Aegis Secure – API Testing

IT314 - Software Engineering

Group - 35

❖ Introduction

This report documents the API testing carried out on the Aegis Secure Backend, hosted at:

Base URL: https://aegis14211-aegissecurebackend.hf.space

Cyber URL: https://akshatbhatt515334-aegis-secure-api.hf.space

The purpose of this testing is to ensure that the backend APIs used in the Aegis Secure (Android application) are functioning correctly, securely and consistently. Testing was performed using Postman. All the positive and negative (edge) cases are considered in the testing.

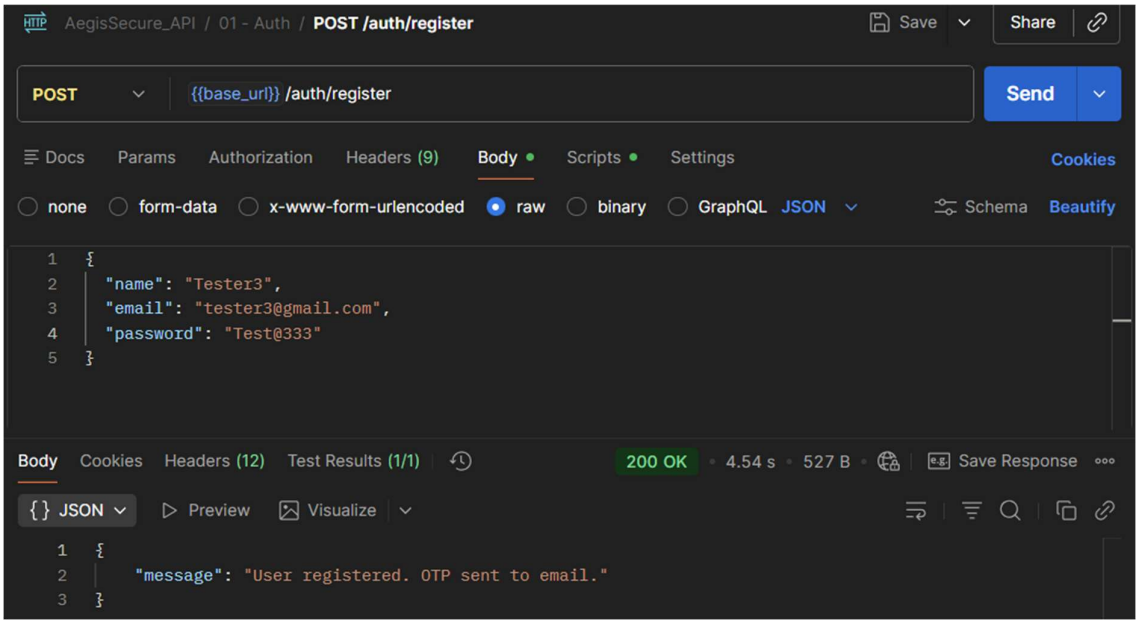
❖ Test Cases

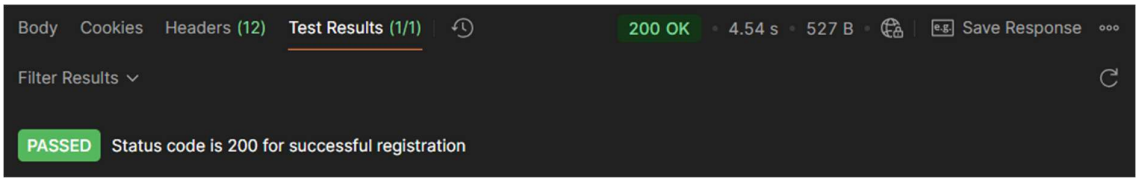
➢ TC-01 : Register New User (Positive)

Description:

Test registering a new user with a new email to see if the system accepts it.

Output:





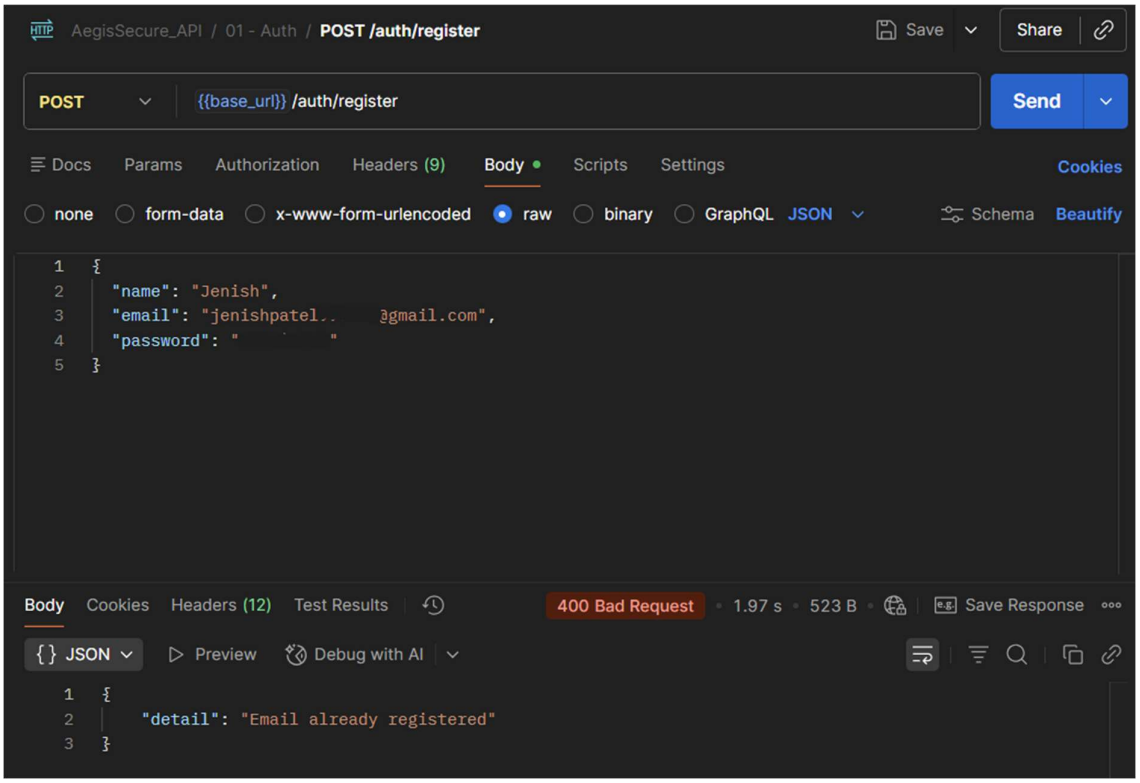
Verdict: PASS

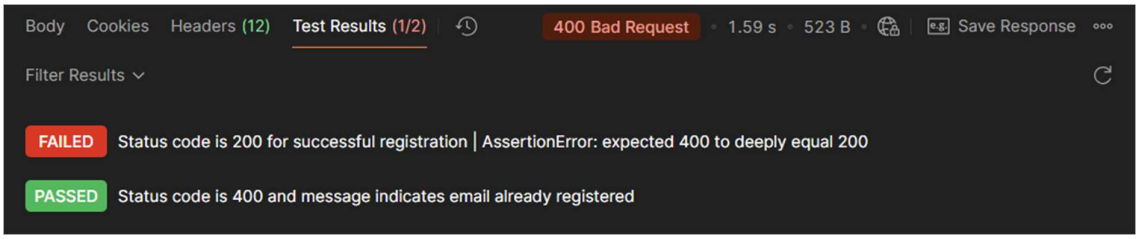
The API accepted the new email and created the user successfully.

➢ TC-02 : Register With Existing Email (Negative)

Description:
Try registering with the existing email to check if the system blocks it.

Output:





