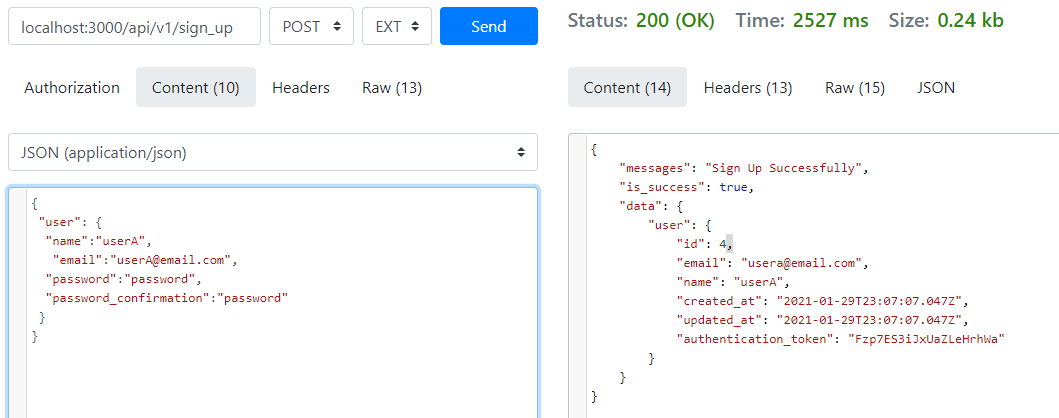
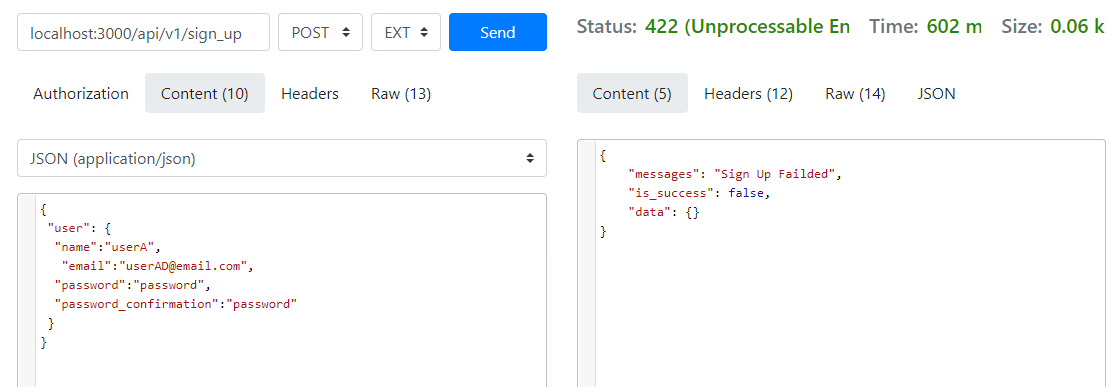
1.sign up

POST : localhost:3000/api/v1/sign\_up

1. If Success



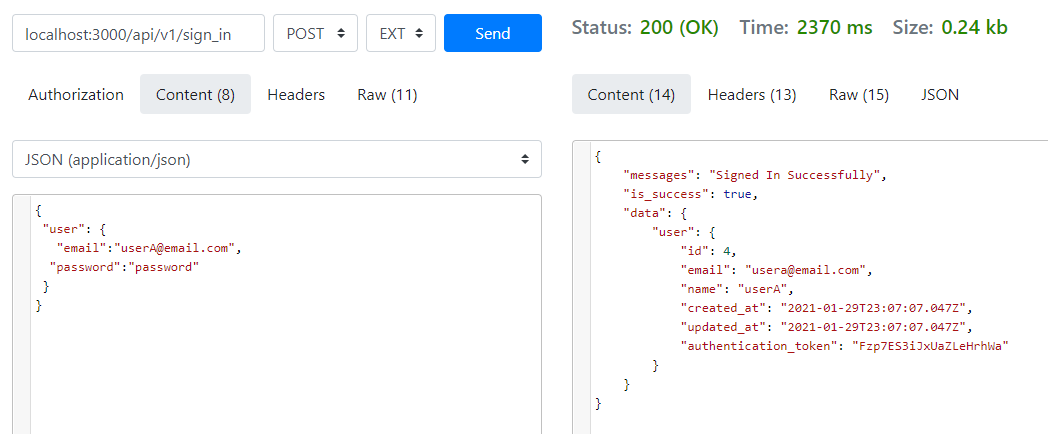
1. If failed cause username duplicate OR email duplicate OR password confirmation not match



