

Joshua Atmadjaya

535230052

TI – B

Back-End Programming

Exercise API Project

1. Prevent Duplicate Email

The screenshot shows a REST client interface with a tab for 'POST POST US...'. The request is a POST to 'http://localhost:5000/api/users'. The 'Body' tab is selected, showing a JSON payload:

```
{  "name" : "MakotoMachiYuki",  "email" : "MakotoMachiYuki@gmail.com",  "password": "123456",  "confirm_password": "123456"}
```

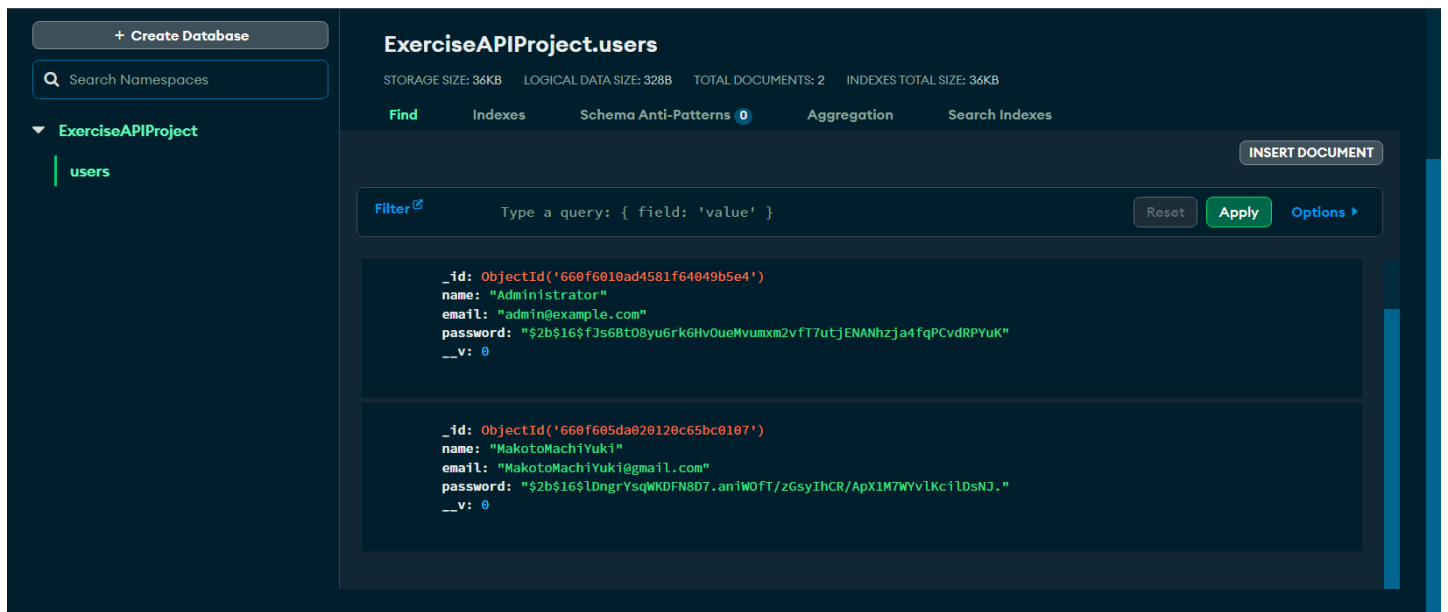
. The 'Response' tab shows a successful response with status '200 OK', time '4.99s', and size '62B'. The response body is:

```
{  "name": "MakotoMachiYuki",  "email": "MakotoMachiYuki@gmail.com"}
```

