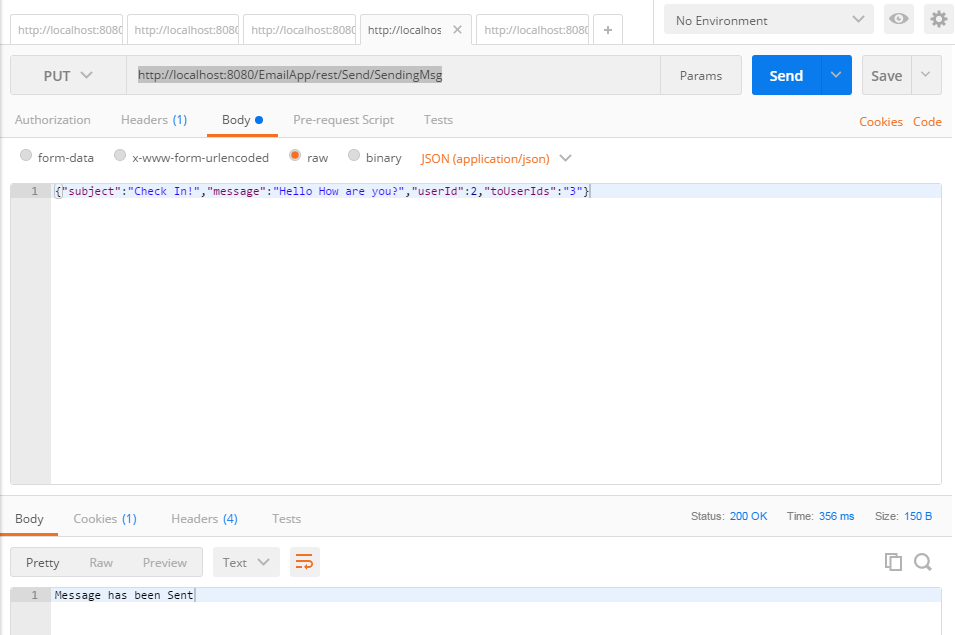
Email App

Created Backend for Email app. I could not create the front end because of time constraints so you will need to use PostMan to test this. I am attaching all the functionalities with their URLs as well as body of requests. The skeleton of all the functionalities needed for an email application has been created but due to lack of time I could not complete the entire assignment. I would like you to kindly have a look and let me know what you think.

1. Send Email



URL: <http://localhost:8080/EmailApp/rest/Send/SendingMsg> (PUT)

Body: {"subject":"Check In!","message":"Hello How are you?","userId":2,"toUserIds":"3"}

