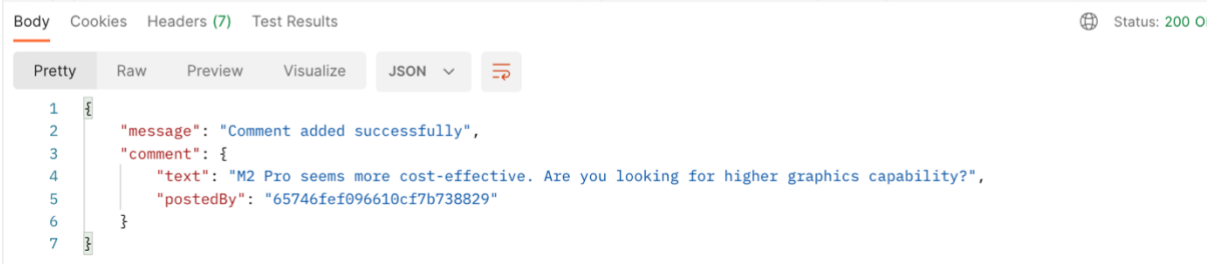


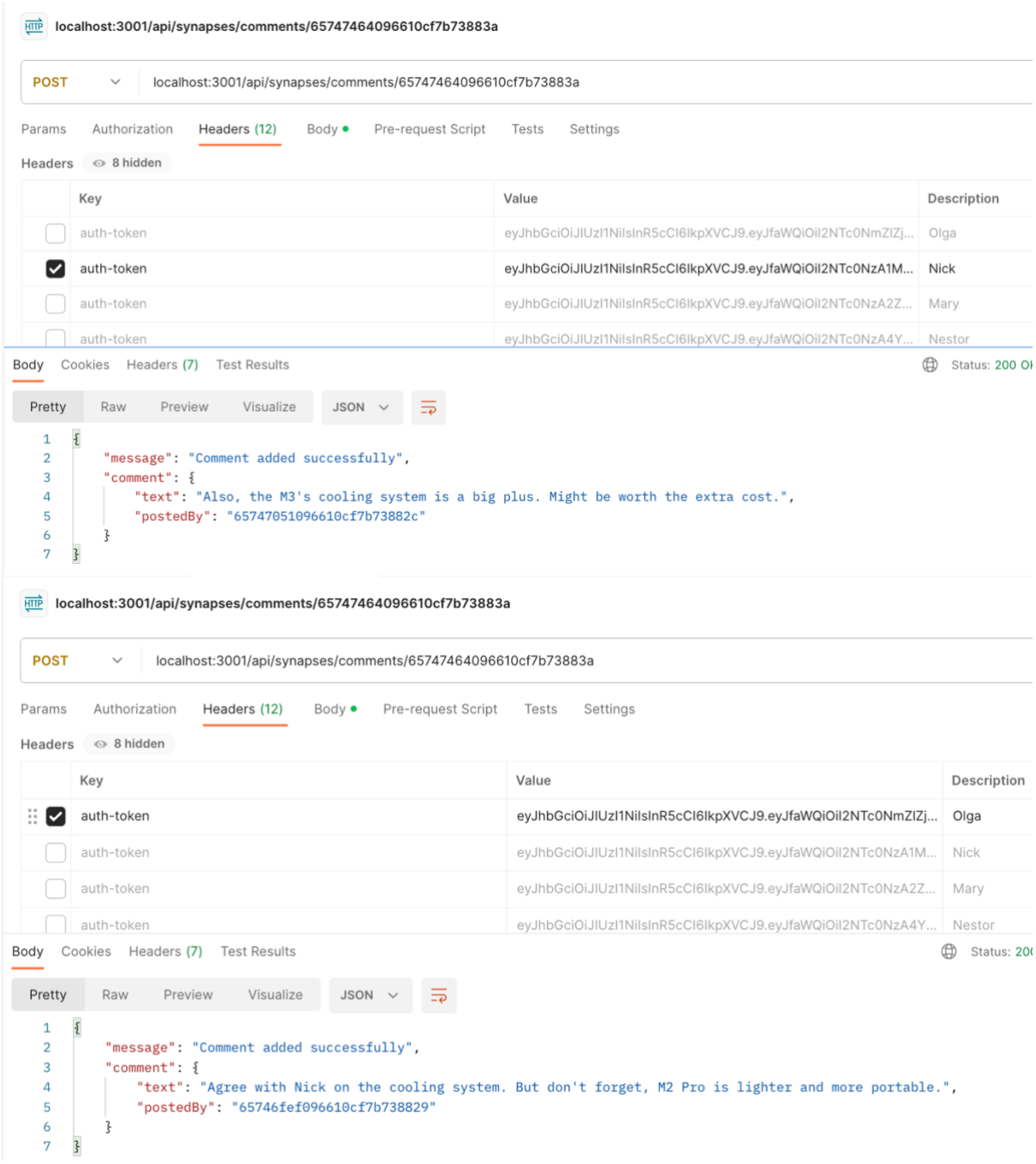
TC12 Nick and Olga comment on Mary's post on the Tech topic in a round-robin fashion (one after the other, adding at least two comments each).