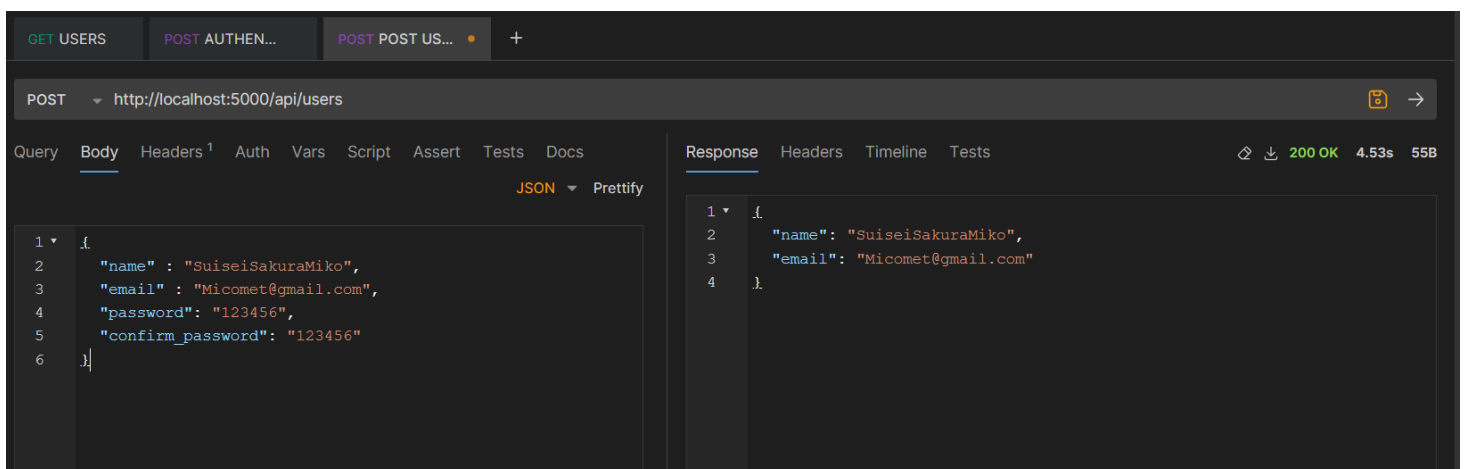
The screenshot shows the same REST client interface, but the response is a '409 Conflict' with status '409 Conflict', time '82ms', and size '147B'. The response body is:

```
{  "statusCode": 409,  "error": "EMAIL_ALREADY_TAKEN_ERROR",  "description": "This email already taken, try use another",  "message": "Email is already taken"}
```