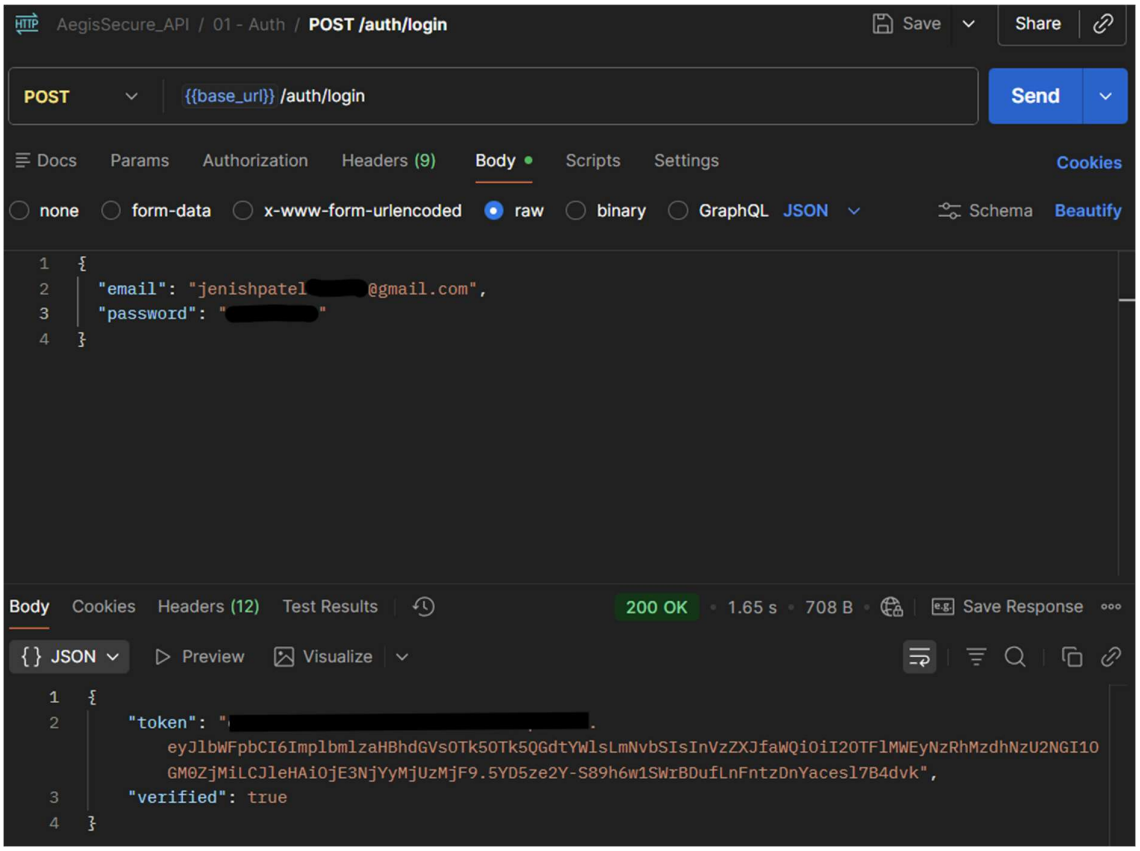
Verdict: PASS

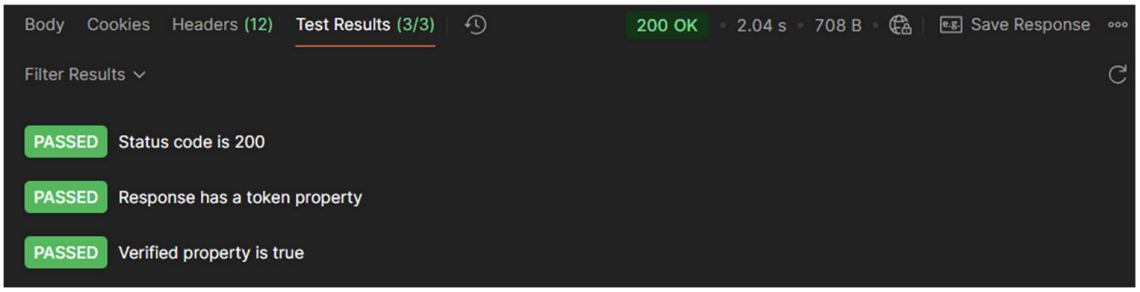
The API blocked the registration because the email was already used.

➢ TC-03 : Login with Valid Credentials (Positive)

Description:
Log in using correct email and password to check if login works.

Output:





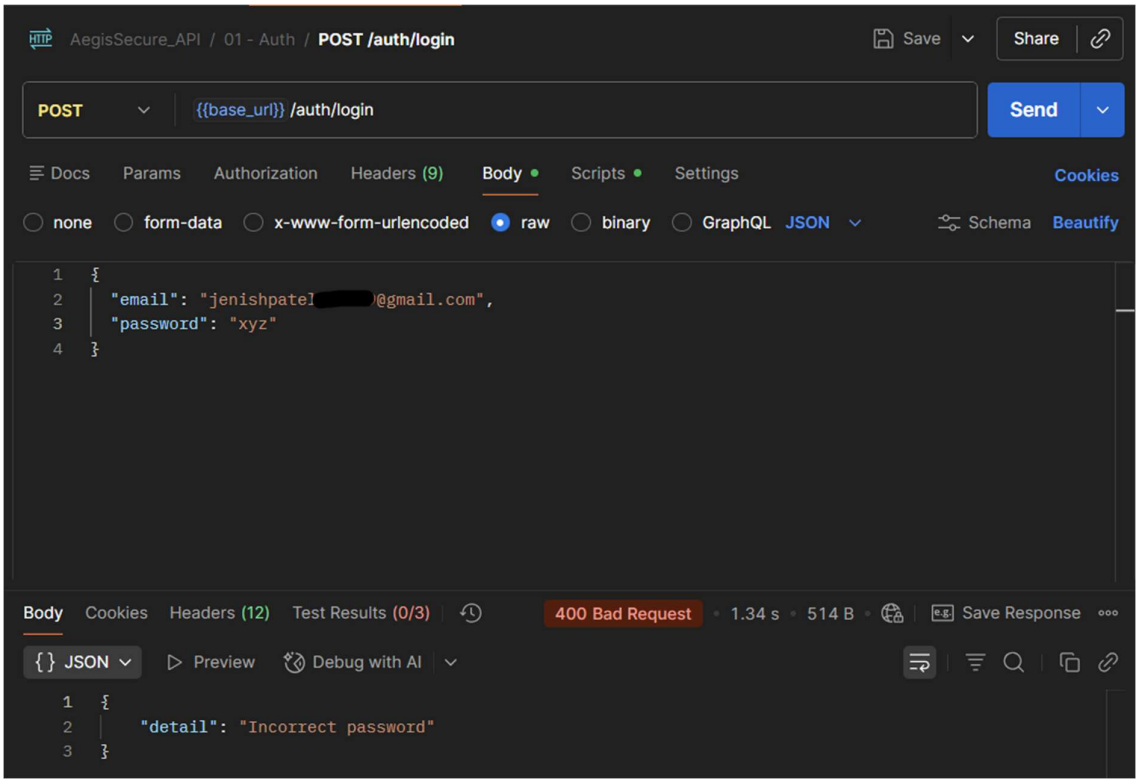
Verdict: PASS

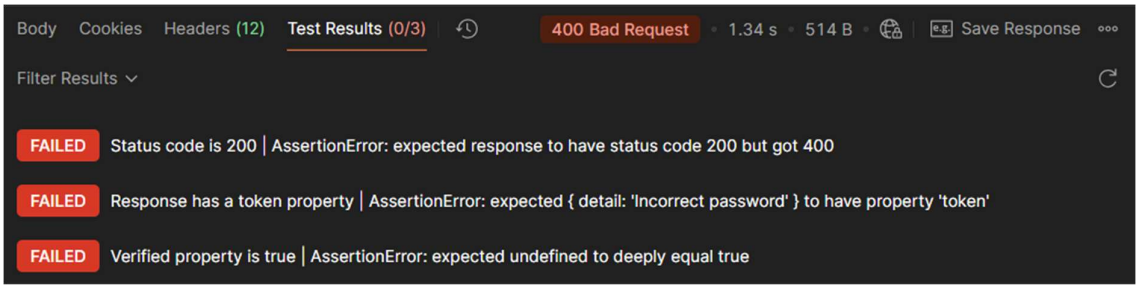
The API logged in the user and returned a valid token.

➢ TC-04 : Login With Wrong Password (Negative)

Description:
Try logging in with a correct email but wrong password.

Output:





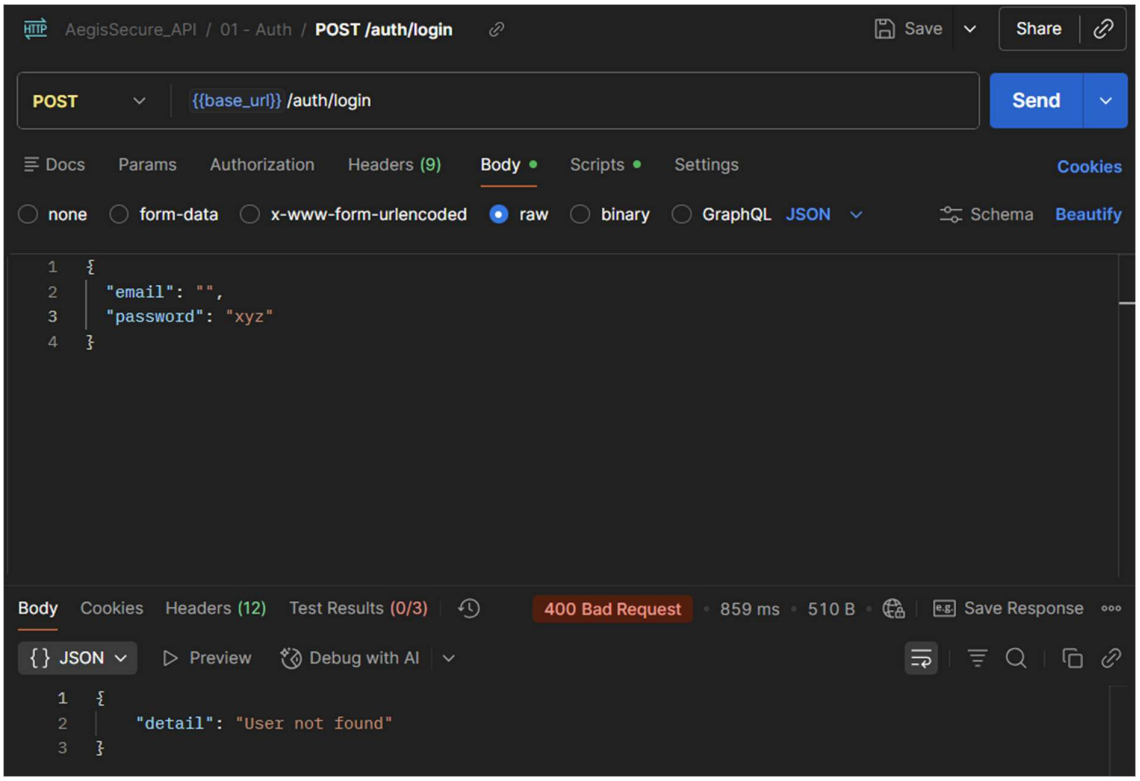
Verdict: PASS

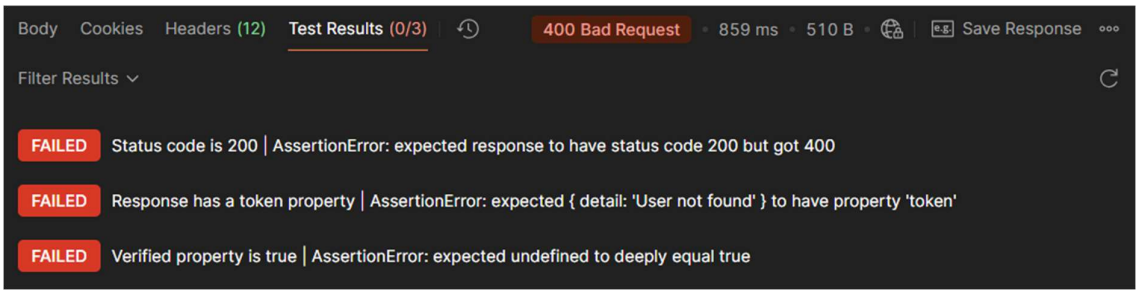
The API rejected the login attempt because the password was wrong.

➢ TC-05 : Login With Missing Fields (Negative)

Description:
Try logging in without entering email or password.

