

Step Wise Approach to fetching data from HubSpot CRM

Step 1 - First created the Python 3.8 environment with the dependencies

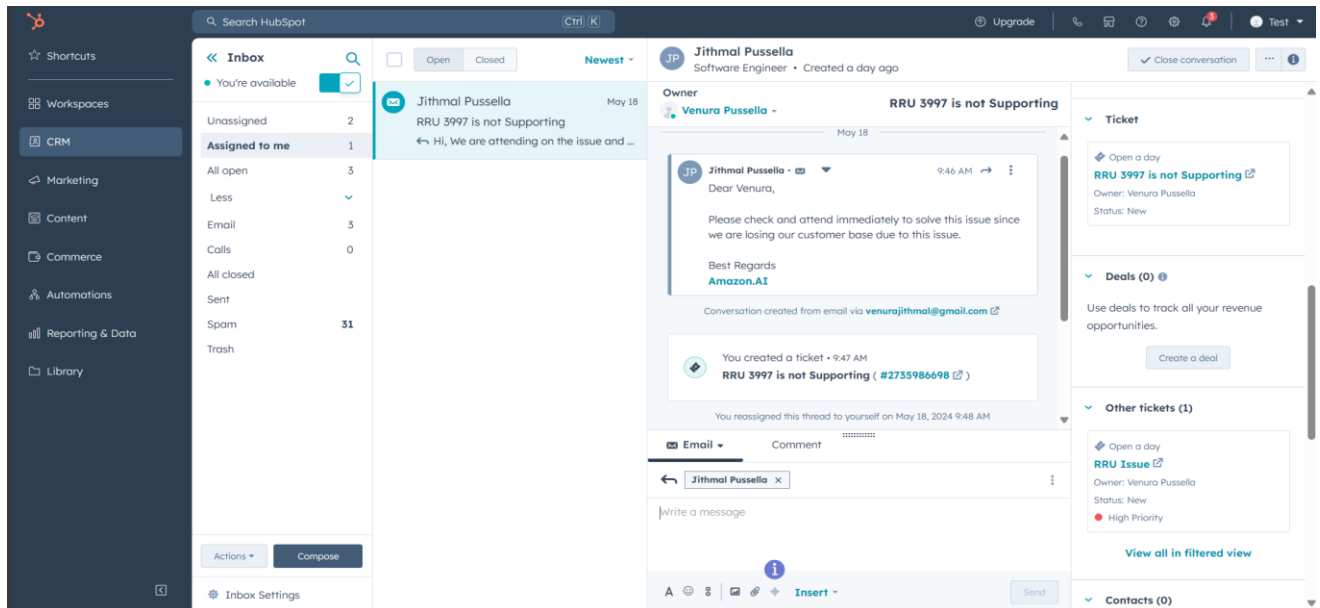
```
Administrator: Anaconda Prompt

(base) C:\Windows\System32>cd C:\Users\Venura Pussella\OneDrive\Desktop\github\Hubspot_API Project

(base) C:\Users\Venura Pussella\OneDrive\Desktop\github\Hubspot_API Project>conda activate hubspot

(hubspot) C:\Users\Venura Pussella\OneDrive\Desktop\github\Hubspot_API Project>pip freeze
```