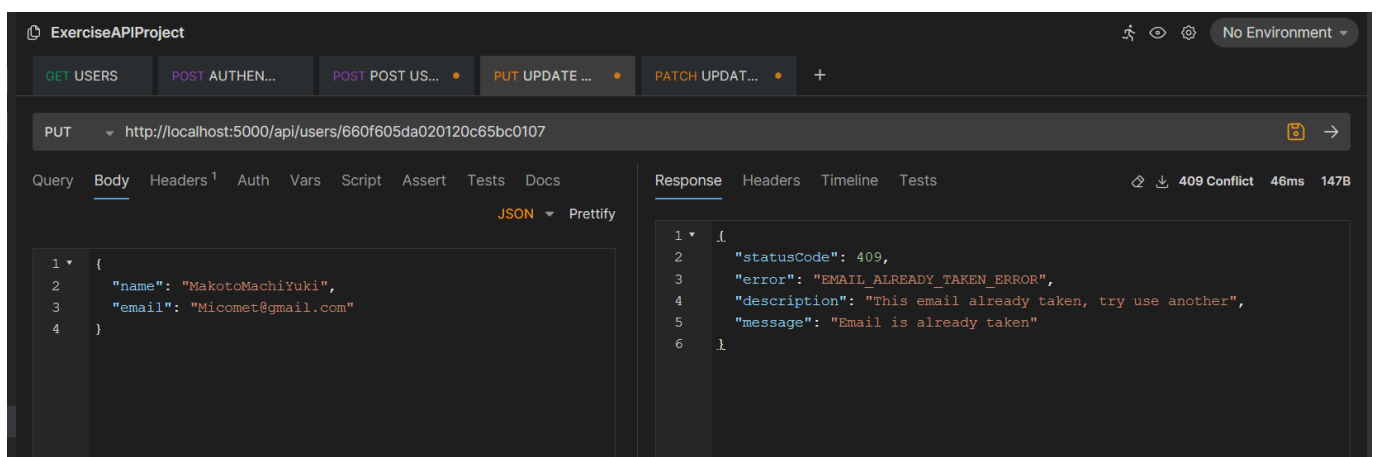
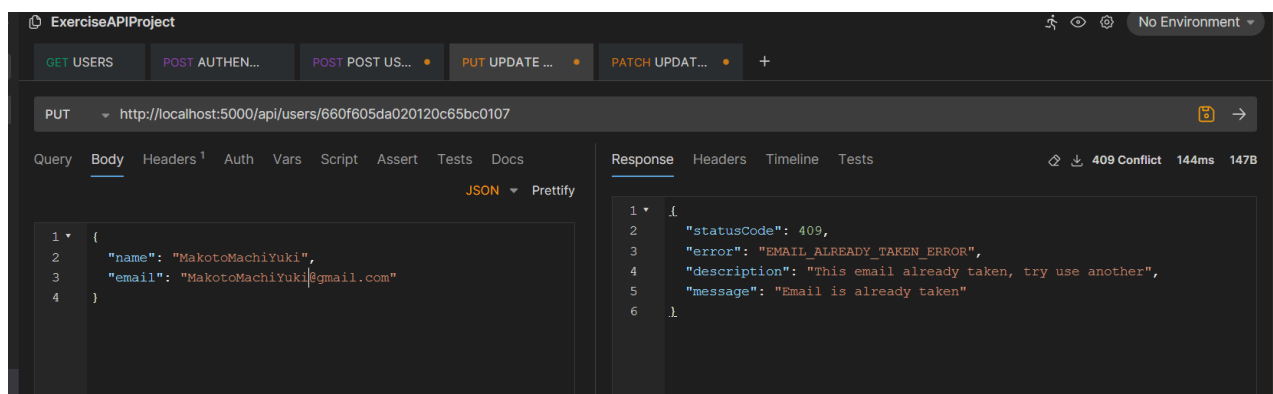
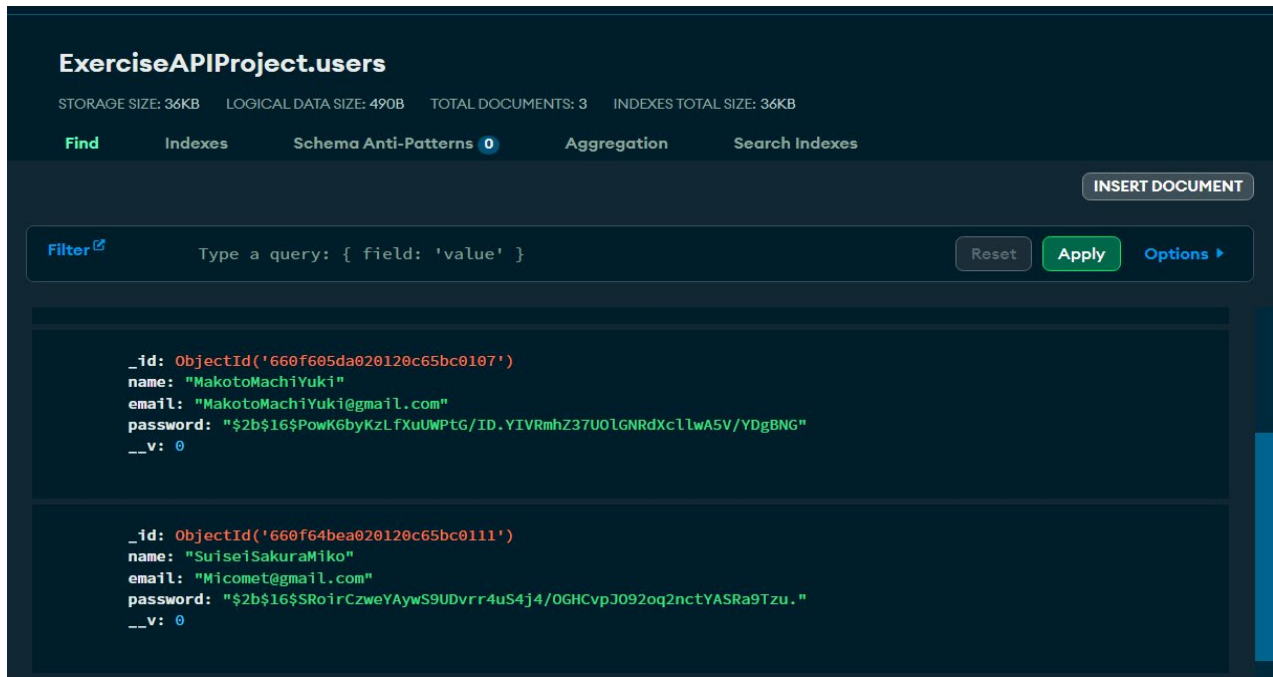
Ketika email yang sudah dicreate dan masuk terdalam database mongodbnnya.. Maka email yg sama tidak akan bisa digunakan lagi karena akan mengembalikan error message 409 dan tidak akan masuk kedalam databasenya