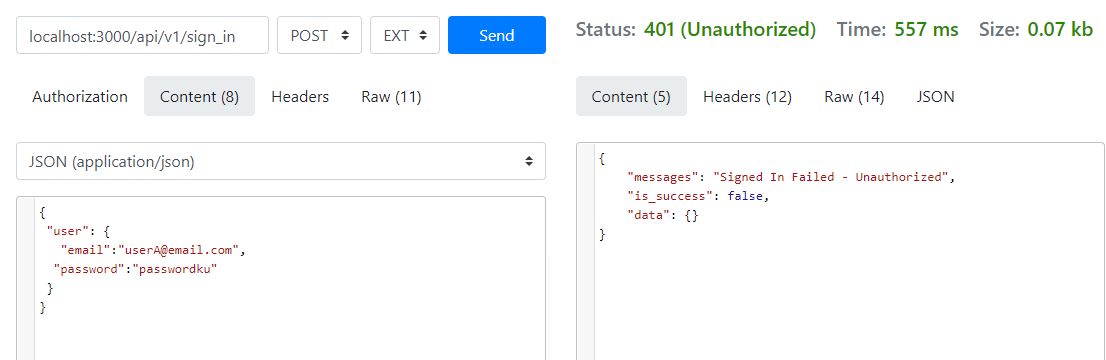
2. Login

POST : localhost:3000/api/v1/sign\_in

1. If Success



1. If Failed cause email or password is wrong

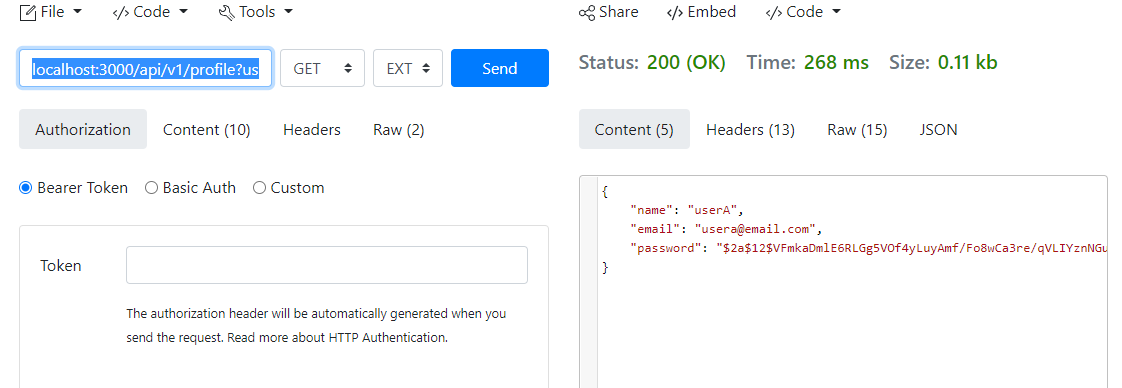


3. See profile

GET : localhost:3000/api/v1/profile?user\_email=email\_user&user\_token=token\_user

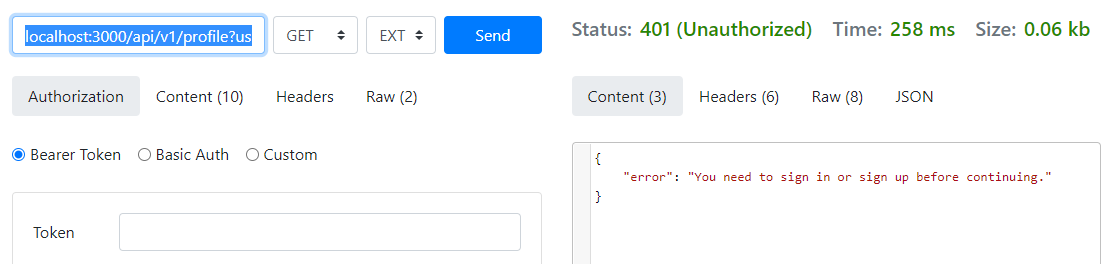
1. If Success

Ex : localhost:3000/api/v1/profile?user\_email=usera@email.com&user\_token=Fzp7ES3iJxUaZLeHrhWa



