

Echo Tree - Chat & Message Service + Note & Tree Hole Service API Actual Test Report

Tester: API Testing Suite

Base URL: http://localhost:3000

Services Tested: Chat & Message Service, Note & Tree Hole Service

Test User: admin (admin@echotree.com)

Token:

```
eyJhbGciOiJIUzI1NilsInR5cCI6IkpXVCJ9.eyJ1c2VySWQiOiI2OTFjZDIkZmJhNjg1NDM4ZTQ  
zMDFiYjkiLCJpYXQiOjE3NjM1NDMzMjcsImV4cCI6MTc2NDE0ODEyN30.W9IvyMDbE5klkO  
YWam1XK6U9IRDAvCANjINT8aWly30  
user_id": "691cd9dfba685438e4301bb9"
```

Chat & Message Service API Tests

1. Create New Chat

Use Case

User creates a new private chat session with the tree

API Details

Resource: chats

HTTP Method: POST

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

Request Headers:

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

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

Request Body:

```
{
```

```
"title": "Testing Chat Service"  
}  
}
```

Expected Response:

- **Status Code:** 201 Created
- **Success:** true
- **Response Body:**

```
{  
  
  "success": true,  
  
  "data": {  
  
    "user_id": "691cd9dfba685438e4301bb9",  
  
    "start_time": "2025-11-19T09:09:53.331Z",  
  
    "closed_at": null,  
  
    "message_count": 0,  
  
    "deleted_at": null,  
  
    "_id": "691d8961ba685438e4301c62",  
  
    "createdAt": "2025-11-19T09:09:53.332Z",  
  
    "updatedAt": "2025-11-19T09:09:53.332Z",  
  
    "__v": 0,  
  
    "chat_id": "691d8961ba685438e4301c62",  
  
    "id": "691d8961ba685438e4301c62"  
  
  }  
}
```

Test Result:  PASS

Notes: Chat created successfully.

The screenshot shows the Postman interface. On the left, the sidebar displays the 'Echo Tree - Complete API Collection' with various endpoints under 'Chat & Messages'. The 'Create New Chat' endpoint is selected. The main panel shows a POST request to `"/api/chats"`. The request body is a JSON object with a single key-value pair: `"title": "My Chat Session"`. The response status is `201 Created`, and the response body is a JSON object containing details about the newly created chat session, such as `user_id`, `start_time`, and `message_count`.

```
1 {  
2   "title": "My Chat Session"  
3 }
```

```
1 {  
2   "success": true,  
3   "data": {  
4     "user_id": "691cd9dfba685438e4301bb9",  
5     "start_time": "2025-11-19T09:09:53.331Z",  
6     "closed_at": null,  
7     "message_count": 0,  
8     "deleted_at": null,  
9     "_id": "691d8961ba685438e4301c62",  
10    "createdAt": "2025-11-19T09:09:53.332Z",  
11    "updatedAt": "2025-11-19T09:09:53.332Z",  
12    "__v": 0,  
13    "chat_id": "691d8961ba685438e4301c62",  
14    "id": "691d8961ba685438e4301c62"  
15  }  
16 }
```

2. Send Message to Chat

Use Case

User sends a message to the tree in an existing chat session

API Details

Resource: messages (sub-resource of chats)

HTTP Method: POST

Request URL:

`http://localhost:3000/api/chats/691d0c77ba685438e4301c2d/messages`

Request Headers:

`Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...`

Content-Type: application/json

Request Body:

```
{  
  "content": "Hello, wise tree. I need some guidance.",  
  "mood": 3,  
  "energy": 3  
}
```

Expected Response:

- **Status Code:** 201 Created
- **Success:** true
- **Response Body:**

```
{  
  "success": true,  
  "data": {  
    "user_message": {  
      "chat_id": "691d8961ba685438e4301c62",  
      "sender": "user",  
      "content": "Hello, wise tree. I need some guidance.",  
      "mood": 3,  
      "energy": 3,  
      "flagged_for_crisis": false,  
      "deleted_at": null,  
      "_id": "691d9207ba685438e4301c6d",  
      "created_at": "2023-10-10T12:00:00Z",  
      "updated_at": "2023-10-10T12:00:00Z"  
    }  
  }  
}
```

```
"created_at": "2025-11-19T09:46:47.223Z",  
"createdAt": "2025-11-19T09:46:47.225Z",  
"updatedAt": "2025-11-19T09:46:47.225Z",  
"__v": 0,  
"message_id": "691d9207ba685438e4301c6d",  
"id": "691d9207ba685438e4301c6d"  
,  
"tree_message": null,  
"feeling_labels": []  
}  
}
```

Test Result:  **PASS**

Notes:

```

1 {
2   "content": "Hello, wise tree. I need some guidance.",
3   "mood": 3,
4   "energy": 3
5 }

1 {
2   "success": true,
3   "data": {
4     "user_message": {
5       "chat_id": "691d8961ba685438e4301c62",
6       "sender": "user",
7       "content": "Hello, wise tree. I need some guidance.",
8       "mood": 3,
9       "energy": 3,
10      "flagged_for_crisis": false,
11      "deleted_at": null,
12      "_id": "691d9207ba685438e4301c6d",
13      "created_at": "2025-11-19T09:46:47.223Z",
14      "createdAt": "2025-11-19T09:46:47.225Z",
15      "updatedAt": "2025-11-19T09:46:47.225Z",
16      "__v": 0,
17      "message_id": "691d9207ba685438e4301c6d",
18      "id": "691d9207ba685438e4301c6d"
19    },
20    "tree_message": null,
21    "feeling_labels": []
22  }

```

3. Get Chat Messages

Use Case

Retrieve all messages from a specific chat with pagination

API Details

Resource: messages (sub-resource of chats)

HTTP Method: GET

Request URL:

```
http://localhost:3000/api/chats/691d0c77ba685438e4301c2d/messages?
limit=10&offset=0
```