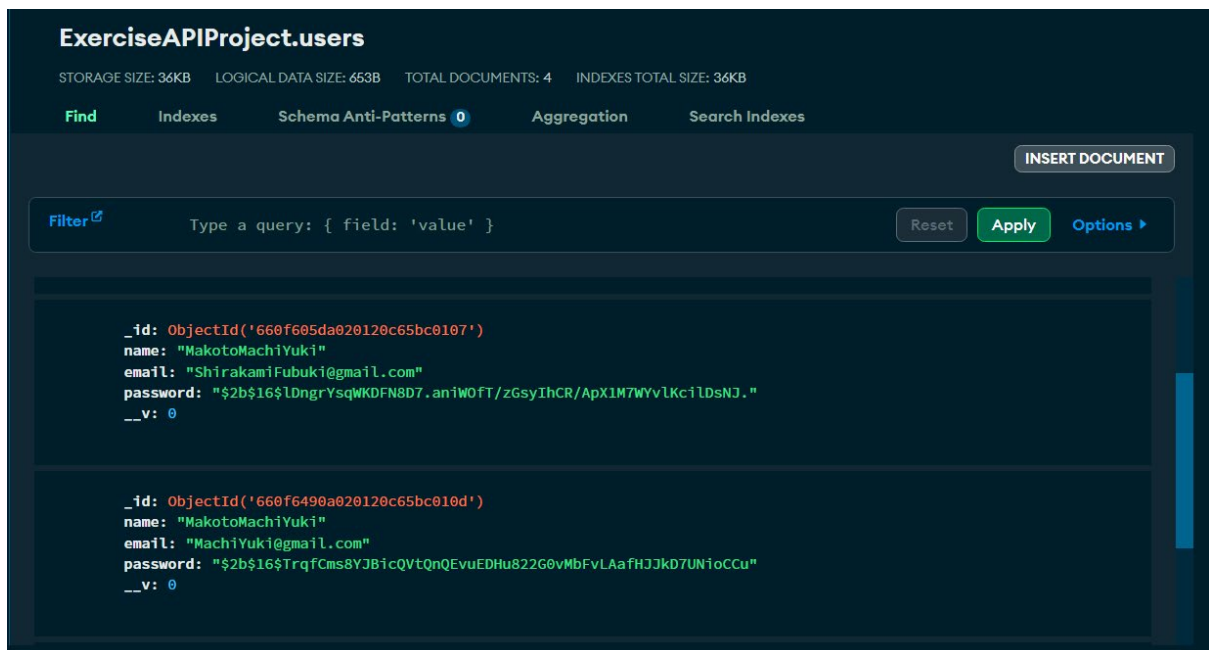
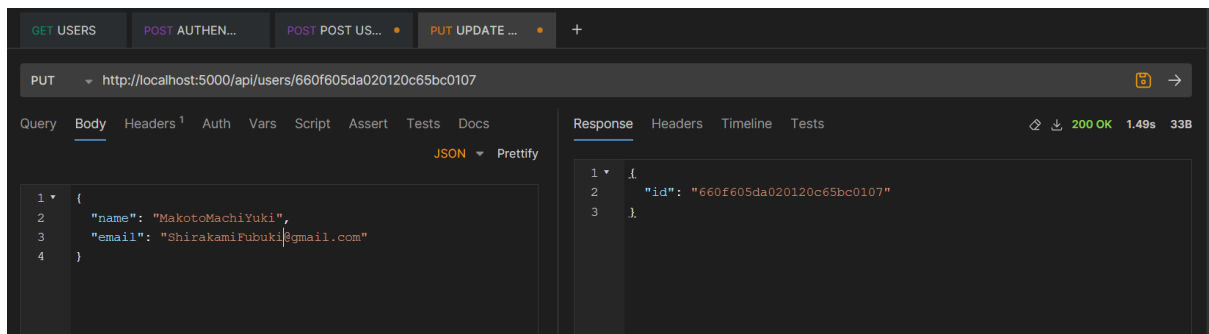
Dan tentu saja email yg berbeda bisa dicreate