1. If wrong token Or email

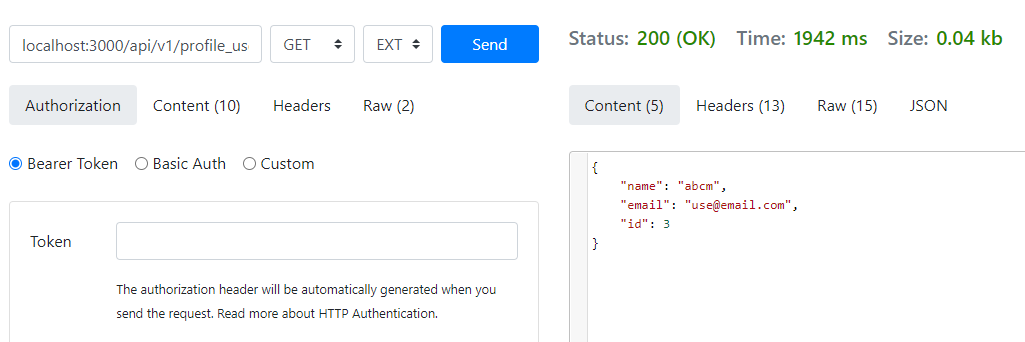
Ex: localhost:3000/api/v1/profile?user\_email=usea@email.com&user\_token=Fzp7ES3iJxUaZLeHrhWa



4. See other user

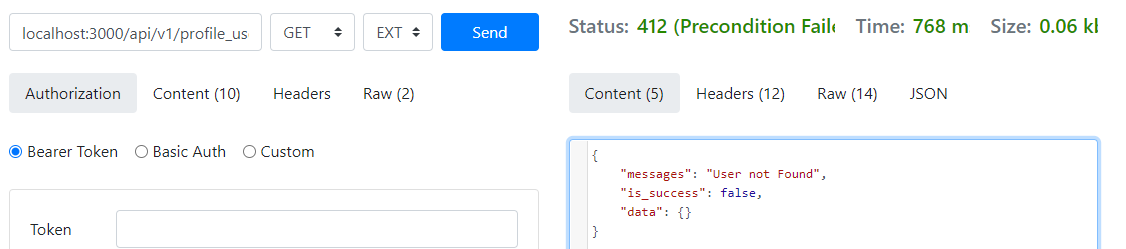
GET : localhost:3000/api/v1/profile\_user/2?user\_email=usera@email.com&user\_token=Fzp7ES3iJxUaZLeHrhWa

