

A
User Guide
ON

Central Mail API

(Version-4.1.1)

Developed By
Krushil Gajjar & Team



X-BYTE
ENTERPRISE CRAWLING

*Bodakdev, Ahmedabad – 380054
India*

TABLE OF CONTENTS

1. Introduction	
1.1. Project Profile	4
1.2. Core Components.....	5
2. Development	
2.1. Payload.....	7
2.2. Endpoints.....	10
2.3. Working Model.....	10
2.4. Working project screenshot.....	13
2.5. Addon.....	14
3. Conclusion	16
4. Reference	18

1. Introduction

1.1 Project Profile

- In this section, we will discuss the profile of project, The project is typically based on Global Mail API that takes some user input & generate the Mail draft that displays graphically data analysis this is fully automated mail system with chunk of inputs
- In this project we are having an API that accept Pull request & return the Mail in response
- In this API we are about to passing the Post request that takes a Payload as user input, in post request we are having 2 different params that works differently.
- Here are the four different endpoints with their specific URL
 - http://192.168.0.39:5020/composite_graph/
 - http://192.168.0.39:5020/comparison_graph/
 - http://192.168.0.39:5020/delta_table/
 - http://192.168.0.39:5020/exclusive_graph/
- The above URLs are having below different params
 - Composite graph
 - The composite graph will take multiple files/tables from user that works on multiple files & generate the graph as per inputs
 - Comparison graph
 - The comparison graph will take one file with having multiple header option that is dynamic, user will have to enter the headers of a particular file, if user leave the headers empty than the graph consider all available header in file/table & generate the mail accordingly
 - Delta table
 - This table having limited central table format, it will take count date wise from fix table format of SQL only
 - Exclusive Graph
 - Now onwards this endpoint having no limits of use, just you need to provide the DataFrame, the graph and table dynamically set along with the DataFrame. Also many more Dynamic parameters are available, lets understand brief below using payload.

1.2 Core Components

- Framework: Python Fast API
- Backend script: Python 3.8
- Database: MySQL/MongoDB
- OS: Linux Ubuntu
- Deployment: Gunicorn service
- Webserver: nginx

2. Development

2.1 Payload

here is a payload that we are taking from the user & generate a graph accordingly

```
payload = {"project_name": 'Project_Name',
          "type": 'sql',
          "host": '192.168.1.42',
          "username": 'root',
          "password": 'xbyte',
          "database_name": 'upscal_delivery_mail',
          "today_table": 'upscal_count_master',
          "yesterday_table": '',
          "label_name": '[]',
          "days_to_compare": "5",
          "dynamic_table": 'no',
          "table_prefix": 'productdata',
          "table_date_format": "%Y_%m_%d",
          "header_flag": "no",
          "header_arr": "[]",
          "file_name": 'File_Name',
          "Client_name": 'Client_Name',
          "Alert_code": "123",
          "graph_type": 'bar', #TODO - Types of graph :: bar, pie & line
          "graph_flag": 'yes',
          "to_arr": "['ashad.khira.xbyte@gmail.com']",
          "cc_arr": "['ashad.khira.xbyte@gmail.com']",
          "bcc_arr": "[]",
          "extra_params_arr": "['File Delivery Path: <Your File Path here>']",
          "file_base64": '',
          "client_io_flag": 'yes',
}
```