Output:





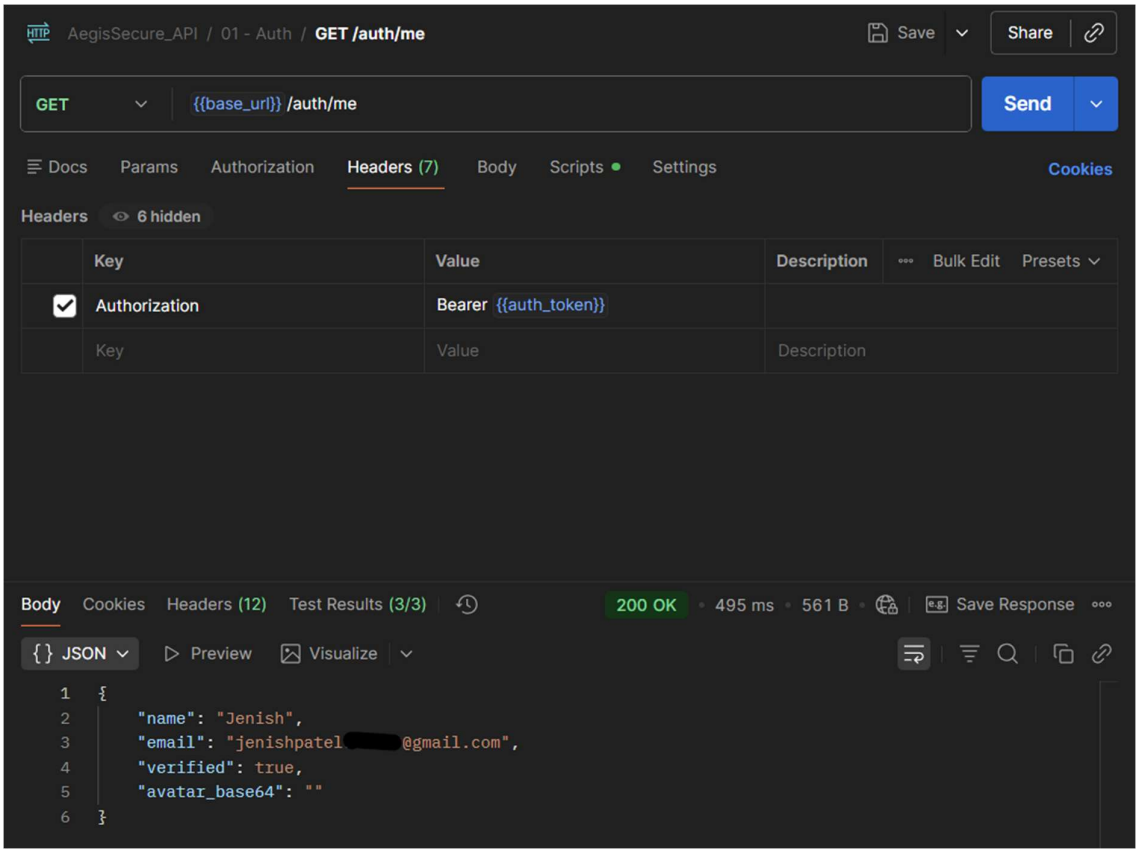
Verdict: PASS

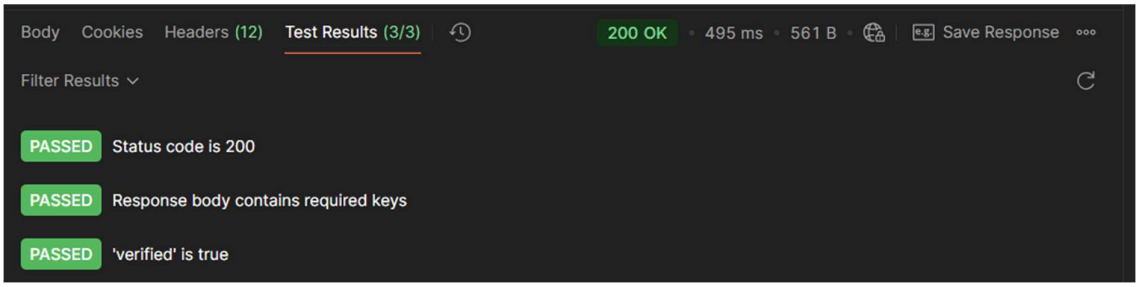
The API gave an error for missing details.

➢ TC-06 : Get User Details with Valid Token (Positive)

Description:
Call /auth/me with a valid token to get user details.

Output:





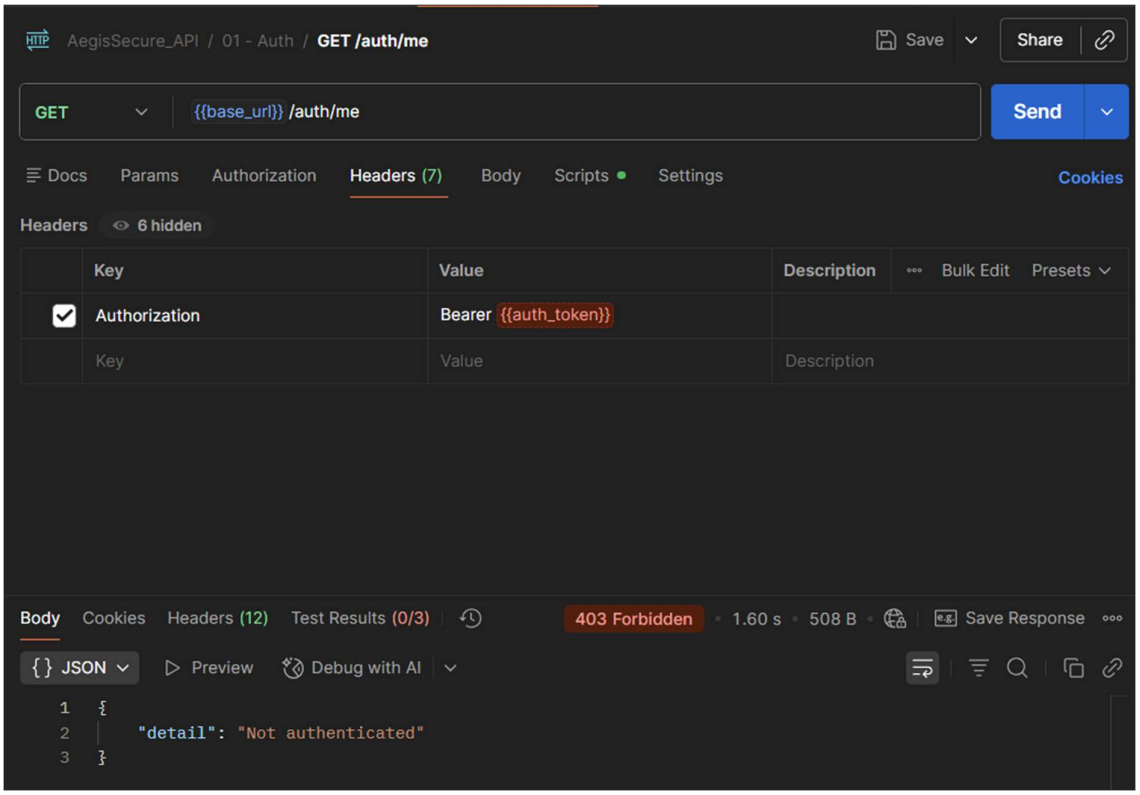
Verdict: PASS

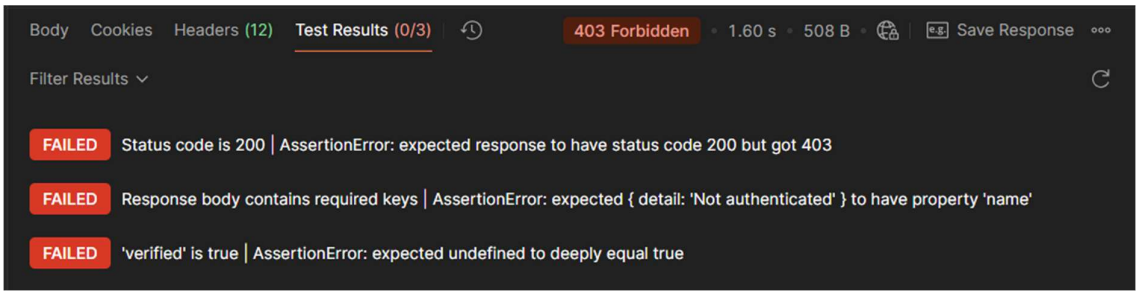
The API returned the correct user information.

➢ TC-07 : Get User Details Without Token (Negative)

Description:
Call /auth/me without sending any token.

Output:





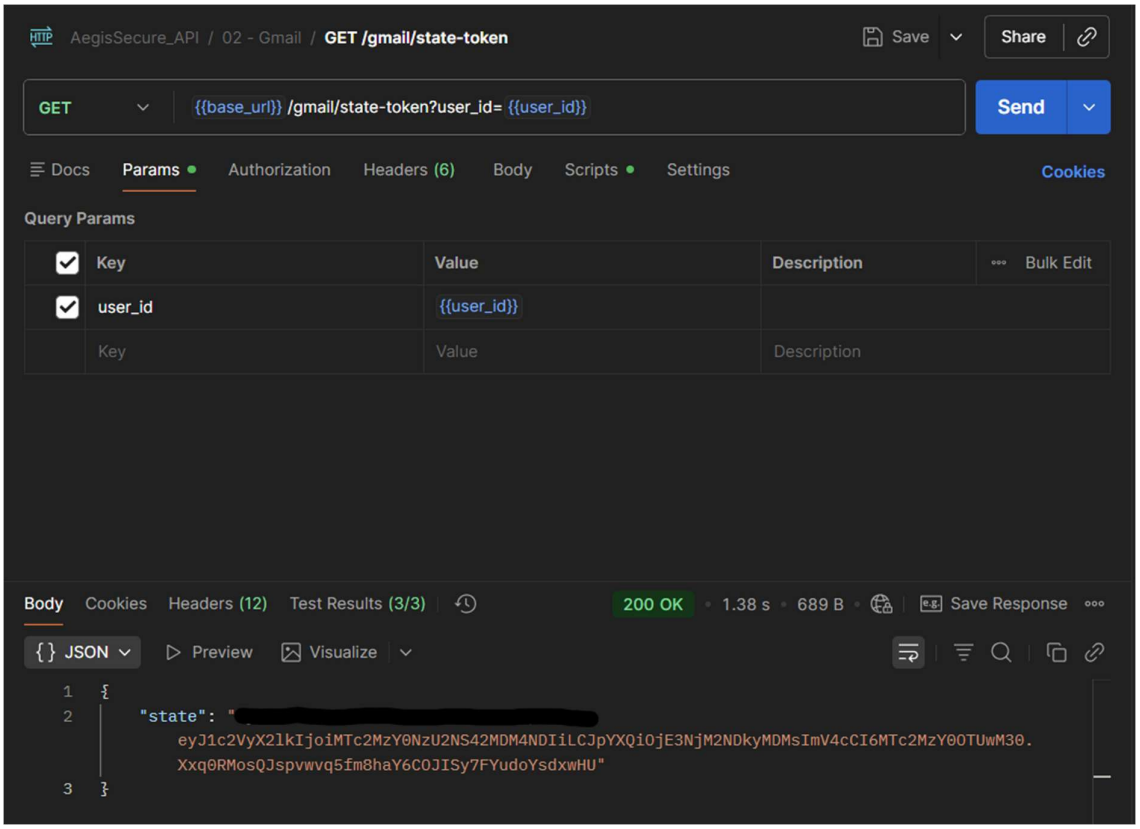
Verdict: PASS

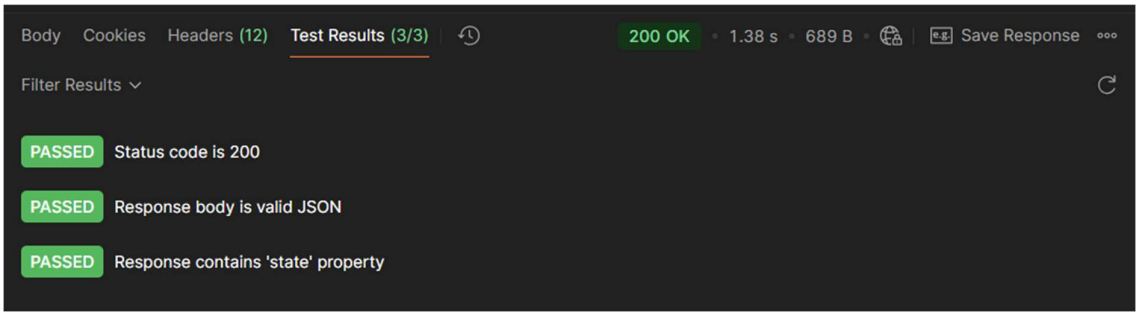
The API blocked the request because no token was provided.

➢ TC-08 : Get Gmail State Token (Positive)

Description:
Check if Gmail OAuth state token is created properly.

Output:





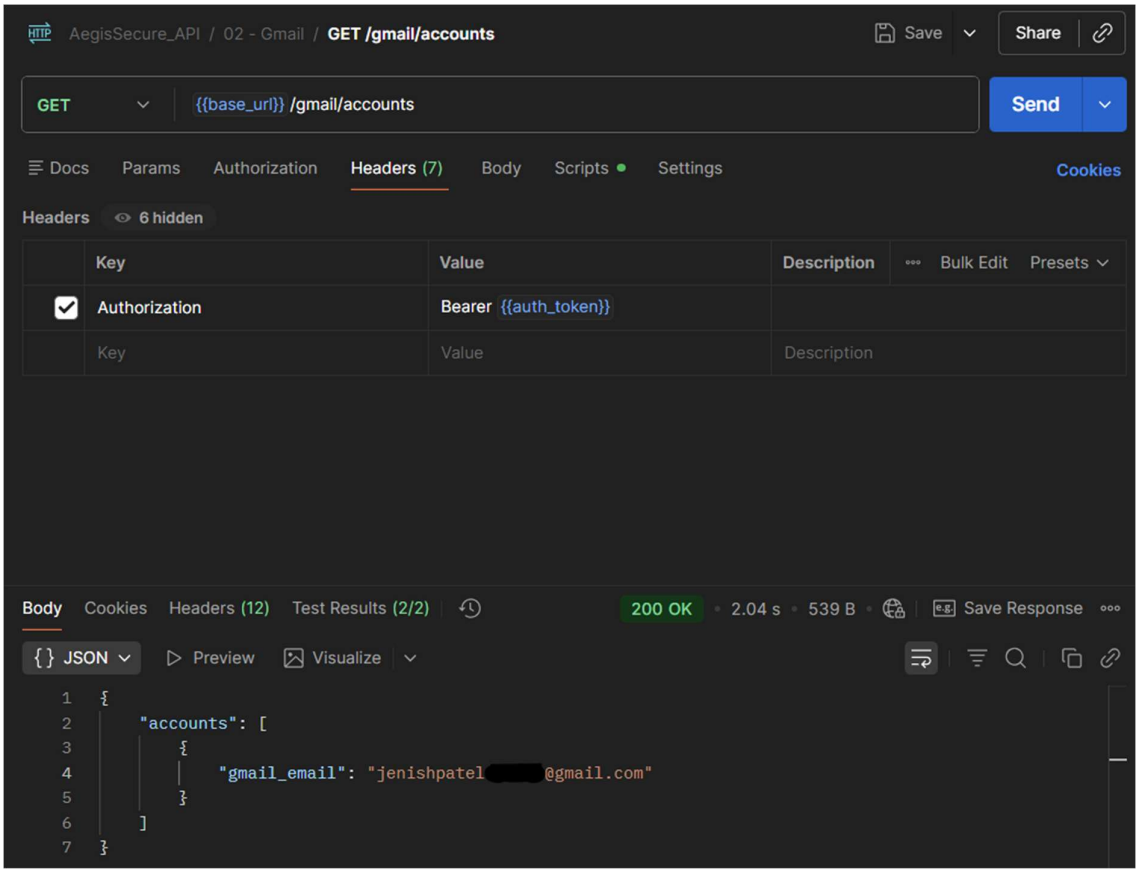
Verdict: PASS

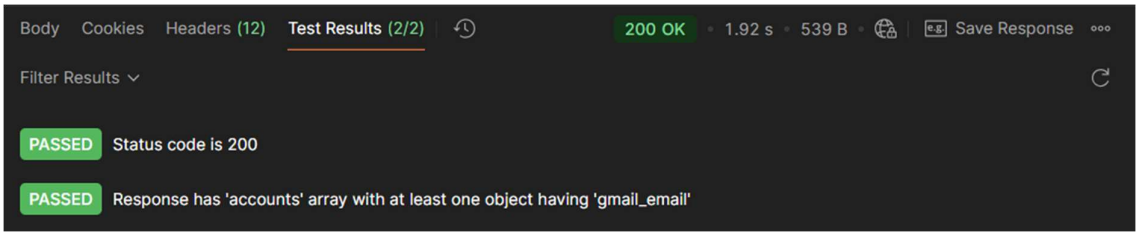
The API generated the state token successfully.

➢ TC-09 : Get Gmail Accounts (Positive)

Description:
Fetch all Gmail accounts connected to the user.

Output:





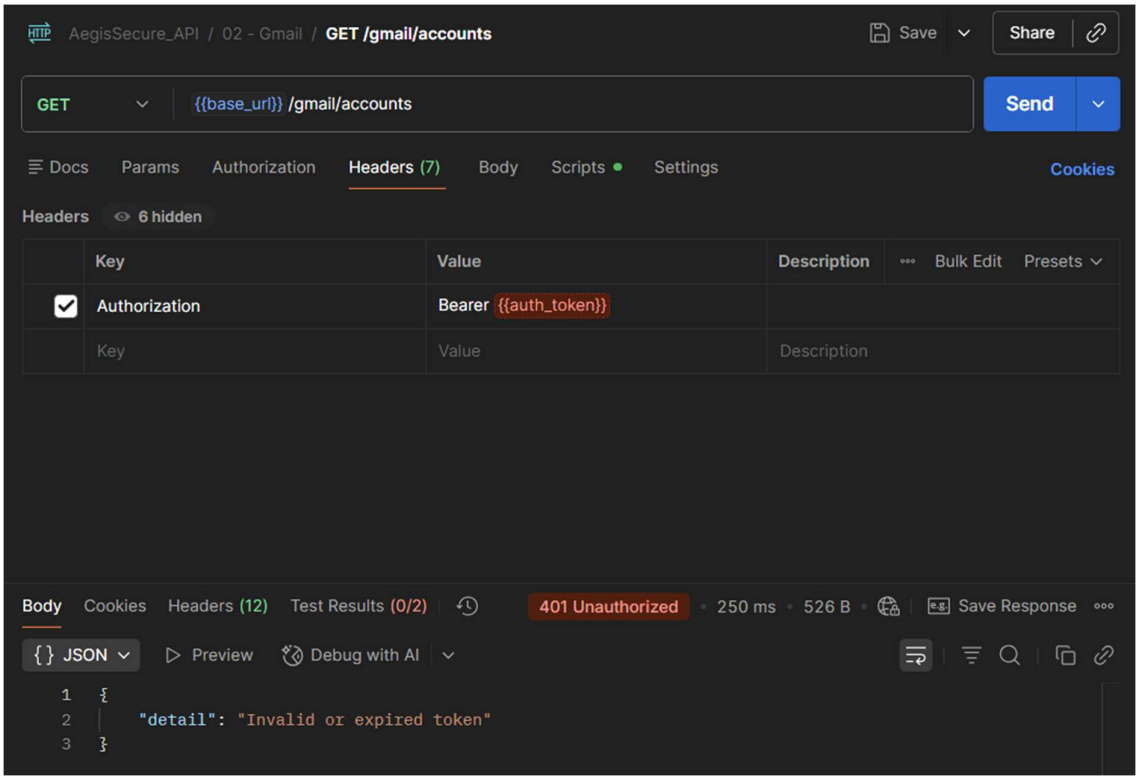
Verdict: PASS

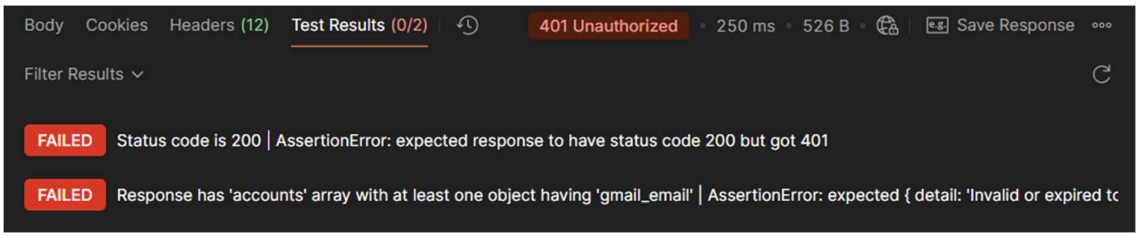
The API returned the connected Gmail accounts correctly.