a. If success



b. If Failed

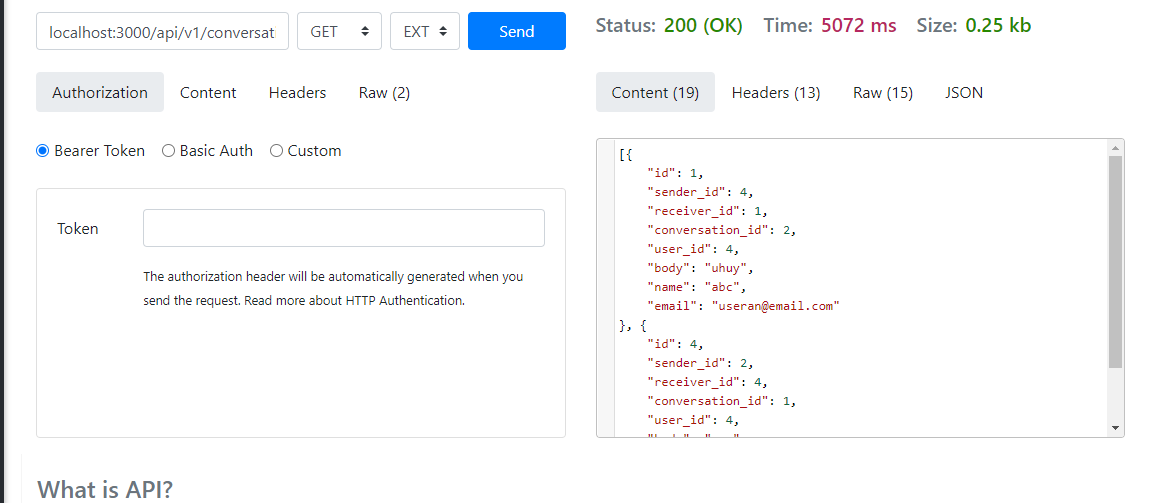
Failed when not found data



5. see list conversation current user

See list conversation with last message

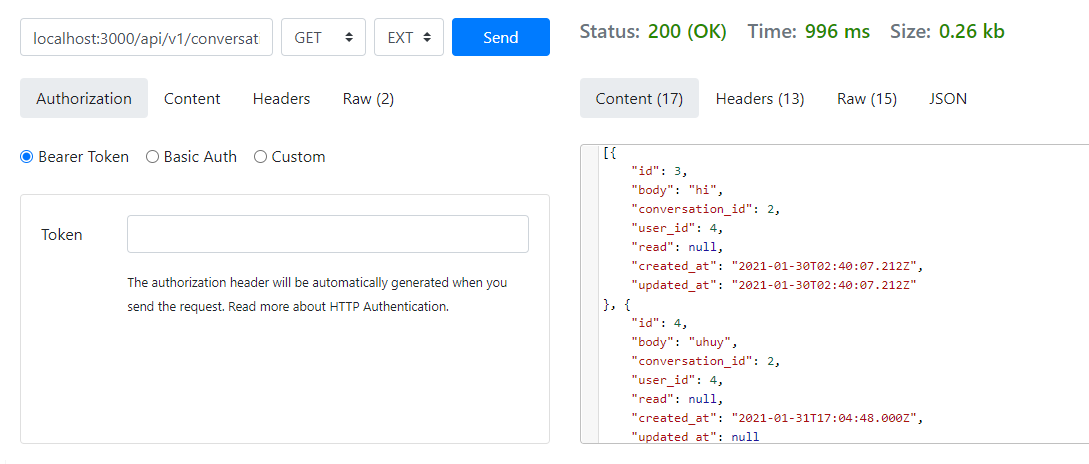
GET: localhost:3000/api/v1/conversations?user\_email=usera@email.com&user\_token=Fzp7ES3iJxUaZLeHrhWa



6. See message userA with abc

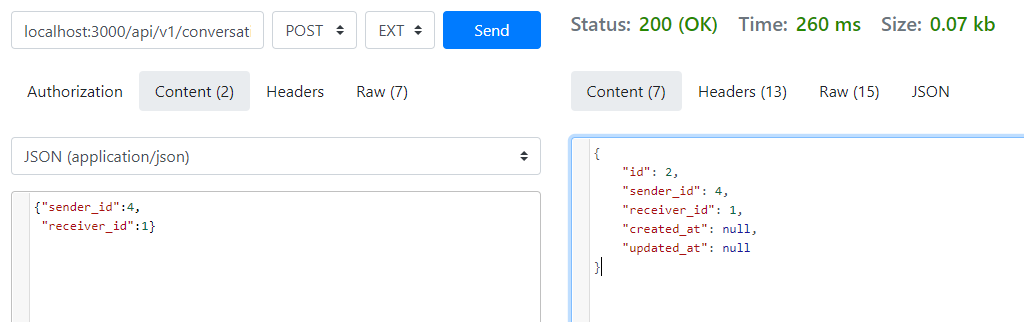
GET: localhost:3000/api/v1/conversations/2/messages?user\_email=usera@email.com&user\_token=Fzp7ES3iJxUaZLeHrhWa

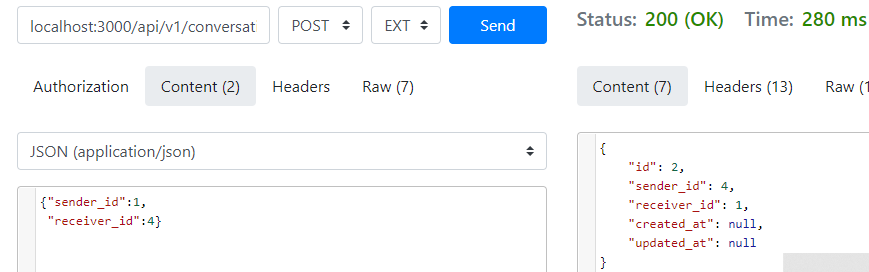
User\_id means who send that message. When user open this endpoint all unread message with this user become read.