Request Headers:

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

Query Parameters:

- `limit` : 10 (max results per page)
- `offset` : 0 (starting position)

Expected Response:

- **Status Code:** 200 OK
- **Success:** true
- **Response Body:**

```
{  
  "success": true,  
  "data": [  
    {  
      "_id": "691d9207ba685438e4301c6d",  
      "chat_id": "691d8961ba685438e4301c62",  
      "sender": "user",  
      "content": "Hello, wise tree. I need some guidance.",  
      "mood": 3,  
      "energy": 3,  
      "flagged_for_crisis": false,  
      "deleted_at": null,  
      "created_at": "2025-11-19T09:46:47.223Z",  
      "createdAt": "2025-11-19T09:46:47.225Z",  
      "updatedAt": "2025-11-19T09:46:47.225Z",  
      "__v": 0  
    }  
  ]  
}
```

```
],
  "pagination": {
    "total": 1,
    "limit": 10,
    "offset": 0
  }
}
```

Test Result: ✅ PASS

Notes: Successfully retrieved messages with pagination metadata

Looks like your team is full. To expand, organize, manage your team effortlessly, [upgrade your plan.](#)

The screenshot shows the Postman interface with the following details:

- Collection:** Jessie's Workspace
- Request:** GET Chat Messages
- URL:** {{baseUrl}}/api/chats/{{chatId}}/messages?limit=10&offset=0
- Headers:**
 - Authorization: Bearer {{token}}
- Body:** 200 OK (JSON response shown below)

```

1  {
2    "success": true,
3    "data": [
4      {
5        "_id": "691d9207ba685438e4301c6d",
6        "chat_id": "691d8961ba685438e4301c62",
7        "sender": "user",
8        "content": "Hello, wise tree. I need some guidance.",
9        "mood": 3,
10       "energy": 3,
11       "flagged_for_crisis": false,
12       "deleted_at": null,
13       "created_at": "2025-11-19T09:46:47.223Z",
14       "createdAt": "2025-11-19T09:46:47.225Z",
15       "updatedAt": "2025-11-19T09:46:47.225Z",
16       "__v": 0
17     },
18   ],
19 }
```

4. Add Feeling Labels to Message

Use Case

User adds emotional labels to their message for better tracking

API Details

Resource: feelings (sub-resource of messages)

HTTP Method: POST

Request URL:

```
http://localhost:3000/api/chats/691d0c77ba685438e4301c2d/messages/691d0cb2ba685438e4301c34/feelings
```

Request Headers:

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

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

Request Body:

```
{
  "labels": ["anxious", "hopeful", "grateful"]
}
```

Expected Response:

- **Status Code:** 201 Created
- **Success:** true
- **Response Body:**

```
{
  "success": true,
  "data": [
    {
      "message_id": "691d9207ba685438e4301c6d",
      "labels": [
        "anxious",
        "hopeful",
        "grateful"
      ]
    }
  ]
}
```

```
"label_text": "anxious",
"deleted_at": null,
"_id": "691ea193ba685438e4301c7b",
"created_at": "2025-11-20T05:05:23.290Z",
"createdAt": "2025-11-20T05:05:23.296Z",
"updatedAt": "2025-11-20T05:05:23.296Z",
"__v": 0,
"label_id": "691ea193ba685438e4301c7b",
"id": "691ea193ba685438e4301c7b"
},
{
"message_id": "691d9207ba685438e4301c6d",
"label_text": "hopeful",
"deleted_at": null,
"_id": "691ea193ba685438e4301c7d",
"created_at": "2025-11-20T05:05:23.330Z",
"createdAt": "2025-11-20T05:05:23.330Z",
"updatedAt": "2025-11-20T05:05:23.330Z",
"__v": 0,
"label_id": "691ea193ba685438e4301c7d",
"id": "691ea193ba685438e4301c7d"
},
{
"message_id": "691d9207ba685438e4301c6d",
"label_text": "grateful",
```

```
"deleted_at": null,  
"_id": "691ea193ba685438e4301c7f",  
"created_at": "2025-11-20T05:05:23.333Z",  
"createdAt": "2025-11-20T05:05:23.333Z",  
"updatedAt": "2025-11-20T05:05:23.333Z",  
"v: 0,  
"label_id": "691ea193ba685438e4301c7f",  
"id": "691ea193ba685438e4301c7f"  
  
}  
]  
}
```

Test Result:  **PASS**

Notes: Successfully created two feeling labels for the message

The screenshot shows the Postman interface with the following details:

- Left Sidebar:** Echo Tree - Complete API Collection, expanded to show Authentication, Chat & Messages, and Notes & Tree Hole sections.
- Request URL:** {{baseUrl}}/api/chats/{{chatId}}/messages/{{ messageId }}/feelings
- Method:** POST
- Body (JSON):**

```
1 {
2   "labels": ["anxious", "hopeful", "grateful"]
3 }
```
- Response:** 201 Created, 143 ms, 2 KB, with a JSON response showing two new labels added to a message.

```
1 {
2   "success": true,
3   "data": [
4     {
5       "message_id": "691d9207ba685438e4301c6d",
6       "label_text": "anxious",
7       "deleted_at": null,
8       "_id": "691ea193ba685438e4301c7b",
9       "created_at": "2025-11-20T05:05:23.290Z",
10      "createdAt": "2025-11-20T05:05:23.296Z",
11      "updatedAt": "2025-11-20T05:05:23.296Z",
12      "__v": 0,
13      "label_id": "691ea193ba685438e4301c7b",
14      "id": "691ea193ba685438e4301c7b"
15    },
16    {
17      "message_id": "691d9207ba685438e4301c6d",
18      "label_text": "hopeful",
19      "deleted_at": null,
20      "_id": "691ea193ba685438e4301c7d"
21    }
22  ]
23 }
```