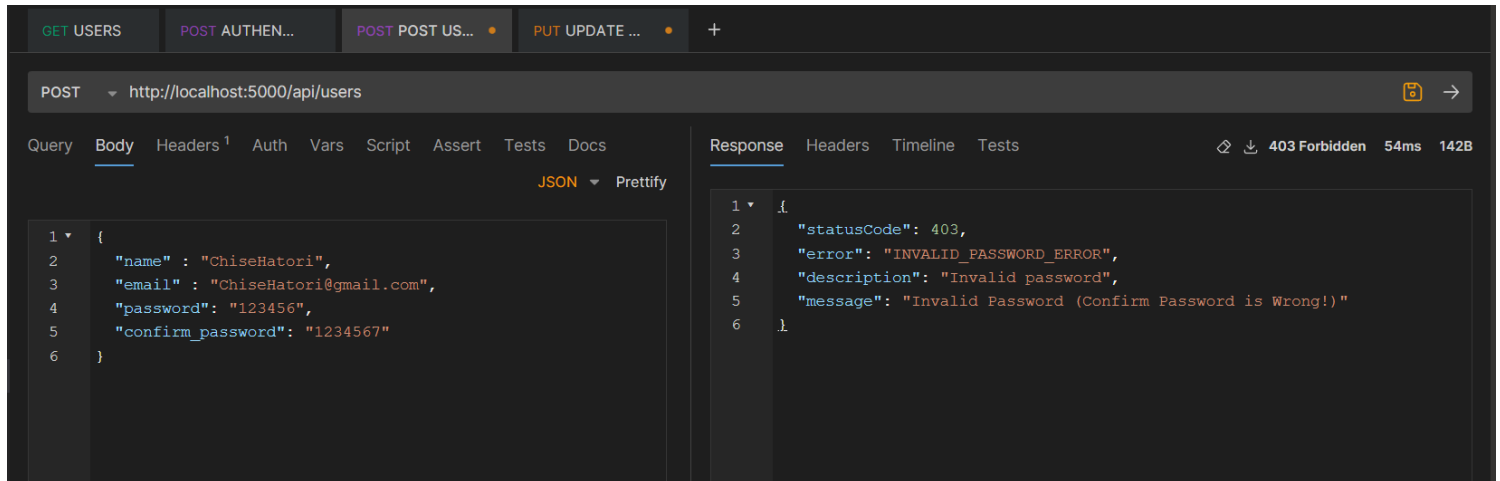
Dan juga ketika emailnya mau diupdate melewati put dengan email sebelumnya maupun email yg sudah terdaftar di databasenya tidak akan bisa diupdate



Dan jika berhasil maka akan return id nya dan emailnya sudah terupdate di databasenya (User yg diupdate adalah user yg email sebelumnya MakotoMachiYuki@gmail.com menjadi ShirakamiFubuki@gmail.com)