localhost:3001/api/synapses/comments/65747464096610cf7b73883a
POST
localhost:3001/api/synapses/comments/65747464096610cf7b73883a
Params
Authorization
Headers (12)
Body
Pre-request Script
Tests
Settings
Headers
8 hidden
Key
Value
Description
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NmZlZj...
Olga
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA1M...
Nick
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA2Z...
Mary
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA4Yj...
Nestor
Key
Value
Description

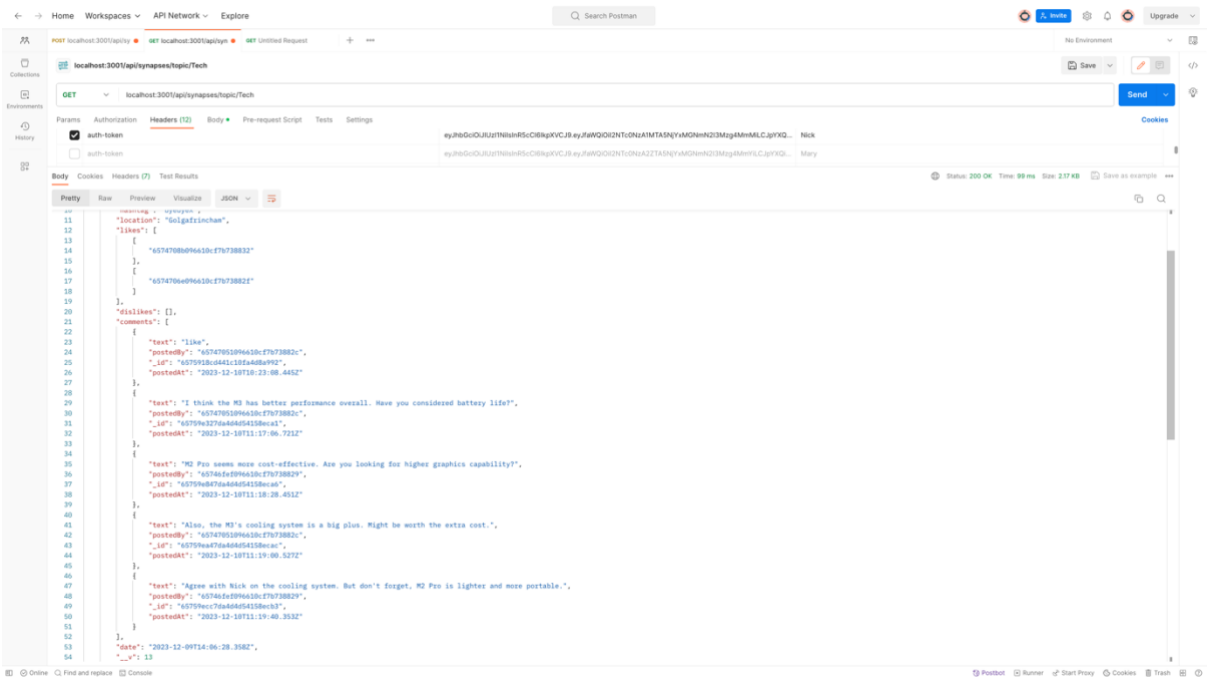

Body
Cookies
Headers (7)
Test Results
Status: 200
Pretty
Raw
Preview
Visualize
JSON
{
"message": "Comment added successfully",
"comment": {
"text": "I think the M3 has better performance overall. Have you considered battery life?",
"postedBy": "65747051096610cf7b73882c"
}
}


localhost:3001/api/synapses/comments/65747464096610cf7b73883a
POST
localhost:3001/api/synapses/comments/65747464096610cf7b73883a
Params
Authorization
Headers (12)
Body
Pre-request Script
Tests
Settings
Headers
8 hidden
Key
Value
Description
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NmZlZj...
Olga
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA1M...
Nick
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA2Z...
Mary
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA4Y...
Nestor