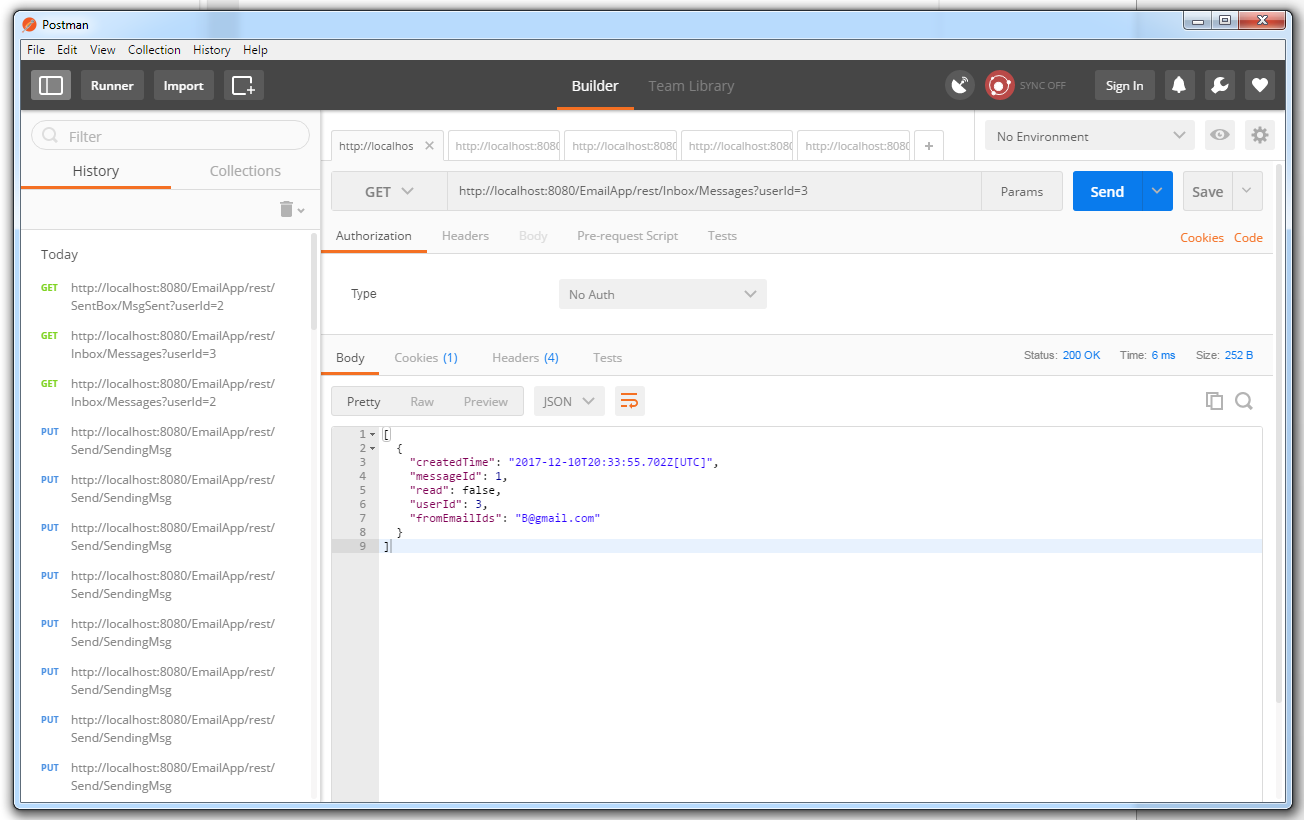
userId -> User for who has sent the email/request

toUserIds -> User to whom this email/request needs to be sent(For the time being there is only one To User)

Subject -> Subject of the email

P.S. Here an email which is sent will be updated on sender’s sent box as well as receiver’s inbox.

2a. All Inbox – All emails in the inbox



URL -> <http://localhost:8080/EmailApp/rest/Inbox/Messages?userId=3> (GET)

Response -> Message from inbox with message ID, read tag, for user ID and who has sent the email