Step 2 - Then created my personal CRM using Hubspot

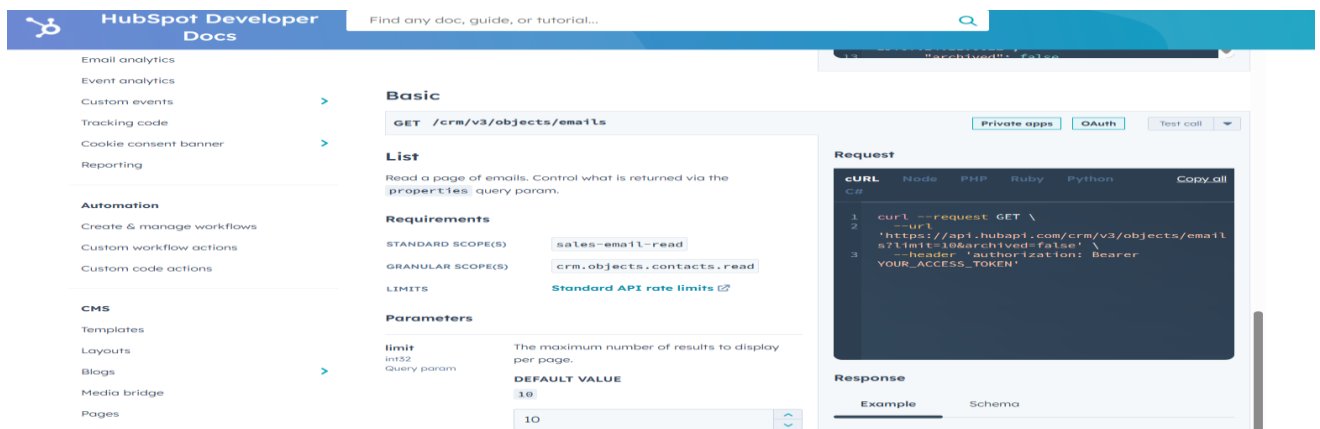


Step -3 - Check for the endpoints to interact with in order to fetch the data from emails and tickets assigned.

1. Emails

[Engagements | Email \(hubspot.com\)](https://developers.hubspot.com/docs/api/email)

First use the basic get method to list what is returned via the properties with below endpoint.



Then use post method with search endpoint and filtered the specific date range of emails and send query parameters to retrieve data in Json format.

Search

POST /crm/v3/objects/emails/search

Requirements

STANDARD SCOPE(S)

GRANULAR SCOPE(S)

LIMITS Exempt from the [per-ten-second API call limit](#)

Parameters

query
string
Body param

limit *
int32
Body param

after *
string
Body param

Request

cURL [Node](#) [PHP](#) [Ruby](#) [Python](#) [Copy all](#)

```
C#  
1 curl --request POST \  
2   --url https://api.hubapi.com/crm/v3/objects/emails/search \  
3   --header 'authorization: Bearer YOUR_ACCESS_TOKEN' \  
4   --header 'content-type: application/json' \  
5   --data '{  
6     "query": "string",  
7     "limit": 0,  
8     "after": "string",  
9     "sorts": [  
10      "string"  
11    ],  
12     "properties": [  
13
```

Response

[Example](#) [Schema](#)

HTTP 200

2. Tickets

[CRM API | Tickets \(hubspot.com\)](#)

Like above (in emails), first used get method with below basic endpoint to see what's returned.

Analytics & Events

- Email analytics
- Event analytics
- Custom events >
- Tracking code
- Cookie consent banner >
- Reporting

Automation

- Create & manage workflows
- Custom workflow actions
- Custom code actions

CMS

- Templates
- Layouts
- Blogs >
- Media bridge

Basic

GET /crm/v3/objects/tickets

List

Read a page of tickets. Control what is returned via the `properties` query param.

Requirements

STANDARD SCOPE(S)

GRANULAR SCOPE(S)

LIMITS [Standard API rate limits](#)

Parameters

limit
int32
Query param The maximum number of results to display per page.
DEFAULT VALUE

after The paging cursor token of the last

Request

cURL [Node](#) [PHP](#) [Ruby](#) [Python](#) [Copy all](#)

```
C#  
1 curl --request GET \  
2   --url https://api.hubapi.com/crm/v3/objects/tickets?limit=10&archived=false \  
3   --header 'authorization: Bearer YOUR_ACCESS_TOKEN'
```

Response

[Example](#) [Schema](#)

HTTP 200

JSON [Copy all](#)