- **Payload for Exclusive Graph:**

```

payload = {
    "df_main": json.loads(df_main.to_json(orient='records')), #TODO - Required
    "df_name": 'Date-wise Count', #TODO - Required
    "df_header": json.loads(df_main.to_json(orient='records')), #TODO - Optional
    "df_header_name": 'My Header Table', #TODO - Optional
    "label_arr": df_today.columns.to_list(), #TODO - Required
    "project_name": "Test-Project", #TODO - Required
    "client_name": "Ashad Khira", #TODO - Required
    "graph_flag": "Yes", #TODO - Required
    "client_io_flag": "No", #TODO - Required
    "Alert_code": "6969", #TODO - Required
    "graph_type": "bar", #TODO - Types = Bar, Line, Radar, Pie, Doughnut, Polar, Scatter,
    Bubble, Radial Gauge / Meter, Speedometer, Box Plots / violin
    "to_arr": ['ashad.khira.xbyte@gmail.com'], #TODO - Optional
    "cc_arr": ['krushil.gajjar.xbyte@gmail.com'], #TODO - Optional
    "bcc_arr": "", #TODO - Optional
    "header_box_label": ['Today Ratio', 'Yesterday Ratio'], #TODO - Optional
    "header_box_value": ['99%', '100%'] #TODO - Optional
}

payload_2 = {
    "df_main": [json.loads(df_main.to_json(orient='records')),
    json.loads(df_main.to_json(orient='records'))], #TODO - For multiple table & graph
    # "df_main": json.loads(df_main.to_json(orient='records')), #TODO - For single table &
    graph
    "df_name": ['Monthly Data Report', 'Monthly Data Report 2'], #TODO - For Multiple
    table & graph
    # "df_name": 'Monthly Data Report', #TODO - For single table & graph
    "label_arr": [df_today.columns.to_list(), df_today.columns.to_list()], #TODO - For
    Multiple table & graph
    # "label_arr": df_today.columns.to_list(), #TODO - For Single table & graph
    "project_name": "Test-Project", #TODO - Required
    "client_name": "Ashad Khira", #TODO - Required
    "graph_flag": "Yes", #TODO - Required
    "client_io_flag": "No", #TODO - Required
    "Alert_code": "6969", #TODO - Required
    "graph_type": "bar", #TODO - Types = Bar, Line, Radar, Pie, Doughnut, Polar, Scatter,
    Bubble, Radial Gauge / Meter, Speedometer, Box Plots / violin
    "to_arr": ['ashad.khira.xbyte@gmail.com'], #TODO - Optional
    "cc_arr": "", #TODO - Optional
    "bcc_arr": "", #TODO - Optional
    "header_box_label": ['Today Ratio', 'Yesterday Ratio'], #TODO - Optional
    "header_box_value": ['99%', '100%'], #TODO - Optional
    "extra_params_arr": "['Status - File Uploaded']",
}

```

df_main :- This param specified for main DataFrame that you are going to pass, also there is a another option for multiple graph & multiple table, to achieve that just give multiple DataFrames in a list

df_name :- The name will be add the title for your given DataFrame, as same as multiple DataFrame, the data frame should have multiple name in a list if more than one DataFrame are pass.

df_header :- The header table are now also dynamically set along with DataFrame

df_header_name :- The title should provide for the same

label_arr :- Now you have to provide the additional label for the graph purpose along with your DataFrame. as same as multiple DataFrame, the data frame should have multiple label_arr in a list if more than one DataFrame are pass.

project_name :- Same as per ordinary payload

client_name :- Same as per ordinary payload

graph_flag :- Same as per ordinary payload

client_io_flag :- Same as per ordinary payload

Alert_code :- Same as per ordinary payload

graph_type :- Same as per ordinary payload

to_arr, cc_arr, bcc_arr :- Same as per ordinary payload (Now CC & BCC will optional)

header_box_label, header_box_value :- This parameter specified for the box that are displayed on head of mail, it should contain label & value for the same

extra_params_arr :- The other message you can pass using 'extra_params_arr' variable

2.2 Endpoints

In our central mail API, we have a Post method request that takes payload from the user,

In this URL we are passing the four endpoints for the different inputs & output

- I. http://192.168.0.39:5020/composite_graph/
- II. http://192.168.0.39:5020/comparison_graph/
- III. http://192.168.0.39:5020/delta_table/
- IV. http://192.168.0.39:5020/exclusive_graph/

As per above, we are having the same URL with four different endpoints

- I. composite_graph
- II. comparison_graph
- III. delta_table
- IV. exclusive_table

2.3 Working Model

In Python we are simply request to the API & get the response as same as we are sending the post request in python using request module with the particular payload syntax

As the payload given, we describe each payload's arguments below:

“project_name”: - This parameter specify the Project Name

“type”: - This parameter specify the Database which can be Mongo or SQL

“host”: - This parameter specify the host of database

“username”: - This parameter specify the username of database

“password”: - This parameter specify the username of database

“database_name”: - This parameter specify the Database Name

“today_table”: - This parameter specify the today’s table which you want to need graph

“yesterday_table”: - This parameter specify the yesterday’s table which you want to need graph

“label_arr”: - This parameter specify that label of files that require in only Composite graph

“dynamic_table”: - This parameter specify the flag for table is known or not known that should be yes or no

“table_prefix”: - This parameter specify the table prefix if the dynamic table is “Yes”

“table_date_format”: - This parameter specify the table date format if the dynamic table is “Yes”

“header_flag”: - This parameter specify that if you want header wise comparison even if in composites graph

“header_arr”: - This parameter specify the headers array in only comparison parameter of API

“file_name”: - This parameter specify the file name which keep the subject in mail format

“Client_name”: - This parameter specifies the file Client name which keep the Client Name in mail format

“Alert_code”: - This parameter specified the Project/Alert code for specific project

“graph_type”: - This parameter specify the type of the graph which can be “bar”, “line”, “pie”

“graph_flag”: - This parameter specify the flag that graph to be shown or not, it should be “yes” or “no”

“to_arr”: - This parameter specify that which we person we should

keep on “TO” mail

“**cc_arr**”: - This parameter specify that which we person we should keep on “CC” mail

“**bcc_arr**”: - This parameter specify that which we person we should keep on “BCC” mail

“**extra_params_arr**”: - This parameter specify the extra params or argument which user want to keep in mail

“**file_base64**”: - Now onwards you can attach the file in mail also using this params, this params accept the file content the code for it given below

“**client_io_flag**”: - Now giving Yes & no you can change ‘from’ Mail id (receiver’s mail ID)

*(Note: if you are giving data in this param you have to specify the file name in **file_name** params)*

Code for file_base64:

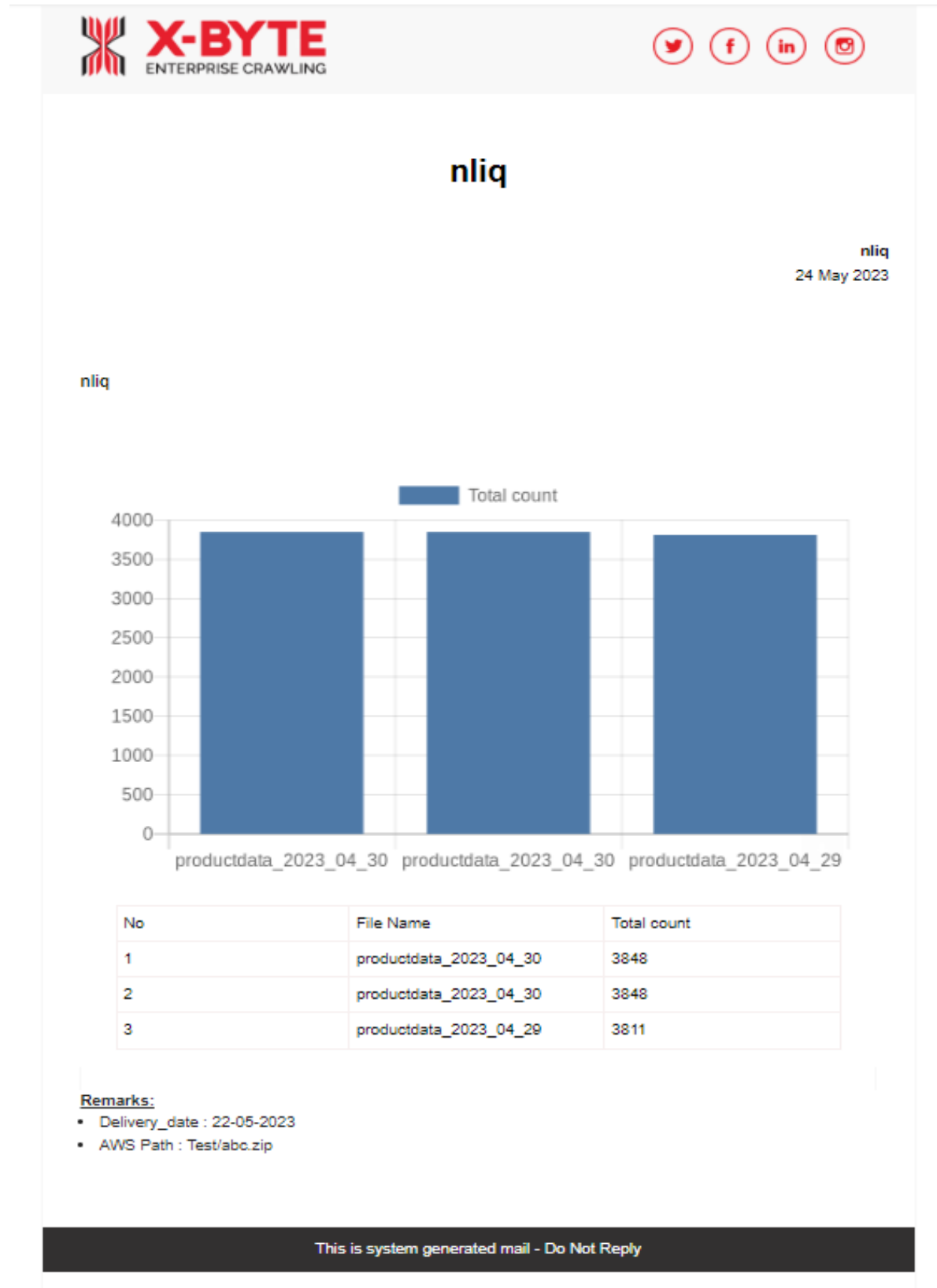
```
with open(<Your File Path>, 'rb') as file:
    # exs = file_path.split()
    file_data = file.read()
    file_base64 = base64.b64encode(file_data).decode('utf-8')
```