[

{

"createdTime": "2017-12-10T20:33:55.702Z[UTC]",

"messageId": 1,

"read": false,

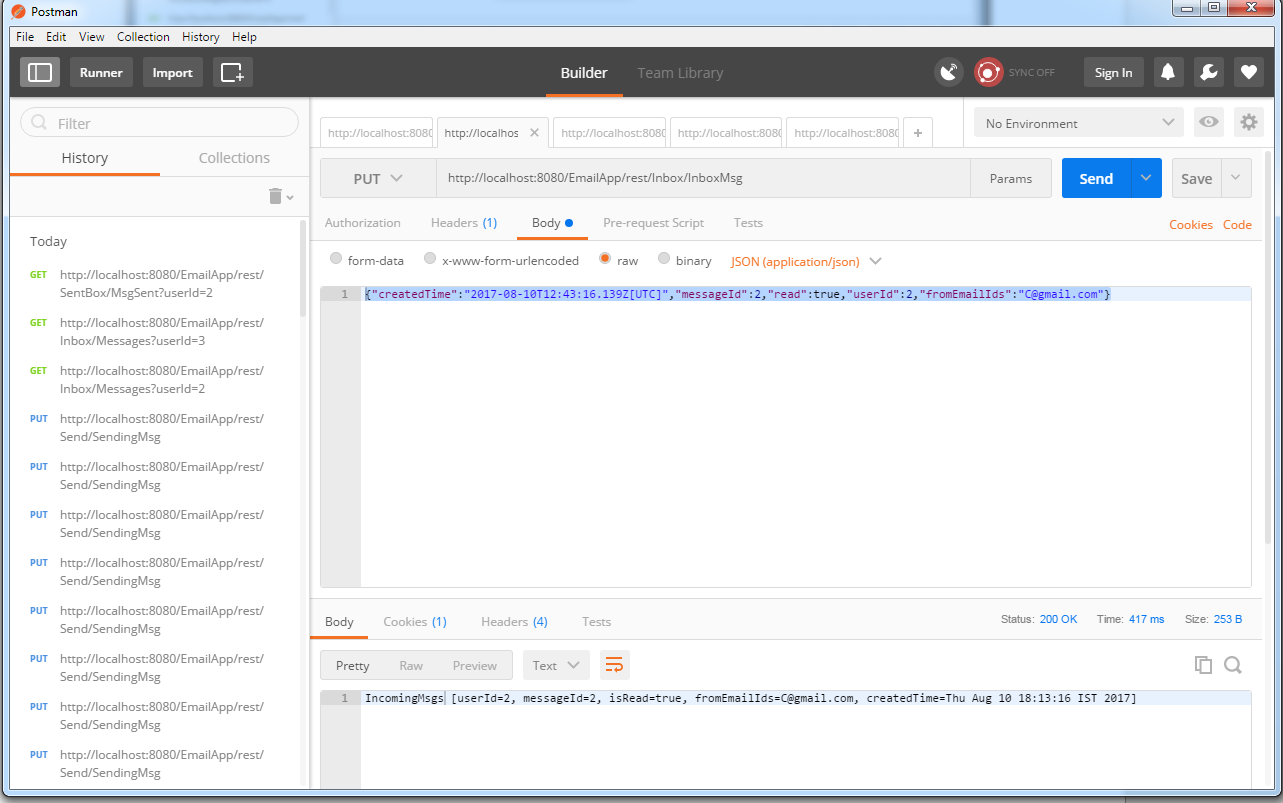
"userId": 3,

"fromEmailIds": "B@gmail.com"

}

]

2b. Put in Direct Inbox Message: Message directly sent to inbox



URL: <http://localhost:8080/EmailApp/rest/Inbox/InboxMsg> (PUT)

Request -> direct data which needs to be passed to Inbox

{"createdTime":"2017-08-10T12:43:16.139Z[UTC]","messageId":2,"read":true,"userId":2,"fromEmailIds":"C@gmail.com"}

This same process is done for Sent Box, trash and Drafts section. The URLS for each are:

3a. All Drafts

URL: [http://localhost:8080/EmailApp/rest/ DraftBox/MsgDraft?userId=1](http://localhost:8080/EmailApp/rest/%20DraftBox/MsgDraft?userId=1)

3b. Put in Direct Draft Message: Message directly sent to draft

URL: http://localhost:8080/EmailApp/rest/DraftBox/DraftMsg (PUT)

4a. All Sent

URL: <http://localhost:8080/EmailApp/rest/SentBox/MsgSent?userId=1>

4b. Put in Direct Sent Message: Message directly sent to Sent Box

URL: <http://localhost:8080/EmailApp/rest/SentBox/>SentBoxMsg (PUT)

5a. All Trash

URL: [http://localhost:8080/EmailApp/rest/TrashBox/MsgTrash?userId=1](http://localhost:8080/EmailApp/rest/SentBox/MsgSent?userId=1)

5b. Put in Direct Trash Message: Message directly sent to Trash Box

URL: <http://localhost:8080/EmailApp/rest/TrashBox/>TrashMsg (PUT)