➢ TC-10 : Get Gmail Accounts Without Token (Negative)

Description:
Try getting Gmail accounts without sending a token.

Output:





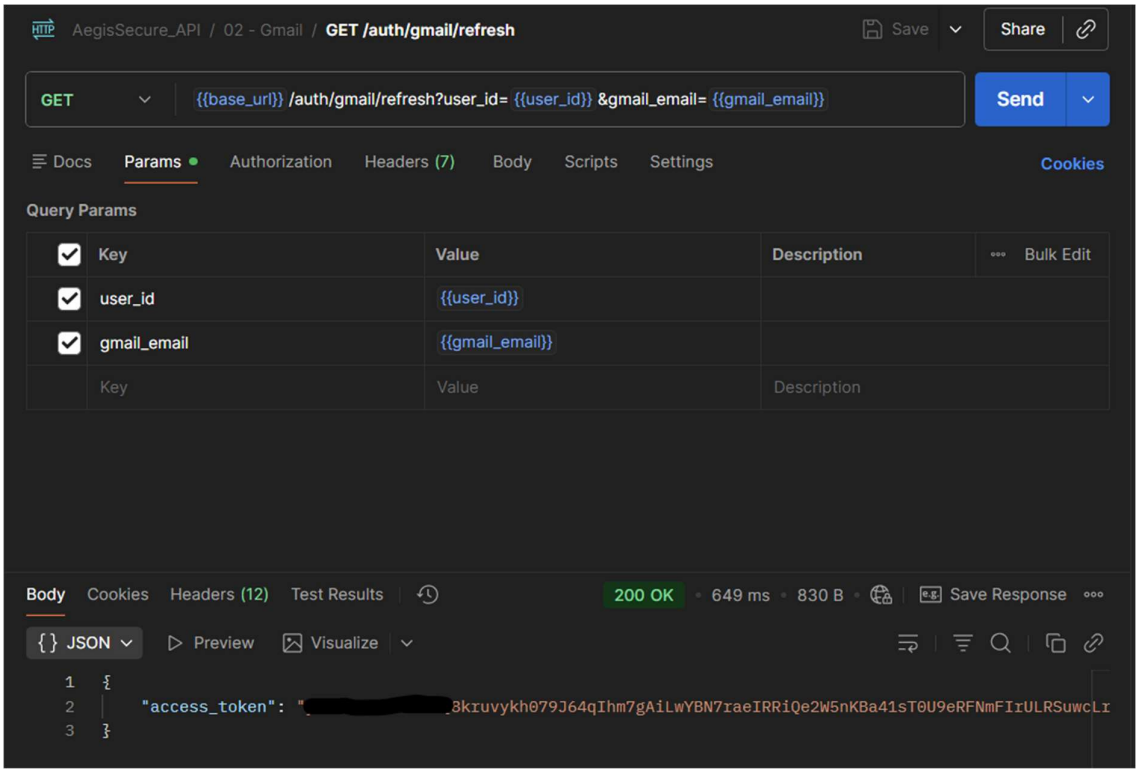
Verdict: PASS

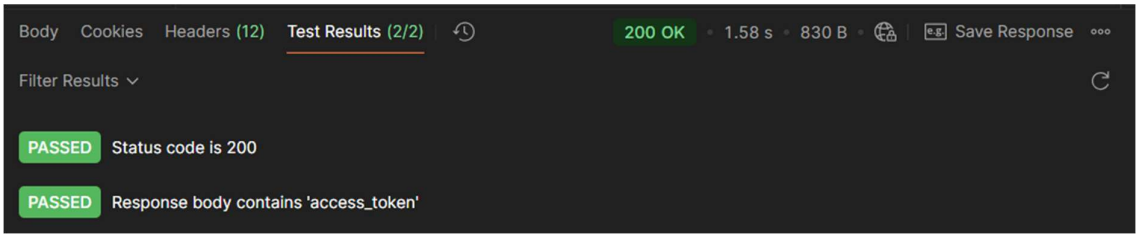
The API refused the request because the token was missing.

➢ TC-11 : Refresh Gmail Token (Positive)

Description:
Refresh Gmail access token using the refresh endpoint.

Output:





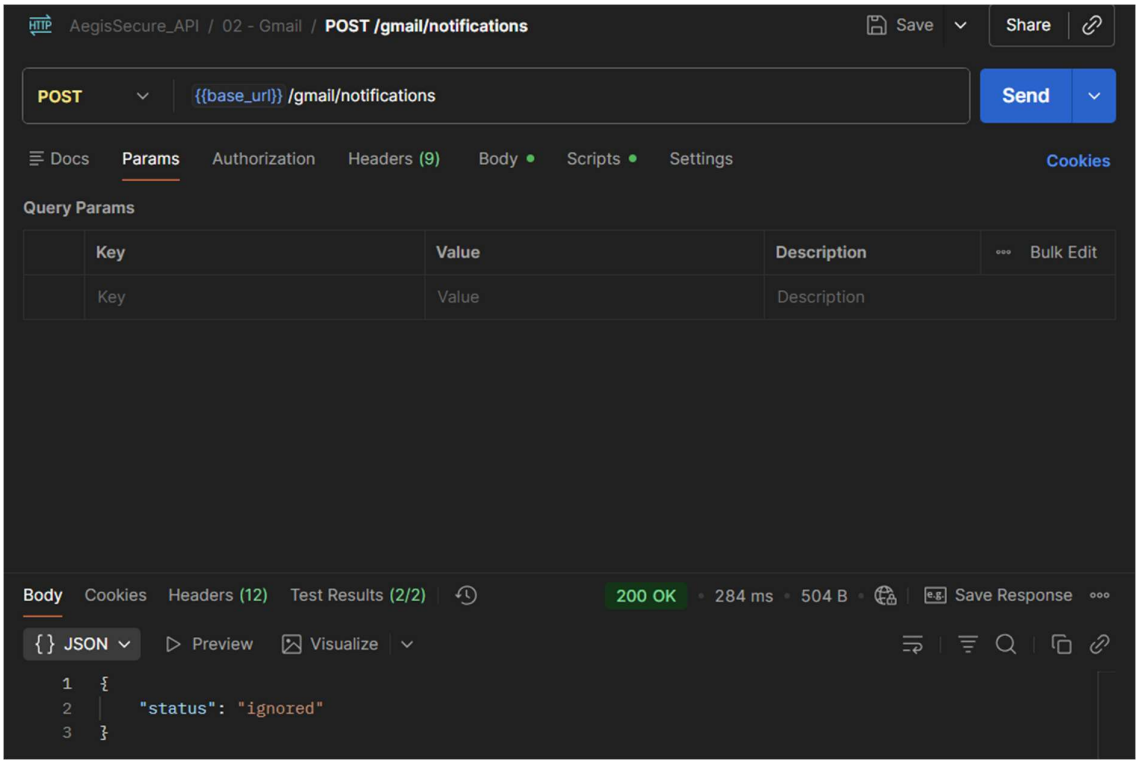
Verdict: PASS

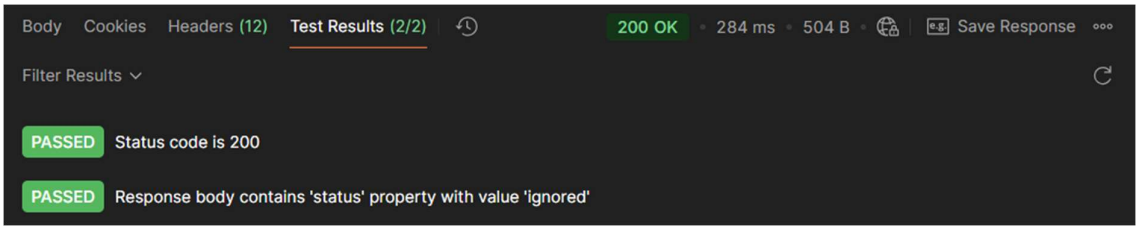
The token was refreshed successfully.

➢ TC-12 : Gmail Notifications Endpoint (Positive)

Description:
Send a sample request to the Gmail notification endpoint to see if it works.

Output:





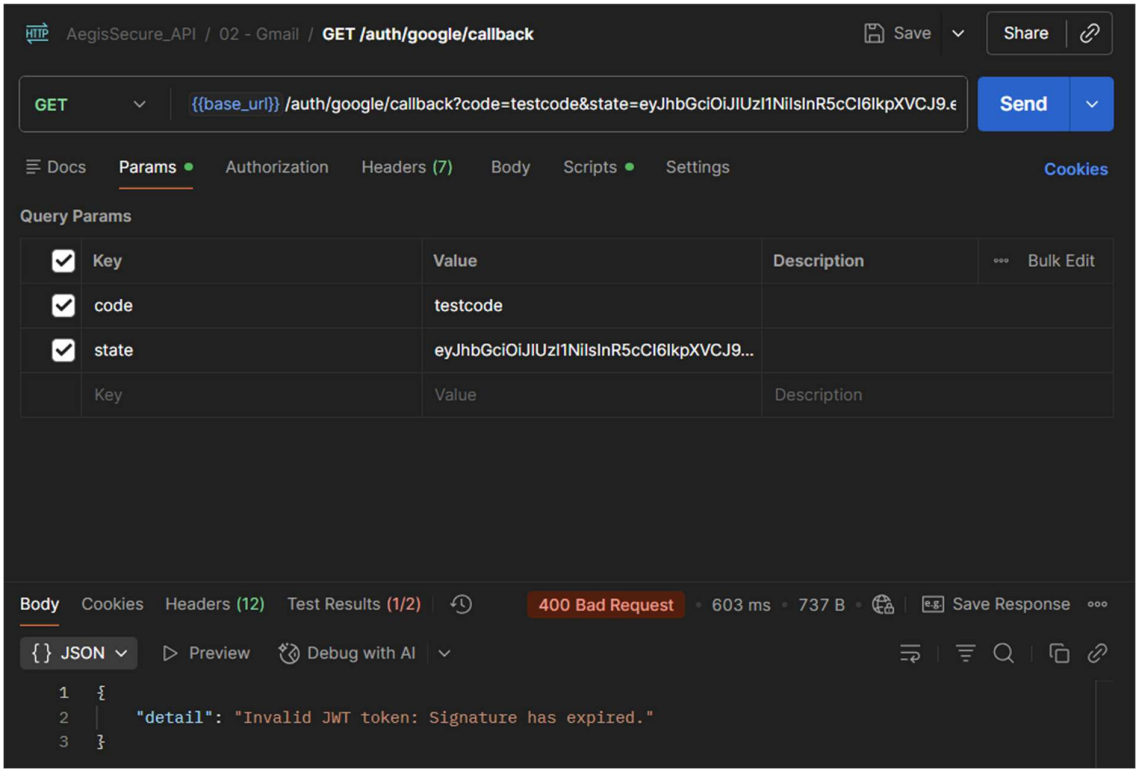
Verdict: PASS

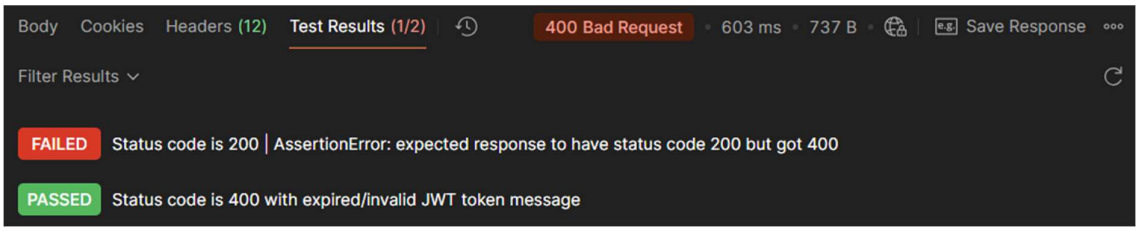
The API accepted the request and responded correctly.

➢ TC-13 : Google OAuth Callback with Invalid or Missing Code (Negative)

Description:
Call the Google OAuth callback endpoint with an incorrect code to check error handling.

Output:





Verdict: PASS

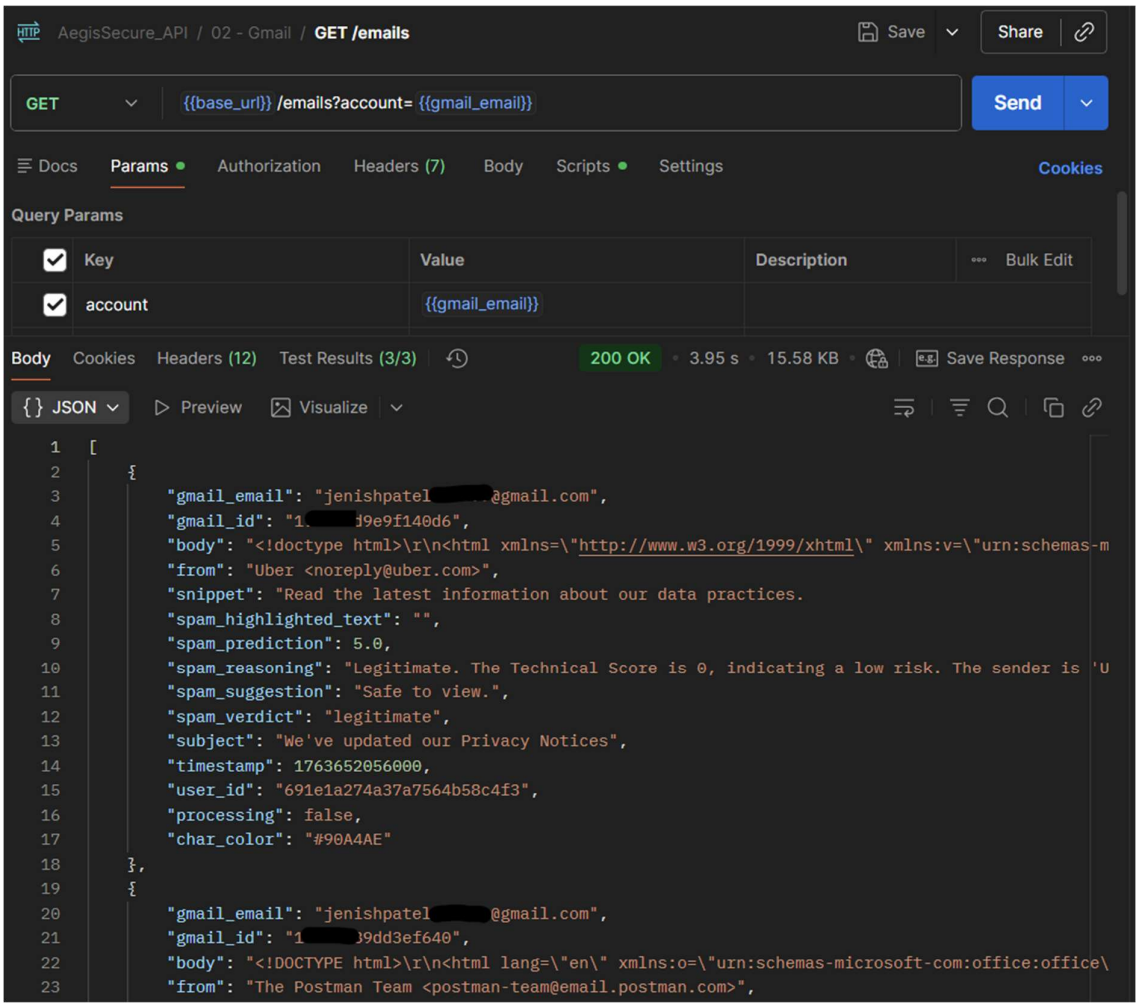
The API correctly showed an error for invalid code, confirming proper validation during the callback stage.

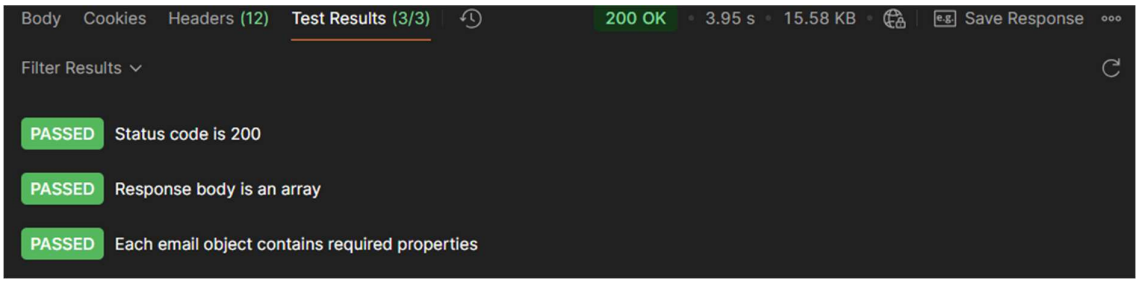
➢ TC-14 : Fetch Emails From Gmail (Positive)

Description:

Test the endpoint that fetches emails for the connected account.

Output:





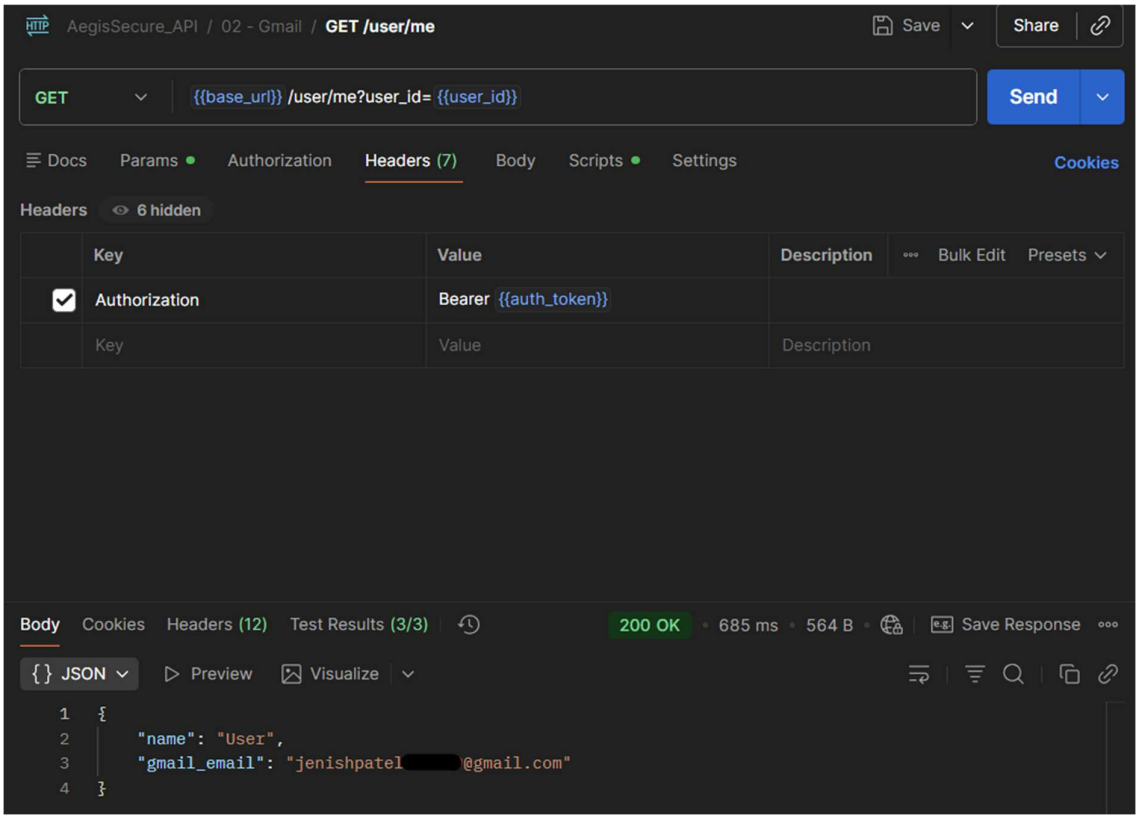
Verdict: PASS

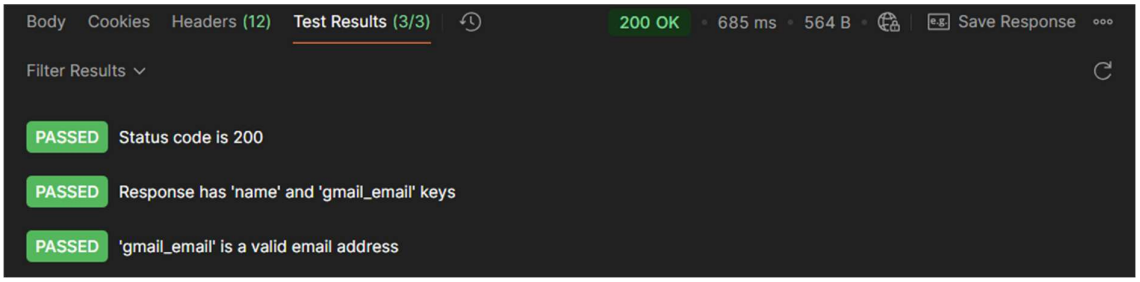
The API successfully fetch the user's emails, confirming that email fetching is working correctly.