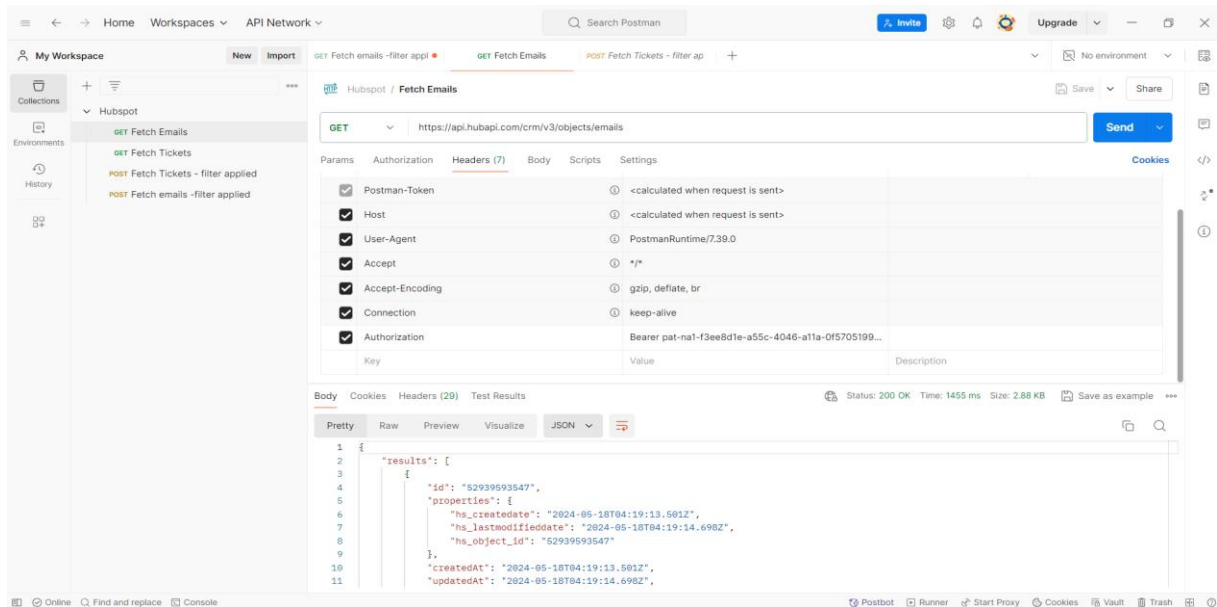
Then use post method with search endpoint and filtered the high priority tickets and send query parameters to retrieve data in Json format.

The screenshot shows the HubSpot Developer Docs page for the `POST /crm/v3/objects/tickets/search` endpoint. The left sidebar lists various CRM categories, with 'Tickets' selected under 'Objects'. The main content area displays the endpoint details, including requirements (Standard Scope: tickets, Granular Scope: None, Limits: Exempt from the per-ten-second API call limit), and parameters (query: string, limit: *). A 'Request' section on the right shows a cURL command for a POST request to the endpoint, including headers for authorization and content-type, and a JSON body with query, limit, after, and sorts parameters.

Step - 4 – Created private app and get API token by selecting the necessary objects mentioned in “Standard scopes” in each end point according to documentation.

The screenshot shows the HubSpot Developer Console for a CRM app. The left sidebar lists various settings, with 'Private Apps' selected under 'Account Setup'. The main content area displays the app's overview, including a graph of API calls (Number of API Calls vs. Date) showing a peak in activity around May 18, 2024. The graph shows client errors (4XX) in orange and succeeded calls in blue. A table of API call recent errors is also visible, showing 11 errors, with the last one occurring 5 hours ago. The 'API CALL RECENT ERRORS' section shows a red dot indicating 4XX errors.

Step 5 - Verified above mentioned endpoints using Postman



Step -6 - Define Python Script to cover the objectives mentioned.

1. Load Environment variables – dotenv (.env)
2. Header configurations - HTTP headers for API request including authorization token
3. Function to extract and transform data into readable format

extract_tickets ()

- Sends a POST request to the HubSpot tickets API endpoint.
- Checks if the response is successful or raise an exception if it fails.
- Filters tickets with high priority and extracts properties such as *hs_object_id*, *subject*, *content*, *hs_ticket_category*, and *hs_ticket_priority*.
- Converts the extracted ticket data into a Pandas DataFrame.
- Returns: Pandas DataFrame

extract_emails()

- Sends a POST request to the HubSpot emails API endpoint.
 - Checks if the response is successful or raise an exception if it fails.
 - Filters emails created within a specific time range and extracts properties such as *hs_object_id*, *hs_createdate*, *hs_email_subject*, *hs_email_text*, *hs_email_status*.
 - Converts the extracted email data into a Pandas Data Frame.
 - Returns: Pandas DataFrame
4. Load parsed data in to an Excel file.
 - Email & ticket data is stored in separate sheets

	A	B	C	D	E
1	create_date	email_id	email_subject	email_text	email_status
2	2024-05-19T10:11:54.715Z	52969711229	Promotion	Dear jithmal,CheckBR VJ	SENT
3	2024-05-19T10:17:35.505Z	52969687992	Re: Promotion	Dear Venura,Issue solved thanksBR,PussellaOn Sun, May 19	SENT
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					

< >

emails tickets +

:

	A	B	C	D	E
1	ticket_id	subject	content	ticket_category	ticket_priority
2	2735499893	RRU Issue			HIGH
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					
20					
21					
22					
23					
24					
25					
26					

< >

emails tickets +