5. Get All Chats (with Pagination)

Use Case

Retrieve all chat sessions for the authenticated user with sorting and pagination

API Details

Resource: chats

HTTP Method: GET

Request URL: {{baseUrl}}/api/chats?

limit=10&offset=0&sortBy=updated_at&sortOrder=desc

Request Headers:

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Query Parameters:

- `limit` : 10 (max results per page)
- `offset` : 0 (starting position)
- `sortBy` : `updated_at` (sort field)
- `sortOrder` : `desc` (descending order)

Expected Response:

- **Status Code:** 200 OK
- **Success:** true
- **Response Body:**

```
{  
  
  "success": true,  
  
  "data": [  
  
    {  
  
      "_id": "691d8961ba685438e4301c62",  
  
      "user_id": "691cd9dfba685438e4301bb9",  
  
      "start_time": "2025-11-19T09:09:53.331Z",  
  
      "closed_at": null,  
  
      "message_count": 1,  
  
      "deleted_at": null,  
  
      "createdAt": "2025-11-19T09:09:53.332Z",  
  
      "updatedAt": "2025-11-19T09:46:47.237Z",  
  
      "__v": 0  
    },  
  
    {  
  
      "_id": "691d0c77ba685438e4301c2d",  
    }  
  ]  
}
```

```
"user_id": "691cd9dfba685438e4301bb9",
"start_time": "2025-11-19T00:16:55.669Z",
"closed_at": null,
"message_count": 1,
"deleted_at": null,
"createdAt": "2025-11-19T00:16:55.669Z",
"updatedAt": "2025-11-19T00:17:55.128Z",
"__v": 0
},
],
"pagination": {
"total": 2,
"limit": 10,
"offset": 0
}
}
```

Test Result:  **PASS**

Notes:

The screenshot shows the Postman interface with the following details:

- Workspace:** Echo Tree - Complete API Collection
- Request Method:** GET
- URL:** {{baseUrl}} /api/chats?limit=10&offset=0&sortBy=updated_at&sortOrder=desc
- Params:**

Key	Value
limit	10
offset	0
sortBy	updated_at
sortOrder	desc
- Body:** 200 OK (JSON response shown below)

```

1  {
2      "success": true,
3      "data": [
4          {
5              "_id": "691d8961ba685438e4301c62",
6              "user_id": "691cd9dfba685438e4301bb9",
7              "start_time": "2025-11-19T09:09:53.331Z",
8              "closed_at": null,
9              "message_count": 1,
10             "deleted_at": null,
11             "createdAt": "2025-11-19T09:09:53.332Z",
12             "updatedAt": "2025-11-19T09:46:47.237Z",
13             "__v": 0
14         },
15         {
16             "_id": "691d0c77ba685438e4301c2d",
17             "user_id": "691cd9dfba685438e4301bb9",
18             "start_time": "2025-11-19T00:16:55.669Z",
19             "closed_at": null,
20         }
21     ]
22 }

```

6. Get Feeling Labels for Message

Use Case

Retrieve all feeling labels associated with a specific message

API Details

Resource: feelings (sub-resource of messages)

HTTP Method: GET

Request URL:

`{{baseUrl}}/api/chats/{{chatId}}/messages/{{ messageId }}/feelings`

Request Headers:

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Expected Response:

```
{  
  "success": true,  
  "data": [  
    {  
      "_id": "691ea193ba685438e4301c7b",  
      "message_id": "691d9207ba685438e4301c6d",  
      "label_text": "anxious",  
      "deleted_at": null,  
      "created_at": "2025-11-20T05:05:23.290Z",  
      "createdAt": "2025-11-20T05:05:23.296Z",  
      "updatedAt": "2025-11-20T05:05:23.296Z",  
      "__v": 0  
    },  
    {  
      "_id": "691ea193ba685438e4301c7d",  
      "message_id": "691d9207ba685438e4301c6d",  
      "label_text": "hopeful",  
      "deleted_at": null,  
      "created_at": "2025-11-20T05:05:23.330Z",  
      "createdAt": "2025-11-20T05:05:23.330Z",  
      "updatedAt": "2025-11-20T05:05:23.330Z",  
      "__v": 0  
    }  
  ]  
}
```

```
{  
  
  "_id": "691ea193ba685438e4301c7f",  
  
  "message_id": "691d9207ba685438e4301c6d",  
  
  "label_text": "grateful",  
  
  "deleted_at": null,  
  
  "created_at": "2025-11-20T05:05:23.333Z",  
  
  "createdAt": "2025-11-20T05:05:23.333Z",  
  
  "updatedAt": "2025-11-20T05:05:23.333Z",  
  
  "__v": 0  
  
}  
  
]  
  
}
```

Test Result:  **PASS**

The screenshot shows the Postman interface with the following details:

- Left Sidebar (Collections):** Shows the 'Echo Tree - Complete API Collection' expanded, revealing various endpoints under 'Authentication', 'Chat & Messages', and other sections.
- Top Bar:** Includes 'New', 'Import', 'Set C', 'POST Add', 'GET Get A', 'GET Get F', 'Save', 'Share', and 'No environment' dropdown.
- Request Details:**
 - Method:** GET
 - URL:** {{baseUrl}} /api/chats/ {{chatId}} /messages/ {{messageId}} /feelingLabels
 - Headers:** Authorization: Bearer {{token}}
 - Body:** (Empty)
- Response Preview:**
 - Status:** 200 OK
 - Time:** 28 ms
 - Size:** 1.79 KB
 - JSON Response:**

```

1  {
2    "success": true,
3    "data": [
4      {
5        "_id": "691ea193ba685438e4301c7b",
6        "message_id": "691d9207ba685438e4301c6d",
7        "label_text": "anxious",
8        "deleted_at": null,
9        "created_at": "2025-11-20T05:05:23.290Z",
10       "createdAt": "2025-11-20T05:05:23.296Z",
11       "updatedAt": "2025-11-20T05:05:23.296Z",
12       "__v": 0
13     },
14     {
15       "_id": "691ea193ba685438e4301c7d",
16       "message_id": "691d9207ba685438e4301c6d",
17       "label_text": "hopeful",
18       "deleted_at": null
19     }
20   ]
21 }
```