➢ TC-15 : Fetch Gmail User Profile (Positive)

Description:
Test the Gmail user profile endpoint to check if user information is fetched correctly from the connected Gmail account.

Output:





Verdict: PASS

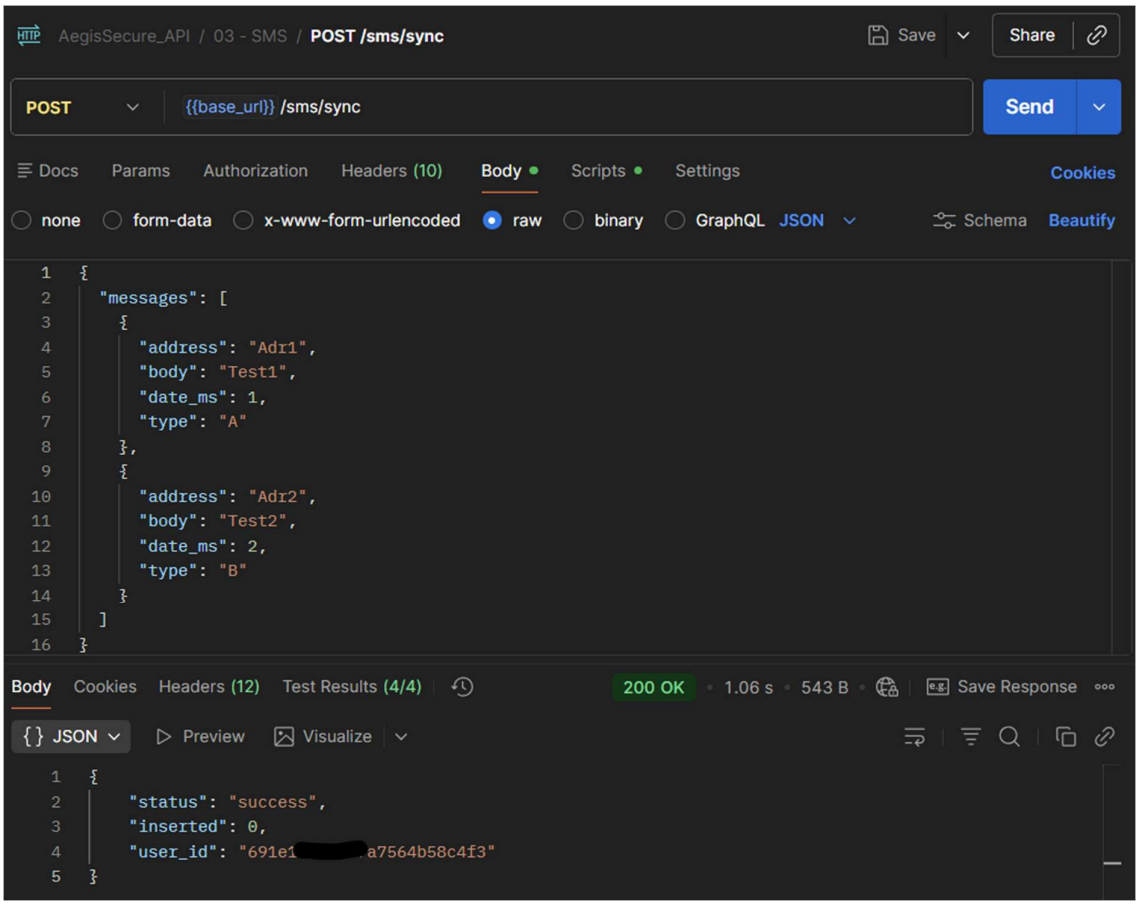
The API returned the Gmail user's profile successfully, confirming proper connection with the Gmail account.

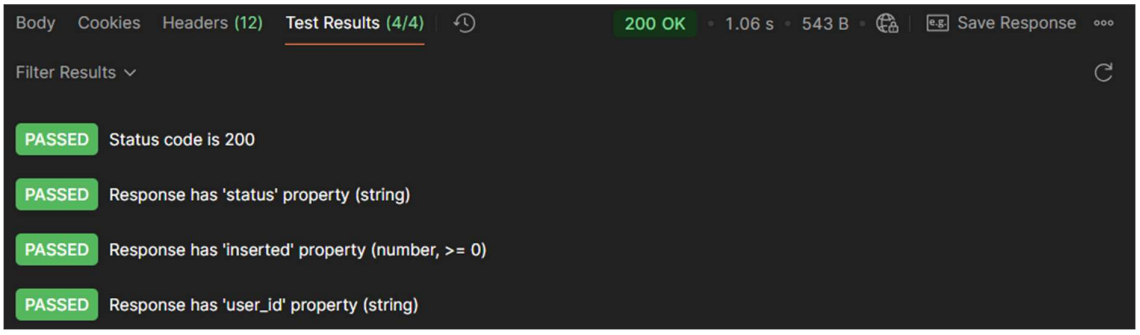
➢ TC-16 : Sync SMS With Valid Data (Positive)

Description:

Sync SMS by sending a valid list of messages.

Output:





Verdict: PASS

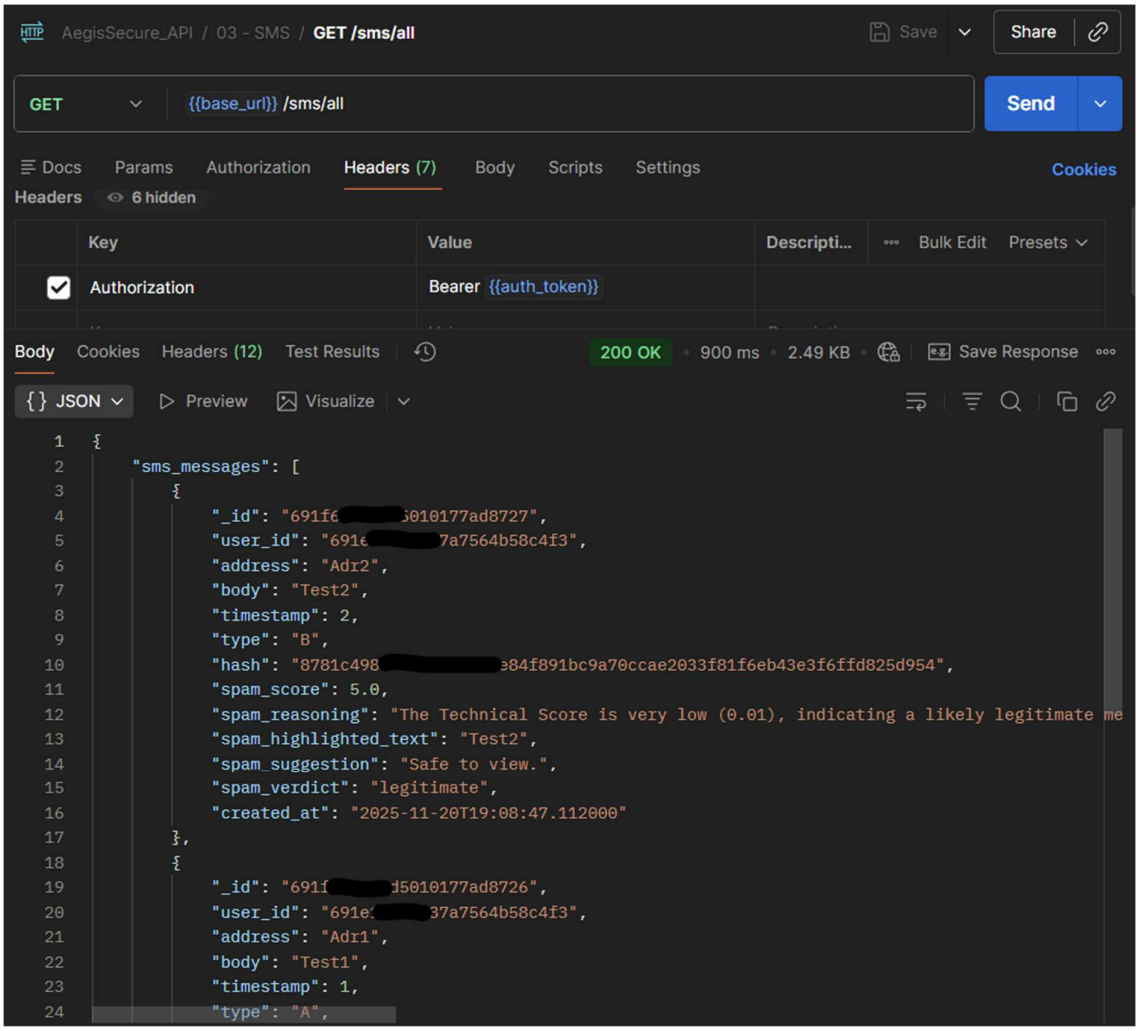
The SMS messages were synced successfully.