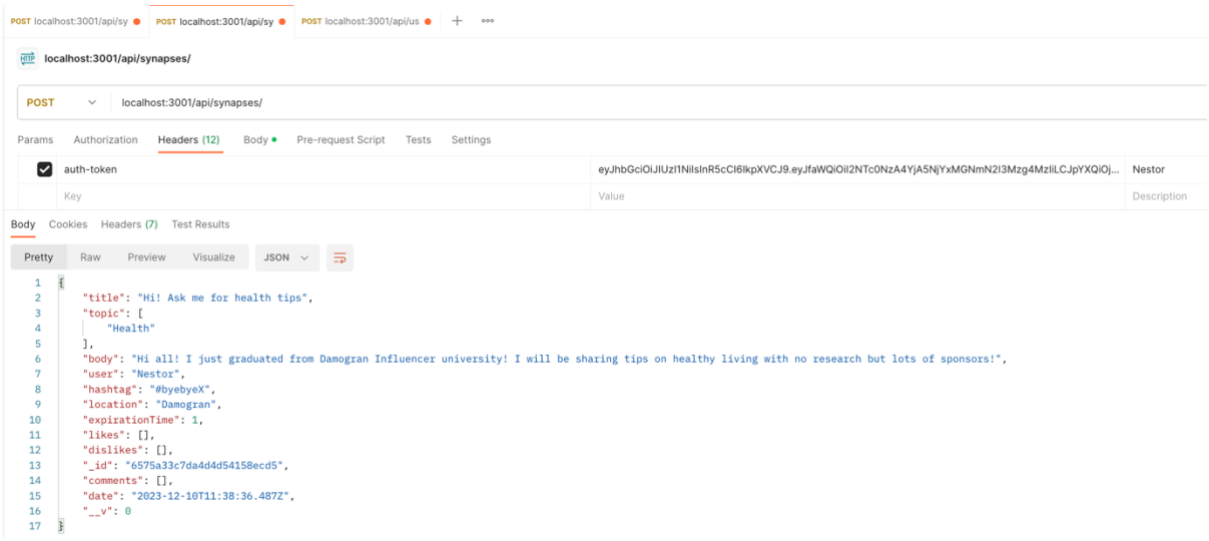

Body
Cookies
Headers (7)
Test Results
Status: 200 O
Pretty
Raw
Preview
Visualize
JSON
{
"message": "Comment added successfully",
"comment": {
"text": "M2 Pro seems more cost-effective. Are you looking for higher graphics capability?",
"postedBy": "65746fef096610cf7b738829"
}
}



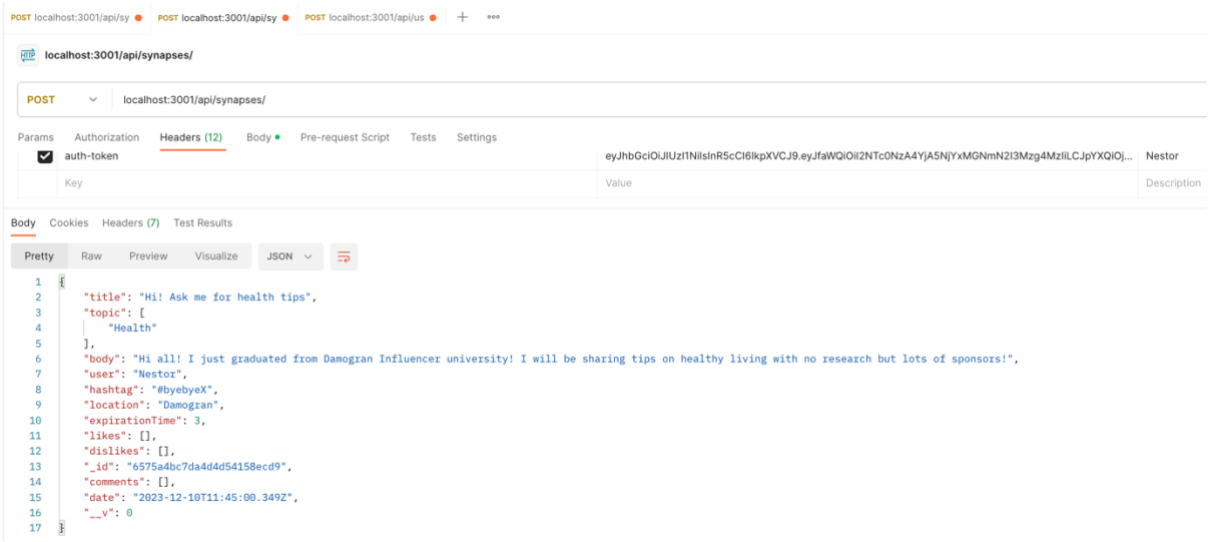
TC13 Nick browses all the available posts in the Tech topic; at this stage, he can see the number of likes and dislikes of each post and the comments made.