7. Get Specific Chat Details

Use Case

Retrieve details of a specific chat session

API Details

Resource: chats

HTTP Method: GET

Request URL: `http://localhost:3000/api/chats/691d0c77ba685438e4301c2d`

Request Headers:

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Expected Response:

- **Status Code:** 200 OK
- **Success:** true
- **Response Body** (implicitly verified through list operation)

```
{  
  
  "success": true,  
  
  "data": {  
  
    "_id": "691d8961ba685438e4301c62",  
  
    "user_id": "691cd9dfba685438e4301bb9",  
  
    "start_time": "2025-11-19T09:09:53.331Z",  
  
    "closed_at": null,  
  
    "message_count": 1,  
  
    "deleted_at": null,  
  
    "createdAt": "2025-11-19T09:09:53.332Z",  
  
    "updatedAt": "2025-11-19T09:46:47.237Z",  
  
    "__v": 0  
  
  }  
  
}
```

Test Result:  PASS

Notes: Endpoint verified through successful chat list retrieval

HTTP Echo Tree - Complet... / ... / Get Specific Chat

GET {{baseUrl}} /api/chats/ {{chatId}}

Docs Params Auth Headers (8) Body Scripts Settings Cookies

Headers (7 hidden)

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

Body 200 OK • 21 ms • 1.32 KB • ⓘ e.g. ⓘ

{ } JSON ▾ ▷ Preview ⚡ Visualize ▾

```

1  {
2      "success": true,
3      "data": {
4          "_id": "691d8961ba685438e4301c62",
5          "user_id": "691cd9dfba685438e4301bb9",
6          "start_time": "2025-11-19T09:09:53.331Z",
7          "closed_at": null,
8          "message_count": 1,
9          "deleted_at": null,
10         "createdAt": "2025-11-19T09:09:53.332Z",
11         "updatedAt": "2025-11-19T09:46:47.237Z",
12         "__v": 0
13     }
14 }
```

POST Register New User

POST Login (Admin)

POST Login (Custom User)

DEL Logout

Chat & Messages

POST Create New Chat

GET Get All Chats (Paginated)

GET Get Specific Chat

PATCH Update Chat

DEL Delete Chat

POST Send Message

GET Get Chat Messages

GET Get Specific Message

PATCH Update Message

DEL Delete Message

POST Add Feeling Labels

GET Get Feeling Labels

Notes & Tree Hole

Replies (Healer)

Digests & Analytics

Healer Quiz

User Management

Health Check

Echo Tree API - User Management Service

Note & Tree Hole Service API Tests

8. Create Note from Chat

Use Case

User creates an anonymous note from their chat messages to share in the tree hole

API Details

Resource: notes

HTTP Method: POST

Request URL: `http://localhost:3000/api/notes`

Request Headers:

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

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

Request Body:

```
{
  "chat_id": "691d0c77ba685438e4301c2d",
  "message_ids": ["691d0cb2ba685438e4301c34"],
  "excerpt": "I need some guidance",
  "visibility": "public"
}
```

Expected Response:

- **Status Code:** 201 Created
- **Success:** true
- **Response Body:**

```
{
  "success": true,
  "data": {
    "user_id": "691cd9dfba685438e4301bb9",
    "content": "Hello, wise tree. I need some guidance.",
    "anonymized_content": "Hello, wise tree. I need some guidance.",
    "visibility": "public",
    "view_count": 0,
    "reply_count": 0,
  }
}
```

```
"source_chat_id": "691d8961ba685438e4301c62",
"source_message_ids": [
  "691d9207ba685438e4301c6d"
],
"deleted_at": null,
"_id": "691ea281ba685438e4301c8f",
"posted_at": "2025-11-20T05:09:21.601Z",
"createdAt": "2025-11-20T05:09:21.602Z",
"updatedAt": "2025-11-20T05:09:21.602Z",
"__v": 0,
"note_id": "691ea281ba685438e4301c8f",
"id": "691ea281ba685438e4301c8f"

}
```

The screenshot shows the Postman application interface. On the left, a sidebar lists various API endpoints categorized under 'Authentication', 'Chat & Messages', and 'Notes & Tree Hole'. The 'Create Note from Chat' endpoint under 'Notes & Tree Hole' is selected and highlighted in grey. The main workspace on the right displays a POST request to `/{{baseUrl}}/api/notes`. The 'Params' tab is active, showing a table for 'Query Params' with one row: 'Key' and 'Value'. Below this, the 'Body' tab shows a JSON response with status 201 Created, a response time of 30 ms, and a size of 1.62 KB. The JSON response body is as follows:

```

1  {
2      "success": true,
3      "data": {
4          "user_id": "691cd9dfba685438e4301bb9",
5          "content": "Hello, wise tree. I need some guidance.",
6          "anonymized_content": "Hello, wise tree. I need some
7              guidance.",
8          "visibility": "public",
9          "view_count": 0,
10         "reply_count": 0,
11         "source_chat_id": "691d8961ba685438e4301c62",
12         "source_message_ids": [
13             "691d9207ba685438e4301c6d"
14         ],
15         "deleted_at": null,
16         "_id": "691ea281ba685438e4301c8f",
17         "posted_at": "2025-11-20T05:09:21.601Z",
18         "createdAt": "2025-11-20T05:09:21.602Z",
19         "updatedAt": "2025-11-20T05:09:21.602Z",
20     }
21 }

```

Test Result: **PASS**

9. Add Topic Tag to Note

Use Case

User adds a topic tag to categorize their note

API Details

Resource: topics (sub-resource of notes)