# New Features

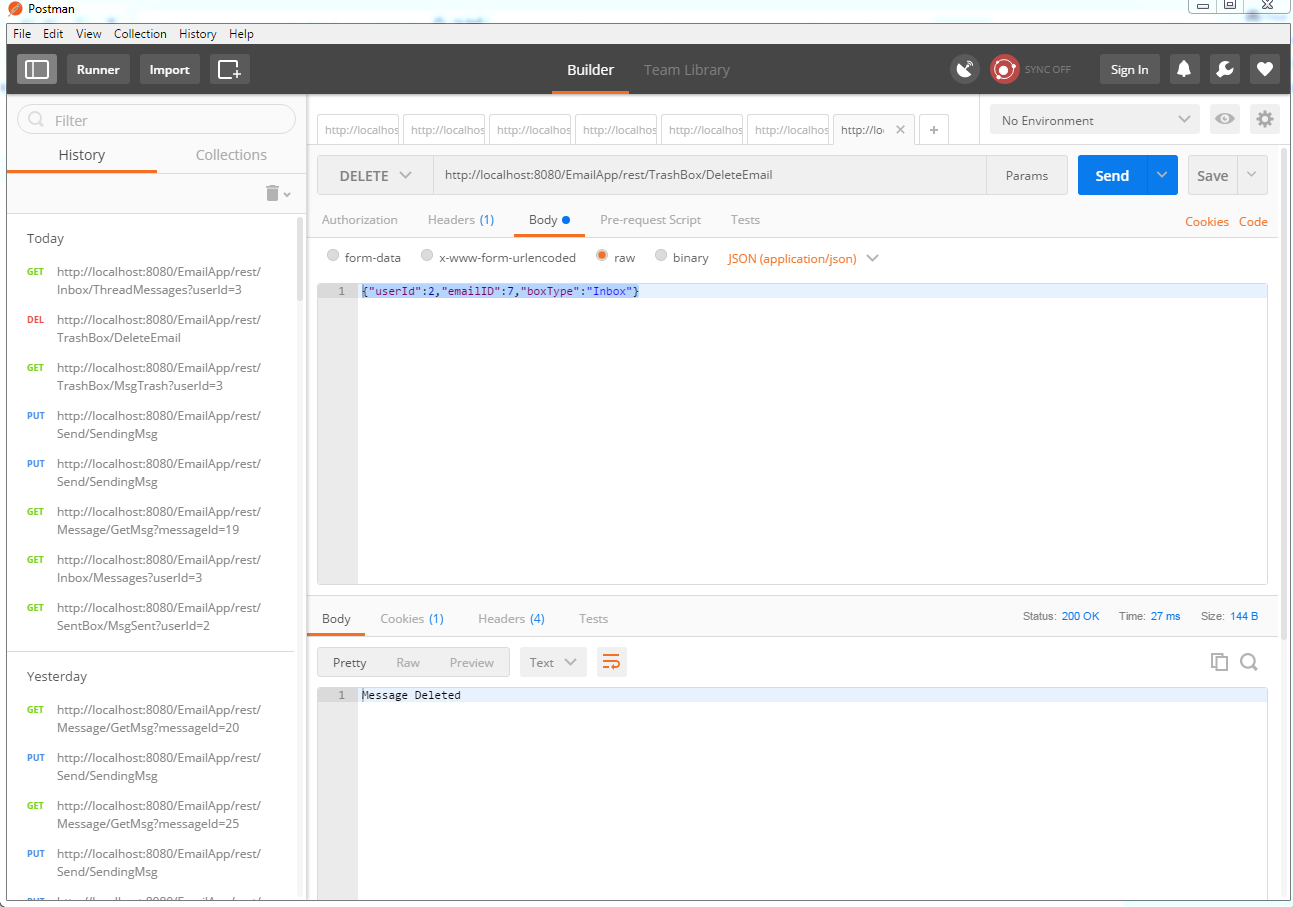
1. Trash Service: Sending Emails (Inbox, Sent, Draft) to trash.