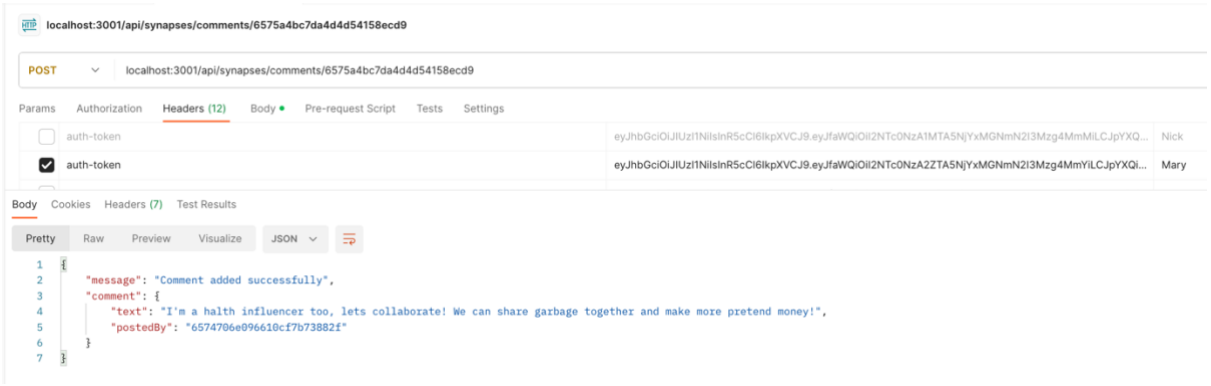
TC14 Nestor posts a message in the Health topic with an expiration time using her token.



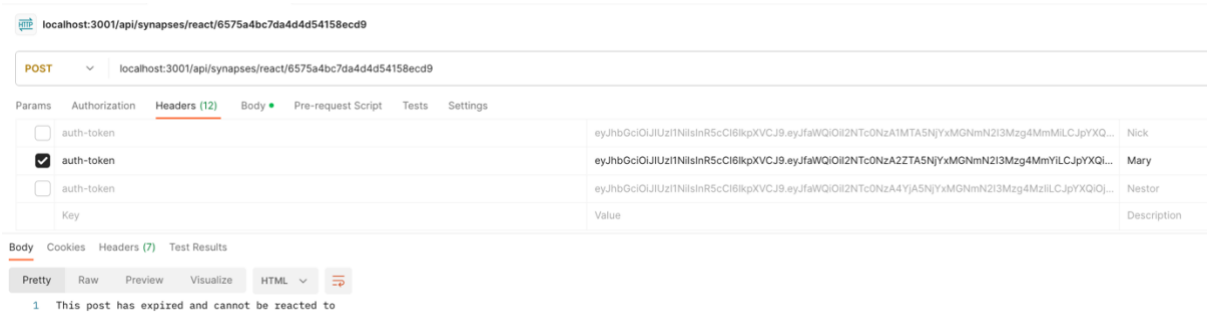
TC15 Mary browses all the available posts on the Health topic; at this stage, she can see only Nestor’s post.