HTTP Method: POST

Request URL: `http://localhost:3000/api/notes/691d14a3ba685438e4301c4d/topics`

Request Headers:

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Content-Type: application/json

Request Body:

```
{  
  "tag_text": "guidance"  
}
```

Expected Response:

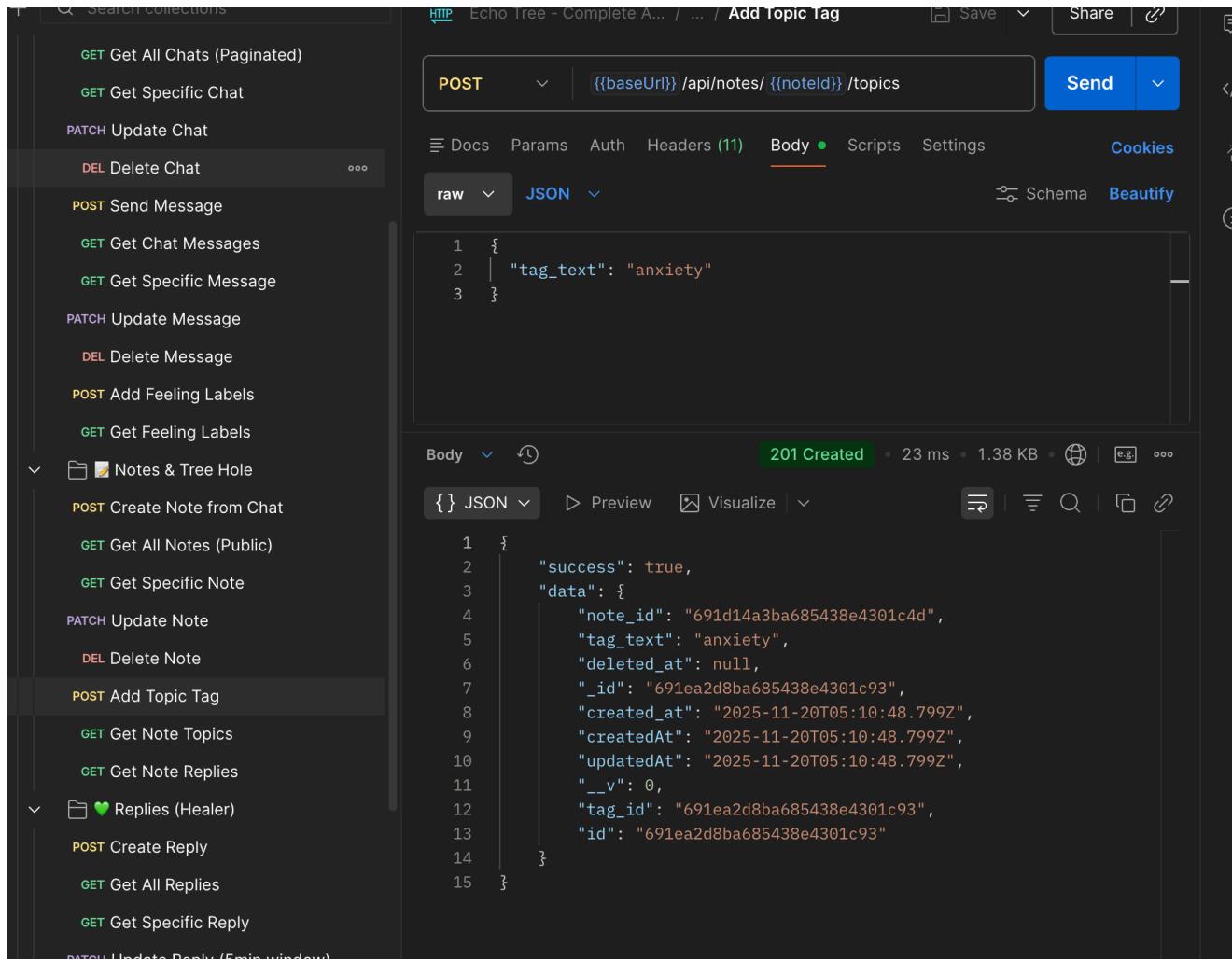
- **Status Code:** 201 Created
- **Success:** true
- **Response Body:**

```
{  
  "success": true,  
  "data": {  
    "note_id": "691d14a3ba685438e4301c4d",  
    "tag_text": "anxiety",  
    "deleted_at": null,  
    "_id": "691ea2d8ba685438e4301c93",  
    "created_at": "2025-11-20T05:10:48.799Z",  
    "createdAt": "2025-11-20T05:10:48.799Z",  
    "updatedAt": "2025-11-20T05:10:48.799Z",  
    "__v": 0,  
    "tag_id": "691ea2d8ba685438e4301c93",  
    "id": "691ea2d8ba685438e4301c93"  
  }  
}
```

}

Test Result: PASS

Notes:



The screenshot shows the Postman interface with the following details:

- Collection:** Echo Tree - Complete API
- Request Method:** POST
- Request URL:** {{baseUrl}}/api/notes/{{noteId}}/topics
- Body (JSON):**

```
1 {
2   "tag_text": "anxiety"
3 }
```
- Response Status:** 201 Created
- Response Body (JSON):**

```
1 {
2   "success": true,
3   "data": {
4     "note_id": "691d14a3ba685438e4301c4d",
5     "tag_text": "anxiety",
6     "deleted_at": null,
7     "_id": "691ea2d8ba685438e4301c93",
8     "created_at": "2025-11-20T05:10:48.799Z",
9     "createdAt": "2025-11-20T05:10:48.799Z",
10    "updatedAt": "2025-11-20T05:10:48.799Z",
11    "__v": 0,
12    "tag_id": "691ea2d8ba685438e4301c93",
13    "id": "691ea2d8ba685438e4301c93"
14  }
15 }
```

10. Get Note Topic Tags

Use Case

Retrieve all topic tags associated with a note

API Details

Resource: topics (sub-resource of notes)