URL: <http://localhost:8080/EmailApp/rest/TrashBox/DeleteEmail>

With DELETE request and JSON body as

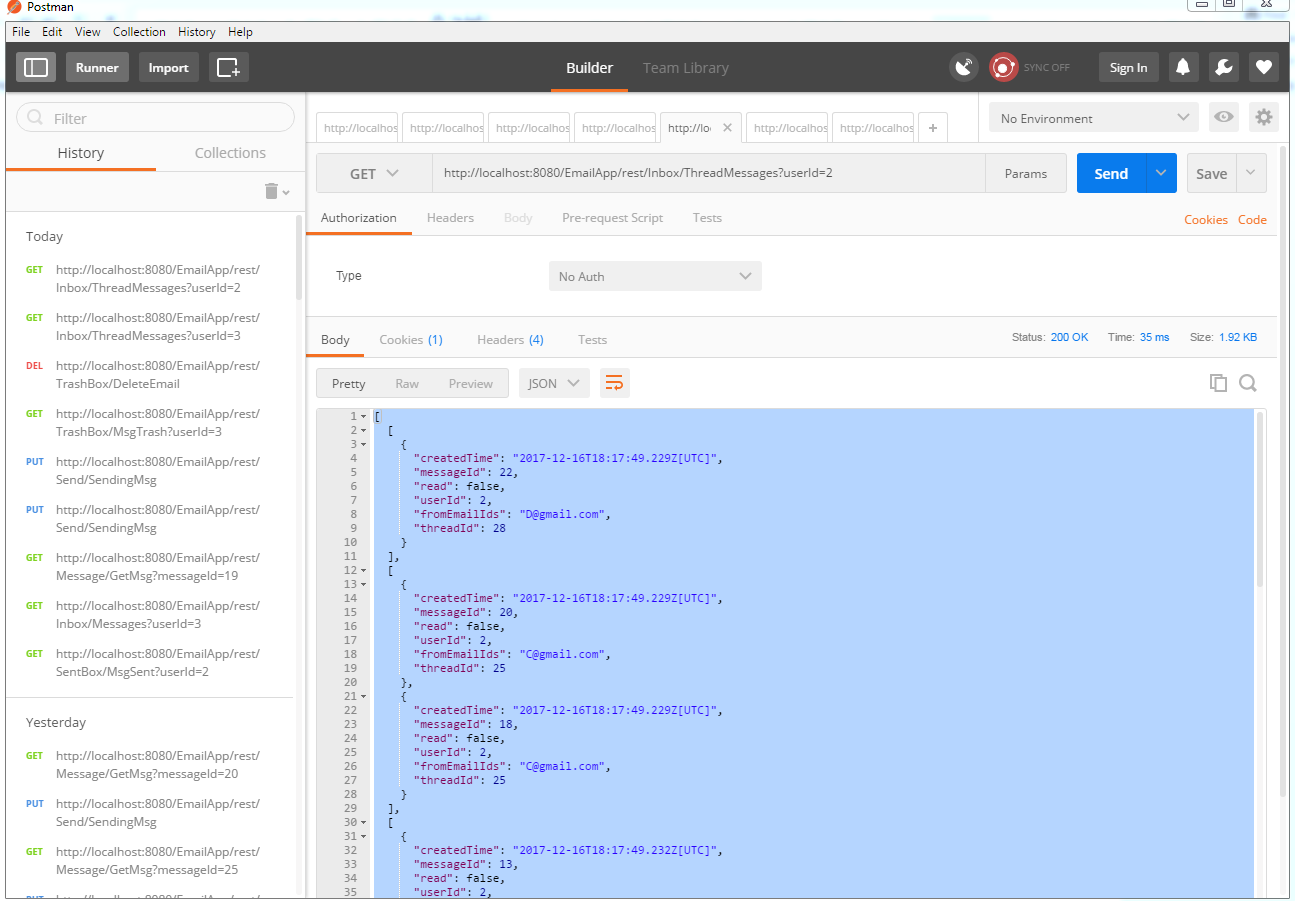
{"userId":2,"emailID":7,"boxType":"Inbox"}

Identifying the userid, emailed and whether it belongs to Inbox/Sent/Draft



1. Thread Service: This has only been implemented for Inbox i.e. emails will be fetched in thread format i.e. all emails in 1 thread will be in one list of emails and there will be first 20 threads once you call this service.