2. Confirm Password



POST `http://localhost:5000/api/users`

Query Body Headers¹ Auth Vars Script Assert Tests Docs

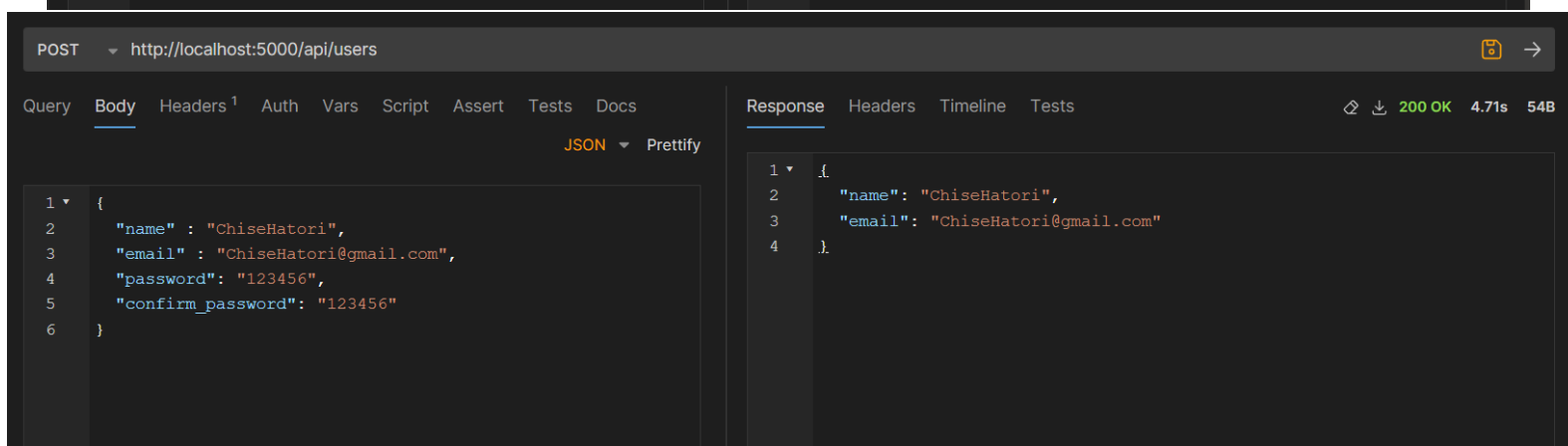
JSON ▾ Prettify

```
1 {
2   "name" : "ChiseHatori",
3   "email" : "ChiseHatori@gmail.com",
4   "password": "123456",
5   "confirm_password": "1234567"
6 }
```

Response Headers Timeline Tests

🔗 ⬇ 403 Forbidden 54ms 142B

```
1 {
2   "statusCode": 403,
3   "error": "INVALID_PASSWORD_ERROR",
4   "description": "Invalid password",
5   "message": "Invalid Password (Confirm Password is Wrong!)"
6 }
```



POST `http://localhost:5000/api/users`

Query Body Headers¹ Auth Vars Script Assert Tests Docs

JSON ▾ Prettify

```
1 {
2   "name" : "ChiseHatori",
3   "email" : "ChiseHatori@gmail.com",
4   "password": "123456",
5   "confirm_password": "123456"
6 }
```

Response Headers Timeline Tests

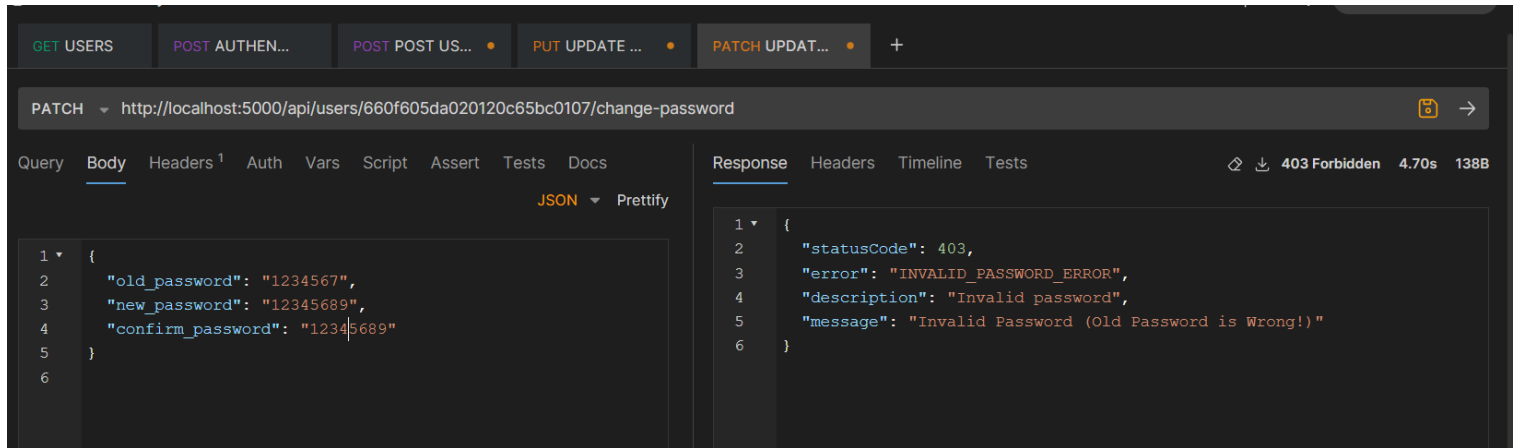
🔗 ⬇ 200 OK 4.71s 54B

```
1 {
2   "name": "ChiseHatori",
3   "email": "ChiseHatori@gmail.com"
4 }
```