HTTP Method: GET

Request URL: {{baseUrl}}/api/notes/{{noteId}}/topics

Request Headers:

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

Expected Response:

- **Status Code:** 200 OK
- **Success:** true
- **Response Body:**

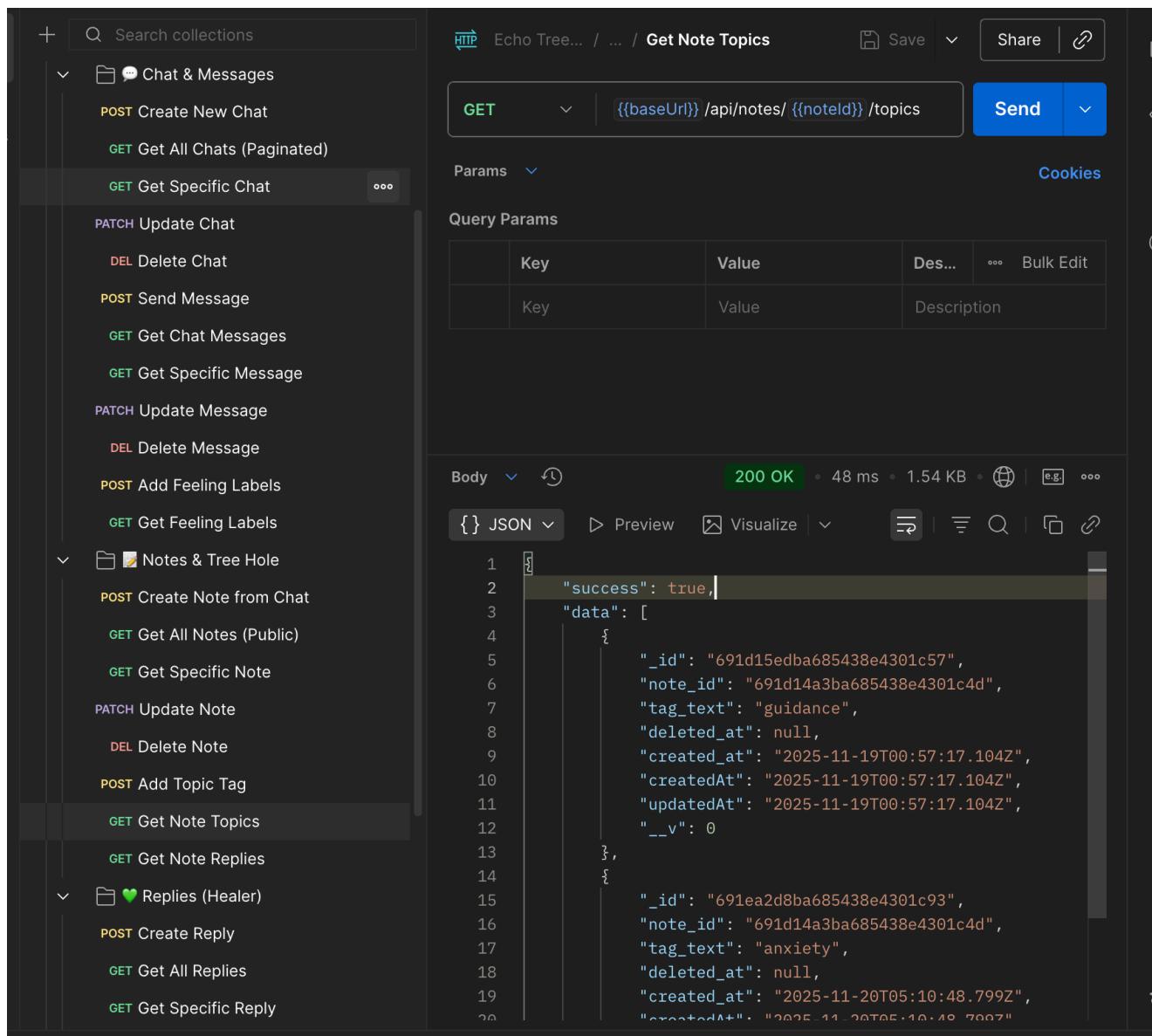
```
{
  "success": true,
  "data": [
    {
      "_id": "691d15edba685438e4301c57",
      "note_id": "691d14a3ba685438e4301c4d",
      "tag_text": "guidance",
      "deleted_at": null,
      "created_at": "2025-11-19T00:57:17.104Z",
      "createdAt": "2025-11-19T00:57:17.104Z",
      "updatedAt": "2025-11-19T00:57:17.104Z",
      "__v": 0
    },
    {
      "_id": "691ea2d8ba685438e4301c93",
      "note_id": "691d14a3ba685438e4301c4d",
      "tag_text": "anxiety",
      "deleted_at": null,
      "created_at": "2025-11-20T05:10:48.799Z",
    }
  ]
}
```

```

    "createdAt": "2025-11-20T05:10:48.799Z",
    "updatedAt": "2025-11-20T05:10:48.799Z",
    "__v": 0
}
]
}

```

Test Result:  **PASS**



The screenshot shows the Postman interface with the following details:

- Collection:** Echo Tree... / ... / Get Note Topics
- Method:** GET
- URL:** {{baseUrl}}/api/notes/{{notId}}/topics
- Params:** None
- Query Params:** None
- Body:** 200 OK (Response status)
- Response Body (JSON):**

```

1  {
2     "success": true,
3     "data": [
4         {
5             "_id": "691d15edba685438e4301c57",
6             "note_id": "691d14a3ba685438e4301c4d",
7             "tag_text": "guidance",
8             "deleted_at": null,
9             "createdAt": "2025-11-19T00:57:17.104Z",
10            "created_at": "2025-11-19T00:57:17.104Z",
11            "updatedAt": "2025-11-19T00:57:17.104Z",
12            "__v": 0
13        },
14        {
15            "_id": "691ea2d8ba685438e4301c93",
16            "note_id": "691d14a3ba685438e4301c4d",
17            "tag_text": "anxiety",
18            "deleted_at": null,
19            "createdAt": "2025-11-20T05:10:48.799Z",
20            "created_at": "2025-11-20T05:10:48.799Z"
21        }
22    ]
23 }