7. make new room

POST: localhost:3000/api/v1/conversations?user\_email=usera@email.com&user\_token=Fzp7ES3iJxUaZLeHrhWa

sender\_id and receiver\_id is unique to check already in conversation or not, when check already in table so will not created. Conversation with sender\_id:1 and receiver\_id:4 before already created , so not created again.



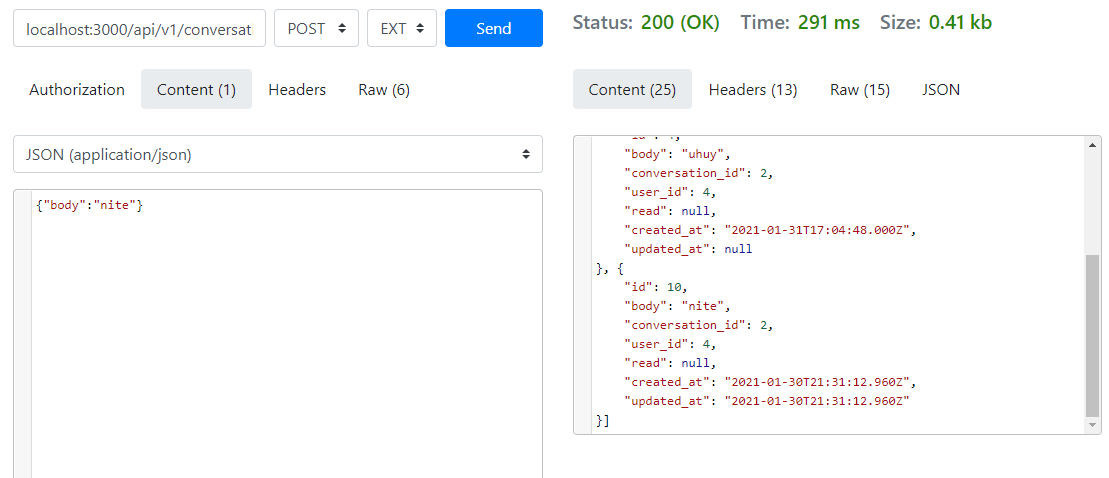
8. Create message

POST:

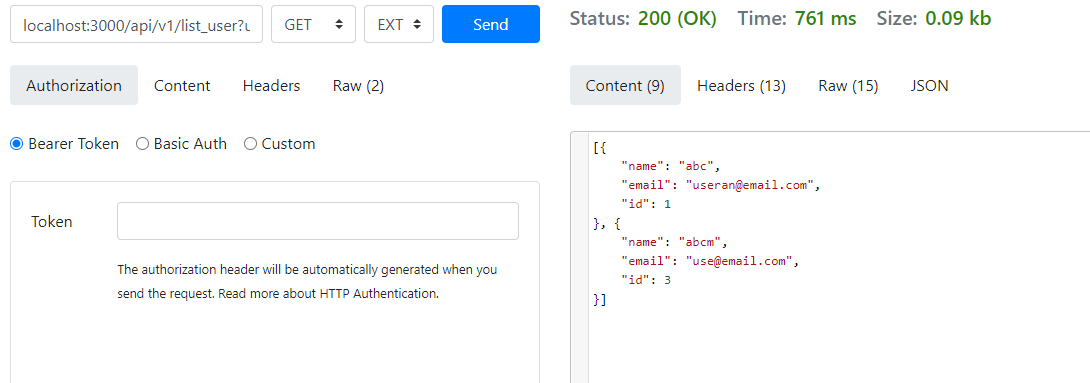
localhost:3000/api/v1/conversations/conversation\_id/messages?user\_email=current\_user\_email@email.com&user\_token=current\_user\_token

ex:

localhost:3000/api/v1/conversations/2/messages?user\_email=usera@email.com&user\_token=Fzp7ES3iJxUaZLeHrhWa



