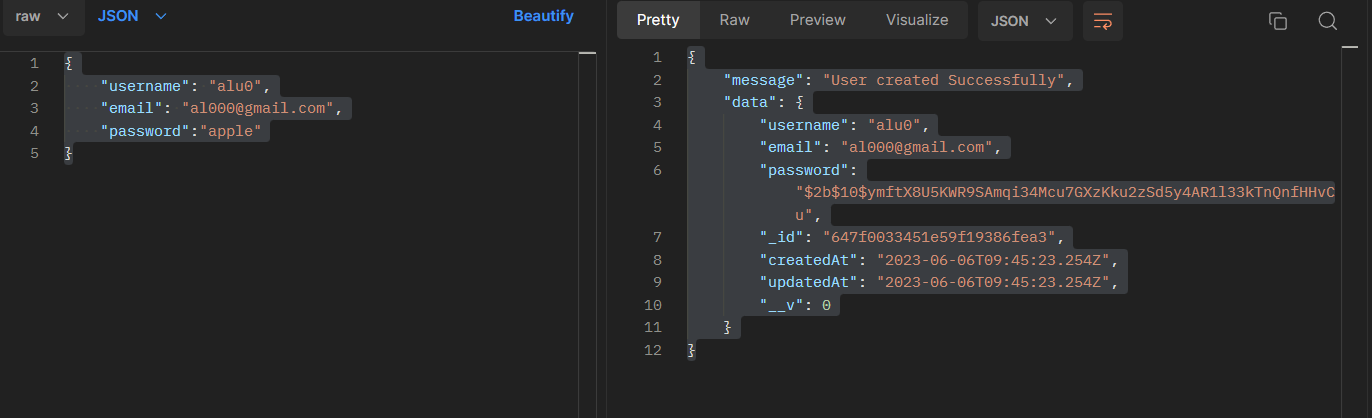
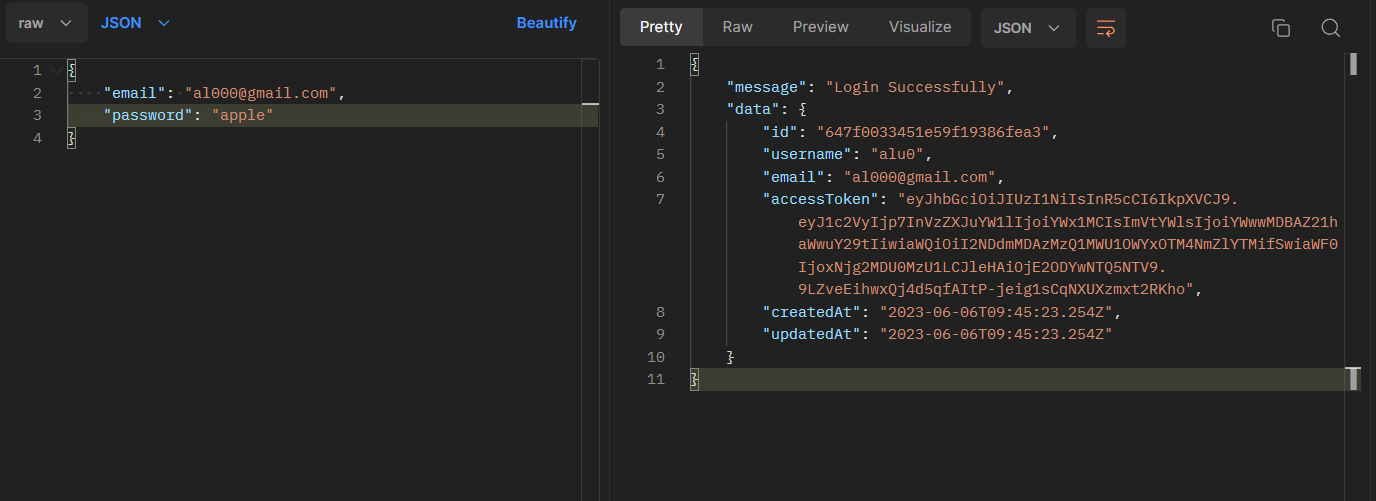
DOCUMENTATION