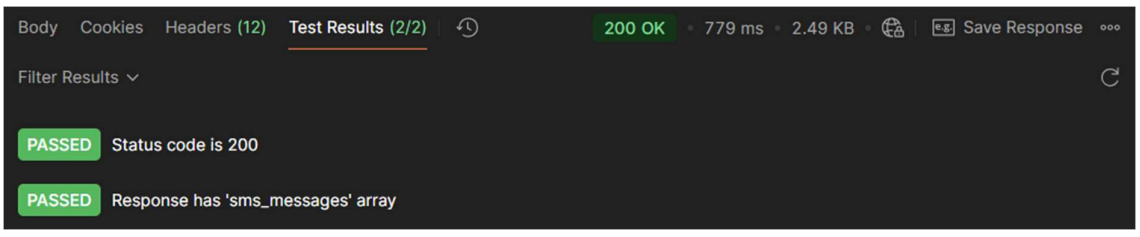
➢ TC-17 : Fetch SMS (Positive)

Description:

Fetch all synced SMS using a valid token.

Output:





Verdict: PASS

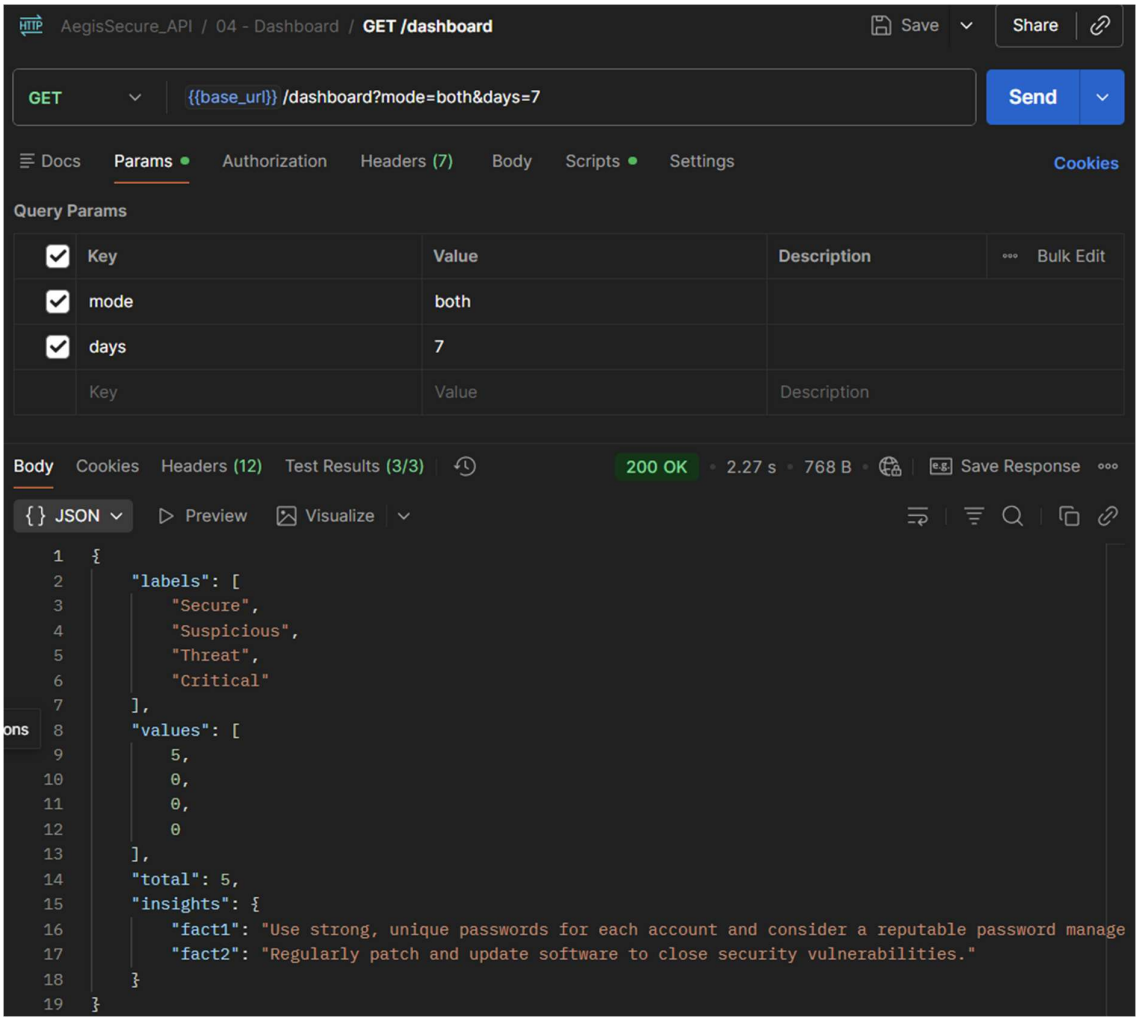
The API returned all SMS messages correctly.

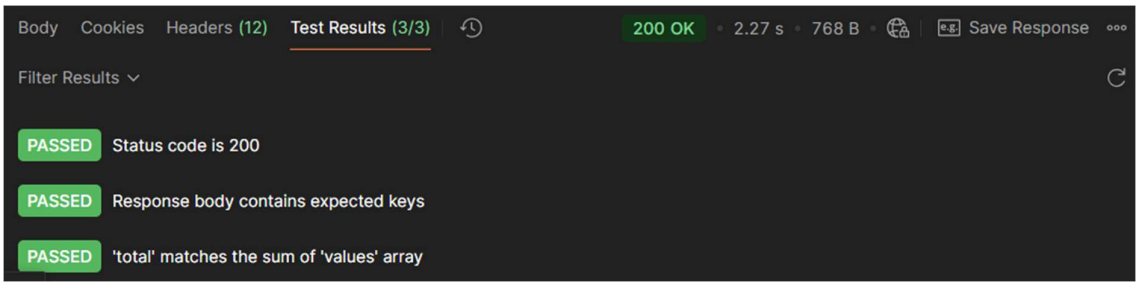
➢ TC-18 : Fetch Dashboard with Valid Inputs (Positive)

Description:

Fetch dashboard data using correct mode and day values.

Output:





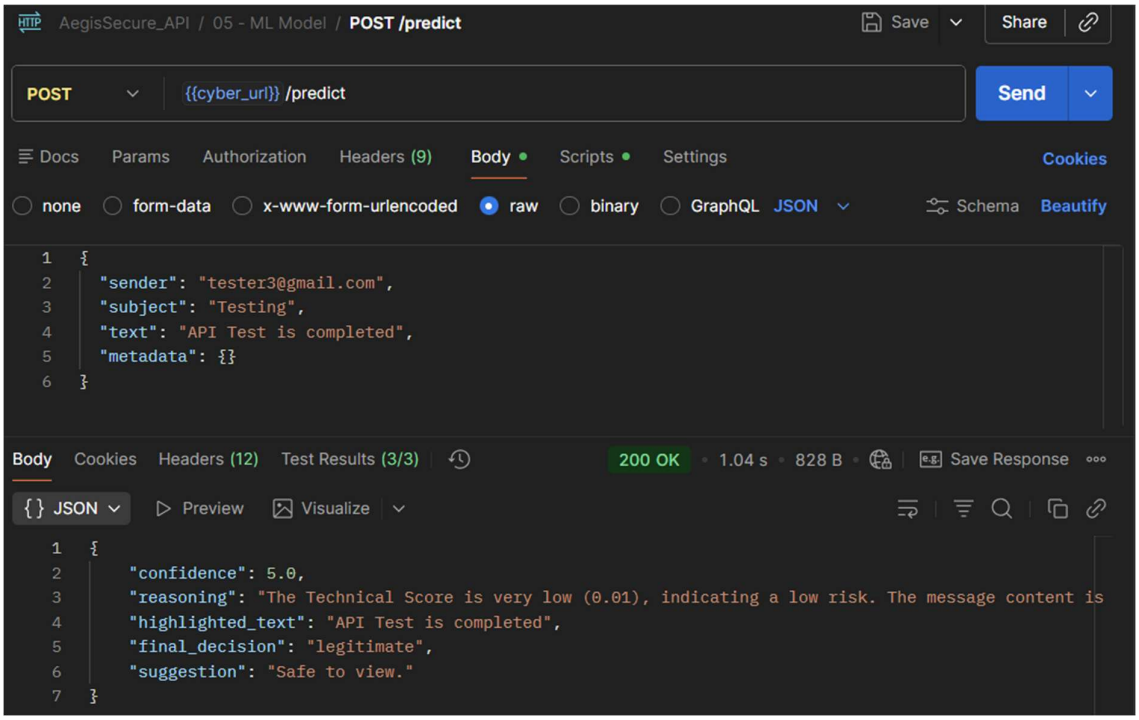
Verdict: PASS

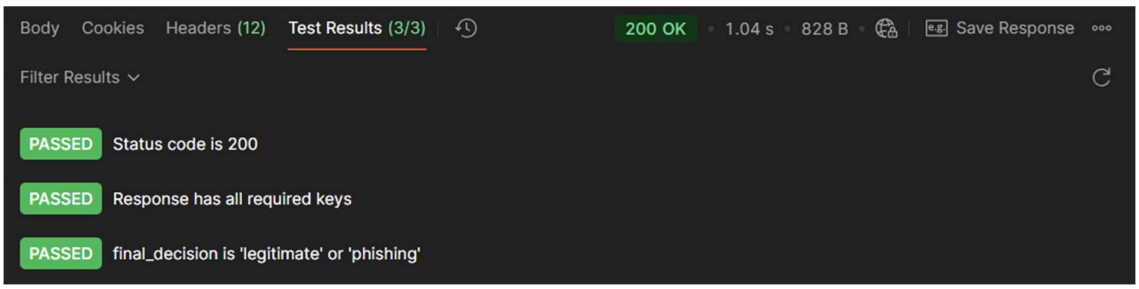
The dashboard data was loaded successfully.

➢ TC-19 : Predict Email Content (Positive)

Description:
Test the ML prediction endpoint by sending a valid email (sender, subject, text). This checks whether the model can analyse the email and return prediction details correctly.

Output:





Verdict: PASS
The API successfully processed the email and returned prediction details such as confidence, reasoning, highlighted text, final decision and suggestion. This confirms the prediction model is working properly.

❖ Conclusion

All the API endpoints of the Aegis Secure Backend were tested and everything worked correctly. Both positive and negative test cases passed without any issues. The authentication system handled correct and incorrect login attempts properly. Gmail-related APIs worked as expected, SMS functions synced correctly and the dashboard data loaded without problems.

Overall, the backend is stable and ready to be used. Error handling, validation and authorization are working well, which means the system is secure and the user experience will be smooth.