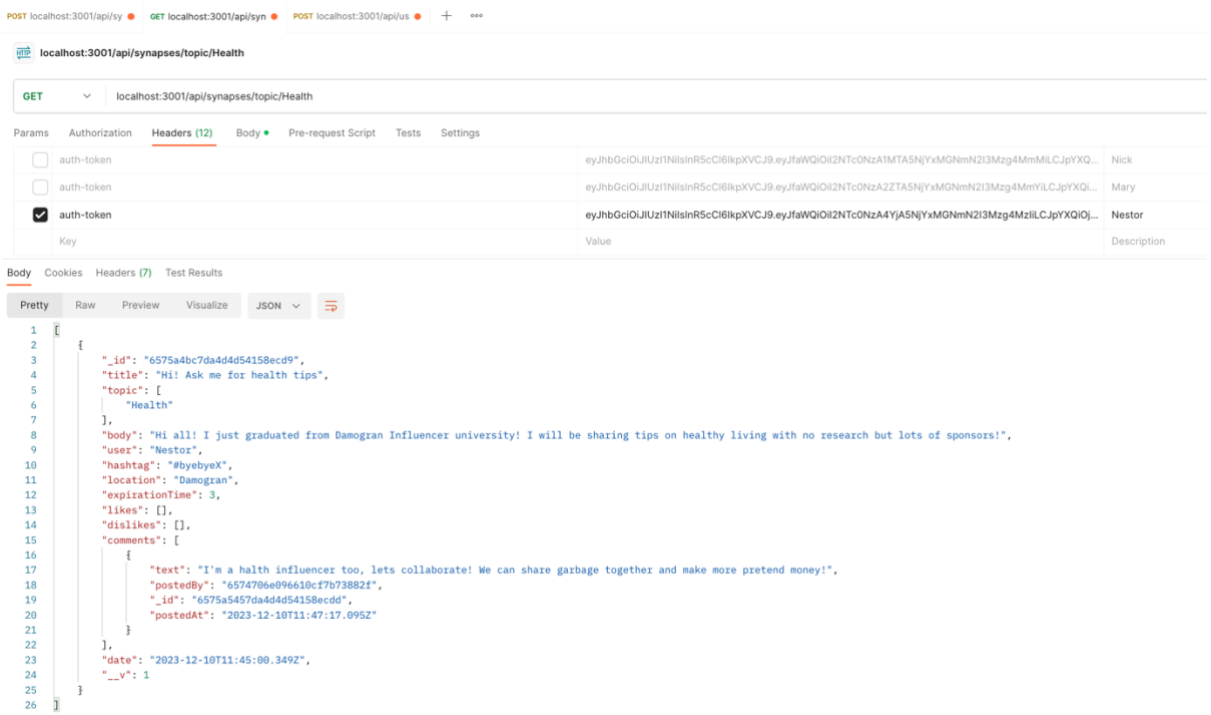
TC16 Mary posts a comment in Nestor's message on the Health topic.



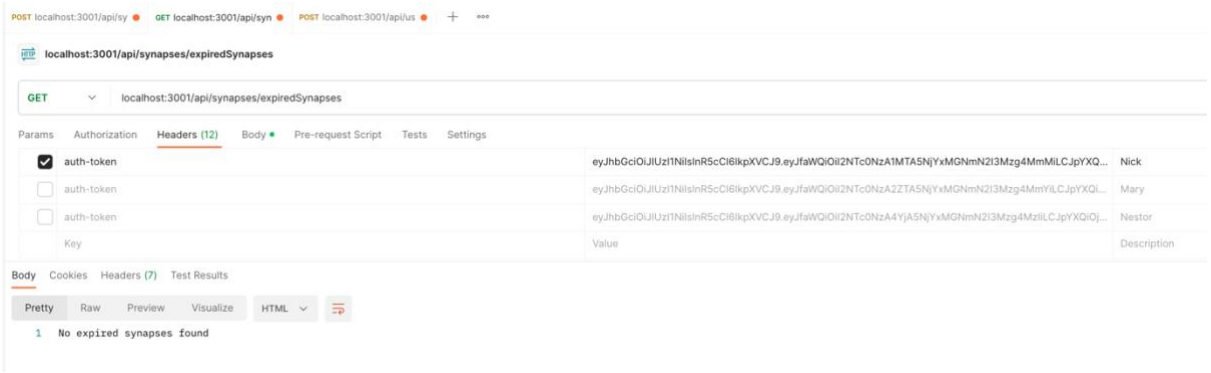
TC17 Mary dislikes Nestor's message on the Health topic after the end of post-expiration time. This should fail.