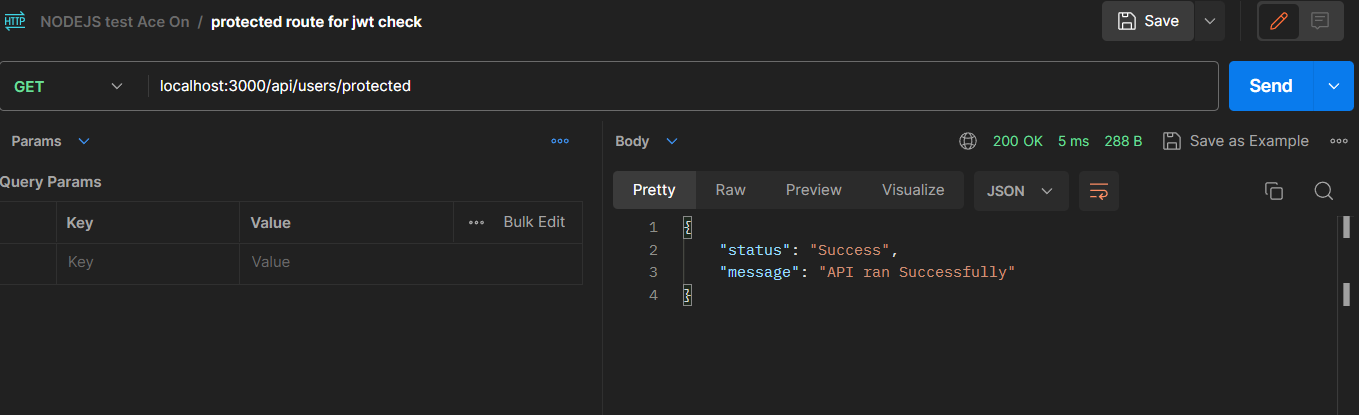
* SIGN UP
  + RESPONSE



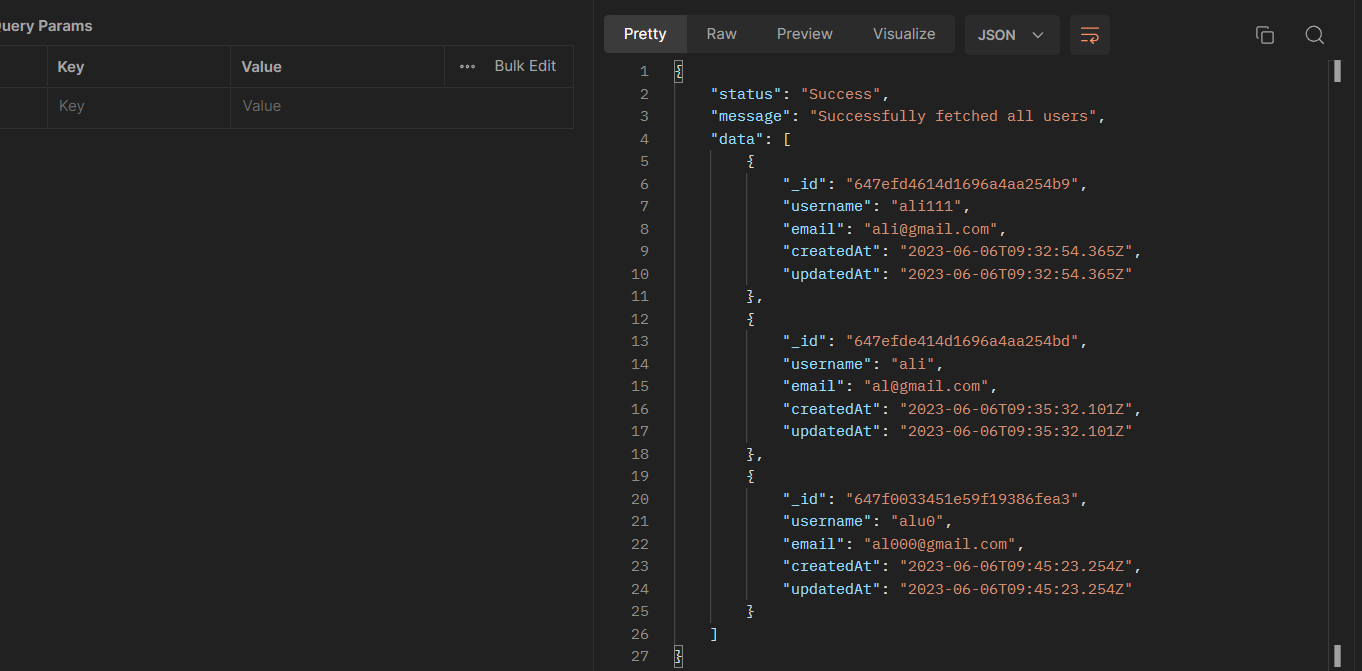
* LOGIN
  + RESPONSE



* PROTECTED
  + RESPONSE



* GET USERS
  + RESPONSE



## **ASSUMPTIONS**

\*I had some queries regarding tasks and asked them in reply of the email but received no response hence I had to make some assumptions.

* Upon hitting login API with 5 failed attempts it will block the user from making any more request until the designated time passed and even successful attempts count for limit as this can be abused too if not taken care of.
* In the task README.md, it was mention to make /user endpoint with rate limit and I assume that /user is a simple GET API that takes no parameters and fetch all the users in the user collection and that pagination is being handled by Front-end.

# **OPTIMIZATIONS**

* Since there is no pagination from backend side on /users I used in-memory database like Redis and the response time went from +250ms to under 20ms.
* Also, an improvement that can be made is to apply pagination on /users to anticipate heavy load when user’s collection gets over +50k documents.
* Added multiple error handling scenarios.