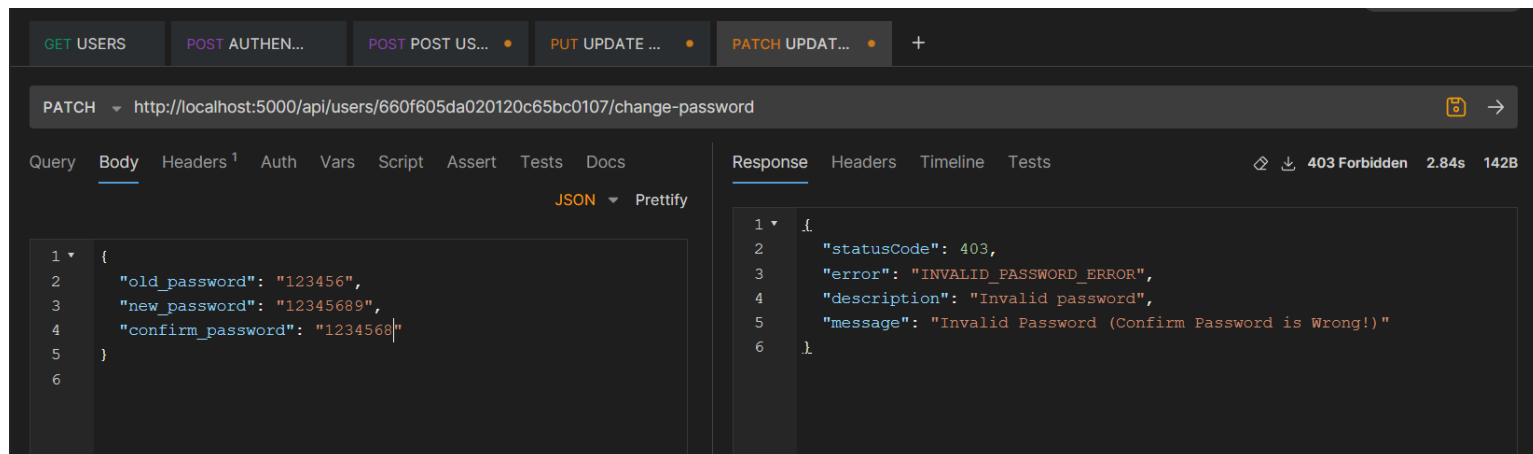
Dan jika `confirm_password` tidak sama dengan `password` maka akan mengembalikan error message “Invalid Password (Confirm Passsword is Wrong!) dengan kode 403