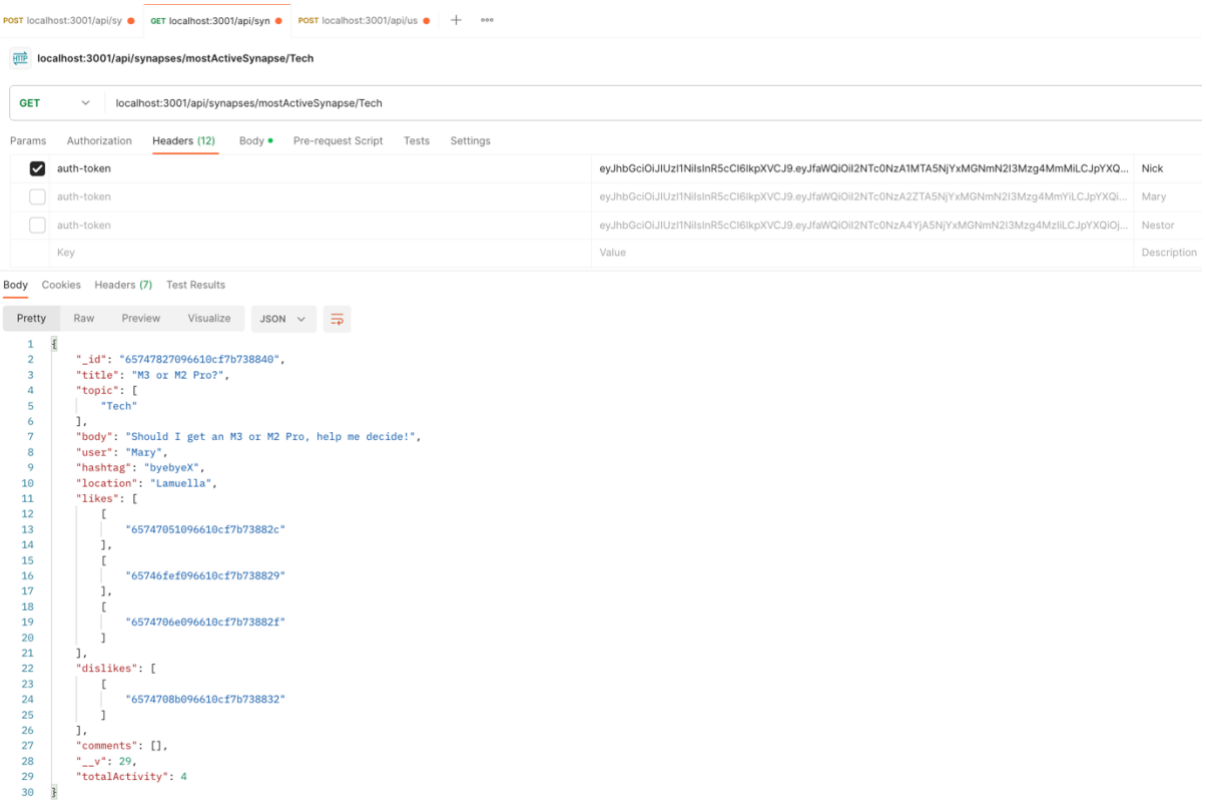
TC18 Nestor browses all the messages on the Health topic. There should be only one post (his own) with one comment (Mary's).



TC19 Nick browses all the expired messages on the Sports topic. These should be empty.



TC20 Nestor queries for an active post with the highest interest (maximum number of likes and dislikes) in the Tech topic. This should be Mary's post.


POST localhost:3001/api/sy
GET localhost:3001/api/syn
POST localhost:3001/api/us
localhost:3001/api/synapses/mostActiveSynapse/Tech
GET
localhost:3001/api/synapses/mostActiveSynapse/Tech
Params
Authorization
Headers (12)
Body
Pre-request Script
Tests
Settings
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA1MTA5NjYxMGNmN2I3Mzg4MmMiLCJpYXQ...
Nick
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA2ZTA5NjYxMGNmN2I3Mzg4MmYiLCJpYXQi...
Mary
auth-token
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJfaWQiOiI2NTc0NzA4YjA5NjYxMGNmN2I3Mzg4MzliLCJpYXQiOj...
Nestor
Key
Value
Description
Body
Cookies
Headers (7)
Test Results
Pretty
Raw
Preview
Visualize
JSON
{
"_id": "65747827096610cf7b738840",
"title": "M3 or M2 Pro?",
"topic": [
"Tech"
],
"body": "Should I get an M3 or M2 Pro, help me decide!",
"user": "Mary",
"hashtag": "byebyeX",
"location": "Lamuella",
"likes": [
[
"65747051096610cf7b73882c"
],
[
"65746fef096610cf7b738829"
],
[
"6574706e096610cf7b73882f"
]
],
"dislikes": [
[
"6574708b096610cf7b738832"
]
],
"comments": [],
"__v": 29,
"totalActivity": 4
}