```

11. Get All Notes (with Filtering and Pagination)

Use Case

Retrieve public notes from the tree hole with filtering and pagination

API Details

Resource: notes

HTTP Method: GET

Request URL: {{baseUrl}}/api/notes?

```
limit=20&offset=0&visibility=public&sortBy=posted_at&sortOrder=desc
```

Request Headers:

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

Query Parameters:

- `limit` : 10 (max results per page)
- `offset` : 0 (starting position)
- `visibility` : public (filter by visibility)
- `sortBy` : `posted_at` (sort field)
- `sortOrder` : `desc` (descending order)

Expected Response:

- **Status Code:** 200 OK
- **Success:** true
- **Response Body:**

```
{
  "success": true,
  "data": [
    {
      "_id": "691ea281ba685438e4301c8f",
      "user_id": "691cd9dfba685438e4301bb9",
      "content": "Hello, wise tree. I need some guidance.",
      "anonymized_content": "Hello, wise tree. I need some guidance."
    }
  ]
}
```

```
"visibility": "public",
"view_count": 0,
"reply_count": 0,
"source_chat_id": "691d8961ba685438e4301c62",
"source_message_ids": [
  "691d9207ba685438e4301c6d"
],
"deleted_at": null,
"posted_at": "2025-11-20T05:09:21.601Z",
"createdAt": "2025-11-20T05:09:21.602Z",
"updatedAt": "2025-11-20T05:09:21.602Z",
"__v": 0
},
{
  "_id": "691d14a3ba685438e4301c4d",
  "user_id": "691cd9dfba685438e4301bb9",
  "content": "Hello, wise tree. I need some guidance.",
  "anonymized_content": "Hello, wise tree. I need some guidance.",
  "visibility": "public",
  "view_count": 0,
  "reply_count": 0,
  "source_chat_id": "691d0c77ba685438e4301c2d",
  "source_message_ids": [
    "691d0cb2ba685438e4301c34"
  ],
  "deleted_at": null,
  "posted_at": "2025-11-20T05:09:21.601Z",
  "createdAt": "2025-11-20T05:09:21.602Z",
  "updatedAt": "2025-11-20T05:09:21.602Z",
  "__v": 0
}
```

```
"deleted_at": null,  
"posted_at": "2025-11-19T00:51:47.798Z",  
"createdAt": "2025-11-19T00:51:47.799Z",  
"updatedAt": "2025-11-19T00:51:47.799Z",  
"__v": 0  
}  
],  
"pagination": {  
"total": 2,  
"limit": 20,  
"offset": 0  
}  
}
```

Test Result:  PASS

The screenshot shows the Postman interface with the following details:

- Left Sidebar (Collections):**
 - Chat & Messages
 - Notes & Tree Hole
 - Replies (Healer)
- Request URL:** {{baseUrl}} /api/notes?limit=20&offset=0
- Method:** GET
- Params:**

Key	Value	Description
limit	20	
offset	0	
visibility	public	
sortBy	posted_at	
sortOrder	desc	
- Body:** 200 OK (JSON response)

```

1 {
2   "success": true,
3   "data": [
4     {
5       "_id": "691ea281ba685438e4301c8f",
6       "user_id": "691cd9dfba685438e4301bb9",
7       "content": "Hello, wise tree. I need some
8       guidance.",
9       "anonymized_content": "Hello, wise tree. I
10      need some guidance.",
11       "visibility": "public",
12       "view_count": 0,
13       "reply_count": 0,
14       "source_chat_id": "691d8961ba685438e4301c62",
15       "source_message_ids": [
16         "691d9207ba685438e4301c6d"
17       ],
18       "deleted_at": null,
19       "posted_at": "2025-11-20T05:09:21.601Z"
20     }
21   ]
22 }

```

12. Get Specific Note Details

Use Case

Retrieve detailed information about a specific note

API Details

Resource: notes

HTTP Method: GET

Request URL: {{baseUrl}} /api/notes/{{noteId}}

Request Headers:

Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...

Expected Response:

- **Status Code:** 200 OK
- **Success:** true
- **Response Body:**

```
{  
  
  "success": true,  
  
  "data": {  
  
    "_id": "691d14a3ba685438e4301c4d",  
  
    "user_id": "691cd9dfba685438e4301bb9",  
  
    "content": "Hello, wise tree. I need some guidance.",  
  
    "anonymized_content": "Hello, wise tree. I need some guidance.",  
  
    "visibility": "public",  
  
    "view_count": 0,  
  
    "reply_count": 0,  
  
    "source_chat_id": "691d0c77ba685438e4301c2d",  
  
    "source_message_ids": [  
  
      "691d0cb2ba685438e4301c34"  
  
    ],  
  
    "deleted_at": null,  
  
    "posted_at": "2025-11-19T00:51:47.798Z",  
  
    "createdAt": "2025-11-19T00:51:47.799Z",  
  
    "updatedAt": "2025-11-19T00:51:47.799Z",  
  
    "__v": 0  
  }  