3. Change Password

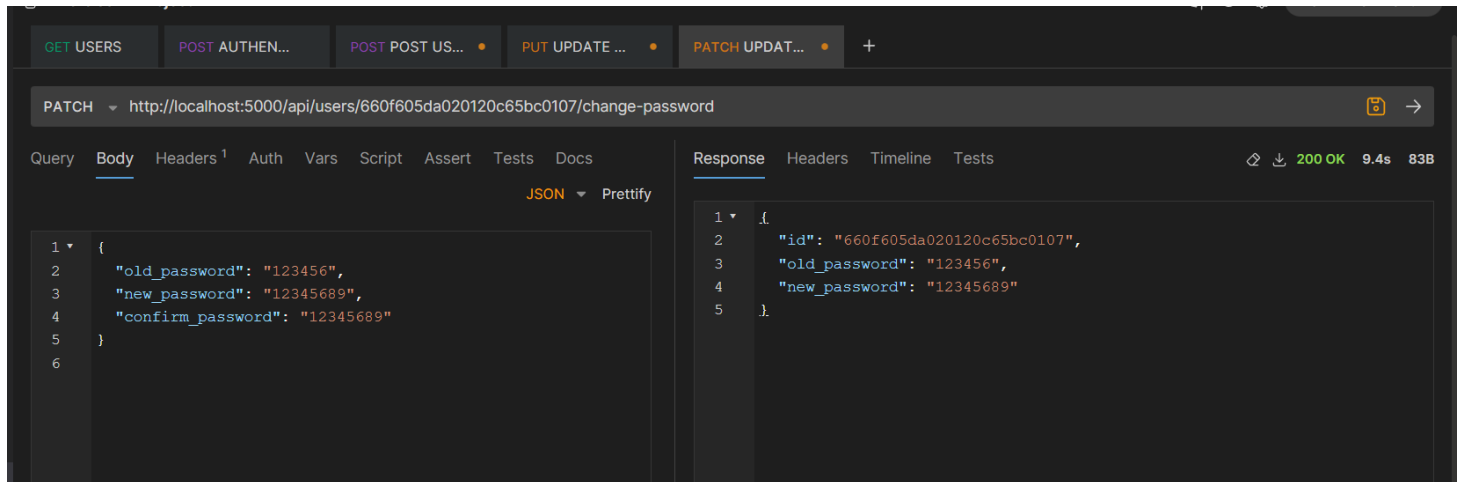
Password awalnya adalah 123456 yg dimana ketika old_passwordnya diisi dengan password yg salah maka akan mengembalikan error message 403 dengan message “Old Password is Wrong”



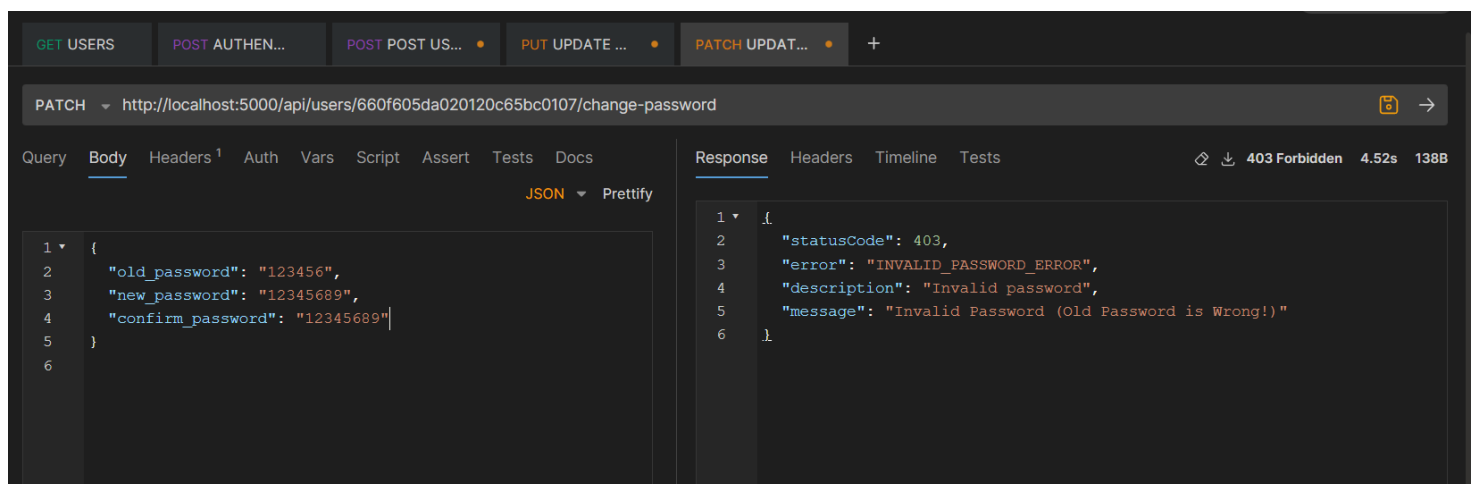
Begitu pula jika confirm_password tidak sama dengan new_password akan mengembalikan message confirm_password is wrong



Dan ketika semua kriteria terpenuhi, `old_password` benar dan `confirm_password` benar maka akan return di response `id`, `old_password` dan `new_password`nya yg akan jadi password sekarang ini



Buktinya adalah ketika body yg sama di send maka akan error karena passwordnya sudah terganti didalam databasenya



Dan ketika `old_password`nya menggunakan password yg barusan diganti (12345689) maka akan sukses