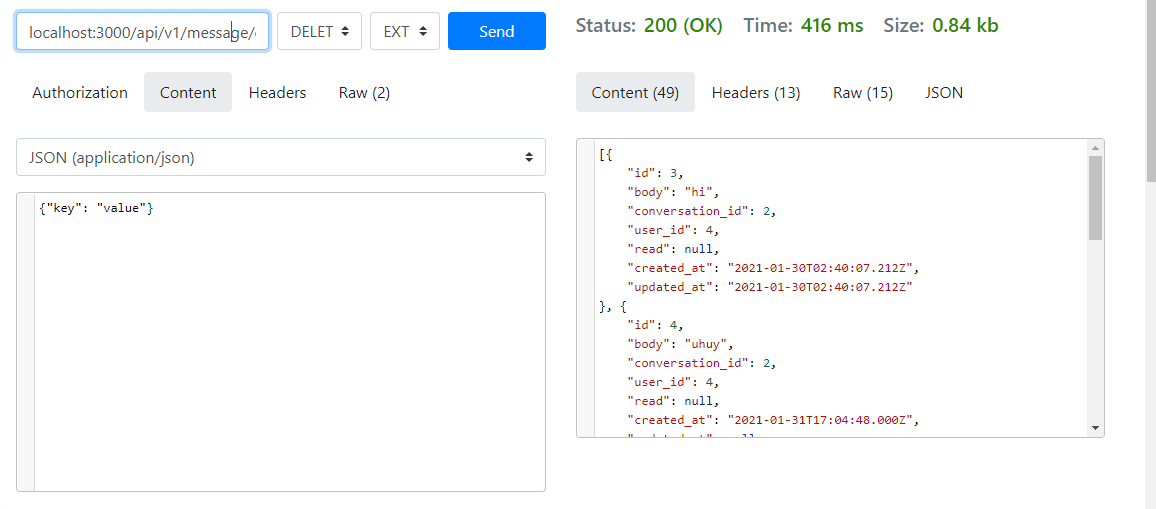
9. list other users

GET: localhost:3000/api/v1/list\_user?user\_email=usera@email.com&user\_token=Fzp7ES3iJxUaZLeHrhWa 

10. Delete message

DELETE : localhost:3000/api/v1/message/delete/5?user\_email=usera@email.com&user\_token=Fzp7ES3iJxUaZLeHrhWa

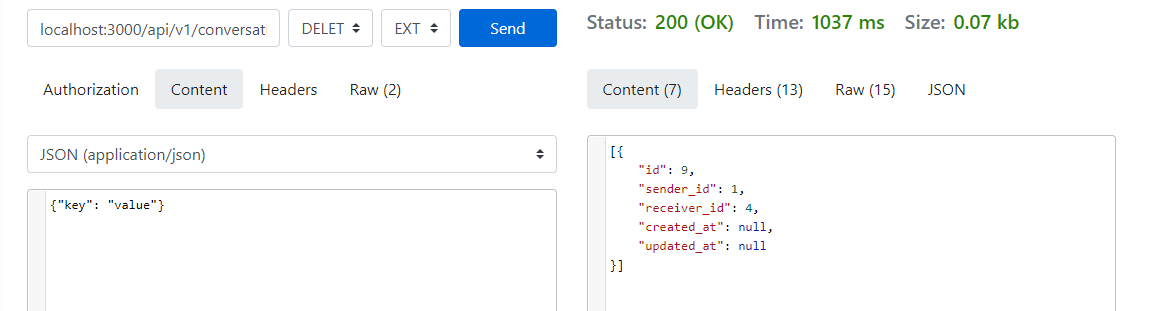
Check based current user\_id and id want to delete. User just can delete message where he/she become sender/receiver in conversation table



11. Delete Conversation

DELETE: localhost:3000/api/v1/conversation/delete/10?user\_email=usera@email.com&user\_token=Fzp7ES3iJxUaZLeHrhWa

Users just can delete conversation that they have it. When a conversation is deleted, so message where have relation with that conversation will be deleted too



Untuk count message sebenarnya sudah saya buat di model tetapi belum saya testing