URL: <http://localhost:8080/EmailApp/rest/Inbox/ThreadMessages?userId=3>



1. Updated Send Email Service

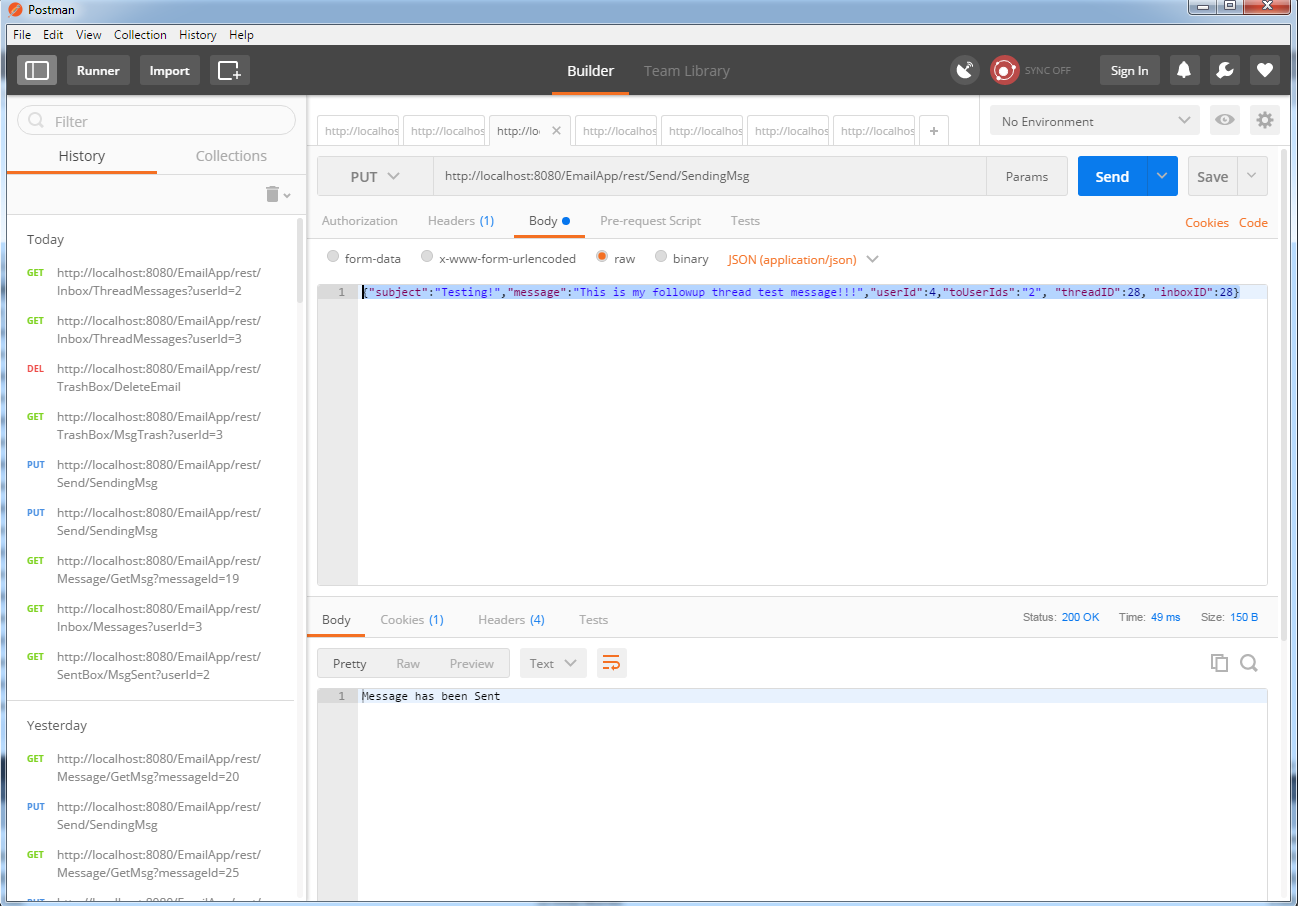
I have added new feature in Send Mail Service for thread Id and inbox ID.

If these two fields are added inbox will be updated with this thread id and the historic message i.e. last message for same user id with same thread Id will be appended to this new message.

{"subject":"Testing!","message":"This is my followup thread test message!!!","userId":4,"toUserIds":"2", "threadID":28, "inboxID":28}

If these fields are not present thread Id will be same as inbox id thus being unique automatically.

{"subject":"Testing!","message":"This is my test added thread test message!!!","userId":6,"toUserIds":"4"}



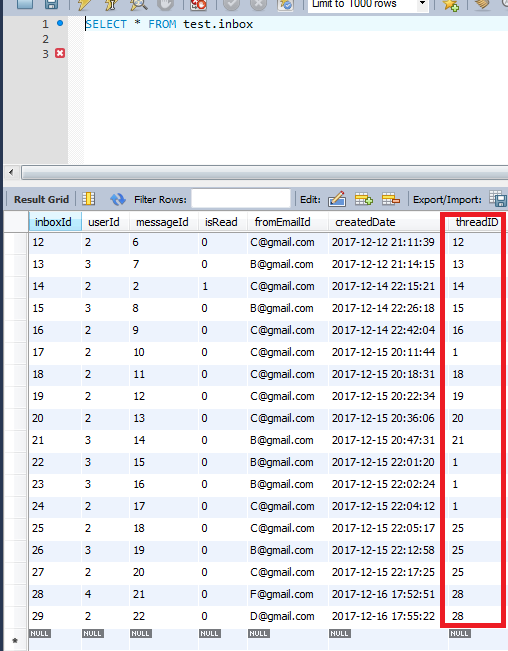


Figure Thread Id added to Table



Figure :Multiple Level Appending