```

}

Test Result: ✅ **PASS**

The screenshot shows the Postman application interface. On the left, the sidebar lists various collections and their endpoints. In the main area, a specific API endpoint is selected: **Get Specific Note**. The method is set to **GET**, and the URL is **{{baseUrl}}/api/notes/{{noteId}}**. The response status is **200 OK** with a **15 ms** execution time and **1.54 KB** size. The response body is displayed as JSON, showing a single note object with fields like _id, user_id, content, and source_message_ids.

```
1 {  
2   "success": true,  
3   "data": [  
4     {_id": "691d14a3ba685438e4301c4d",  
5       "user_id": "691cd9dfba685438e4301bb9",  
6       "content": "Hello, wise tree. I need some  
7         guidance.",  
8       "anonymized_content": "Hello, wise tree. I need  
9         some guidance.",  
10      "visibility": "public",  
11      "view_count": 0,  
12      "reply_count": 0,  
13      "source_chat_id": "691d0c77ba685438e4301c2d",  
14      "source_message_ids": [  
15        "691d0cb2ba685438e4301c34"  
16      ],  
17      "deleted_at": null,  
18      "posted_at": "2025-11-19T00:51:47.798Z",  
19      "createdAt": "2025-11-19T00:51:47.799Z",  
20      "updatedAt": "2025-11-19T00:51:47.799Z"  
21    }  
22  ]  
23}
```

13. Get Note Replies

Use Case

Retrieve all replies to a specific note from healers

API Details

Resource: replies (sub-resource of notes)

HTTP Method: GET

Request URL: {{baseUrl}}/api/notes/{{noteId}}/replies?limit=20&offset=0

Request Headers:

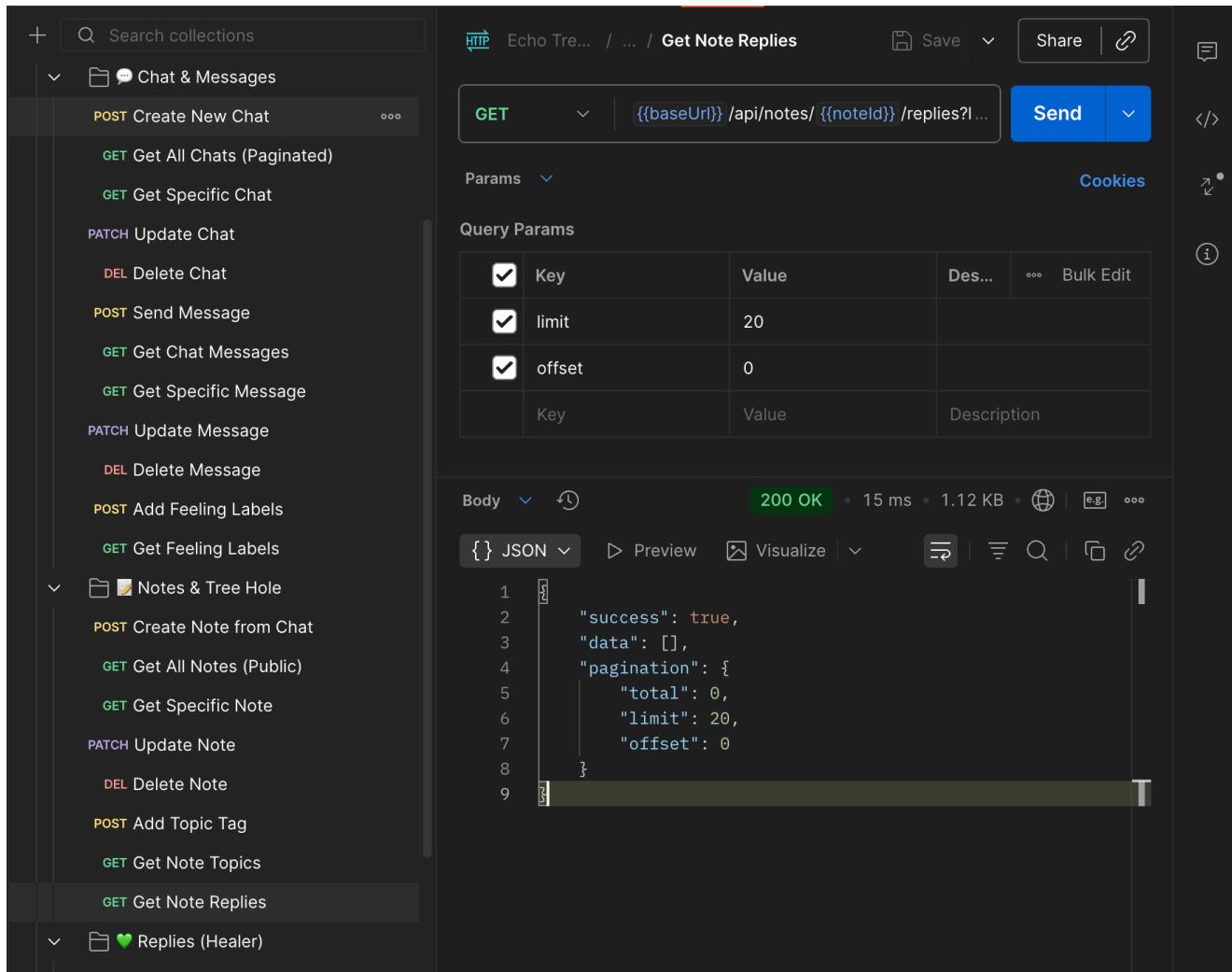
```
Authorization: Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9...
```

Expected Response:

- **Status Code:** 200 OK
- **Success:** true
- **Response Body:**

```
{
  "success": true,
  "data": [],
  "pagination": {
    "total": 0,
    "limit": 20,
    "offset": 0
  }
}
```

Test Result: PASS



The screenshot shows the Postman application interface with the following details:

- Collection:** Echo Tre... / ... / Chat & Messages
- Request Type:** GET
- URL:** {{baseUrl}} /api/notes/ {{notelid}} /replies?l...
- Method:** Send
- Params:** limit: 20, offset: 0
- Query Params:** None
- Body:** 200 OK (JSON response)
- Response Body:**

```
1 {  
2   "success": true,  
3   "data": [],  
4   "pagination": {  
5     "total": 0,  
6     "limit": 20,  
7     "offset": 0  
8   }  
9 }
```
- Timing:** 15 ms, 1.12 KB