2.4 Addon

When sending client emails (with the flag client_io_flag: "Yes"), the email will be sent from the address alerts@xbyte.io to the designated recipients in the "To" and "CC" fields. However, the "BCC" field will exclude alerts@xbyte.io, and a copy of the email will instead be sent from a [*@gmail.com](mailto: *@gmail.com) address.

2.5 Working project screenshot

Composite Graph



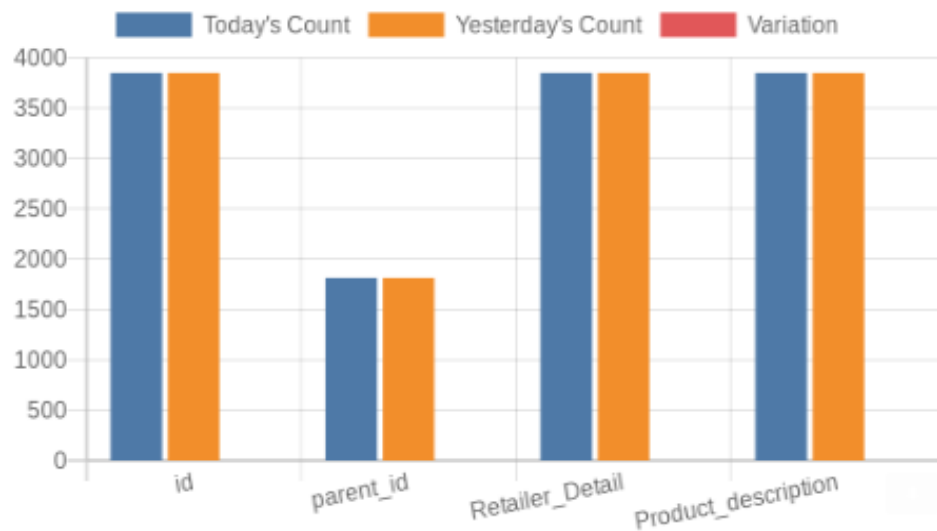
Comparison Graph



Project_Name

Client_Name
24 May 2023

File_Name



No	Header Name	Today's Count	Yesterday's Count	Variation
1	id	3848	3848	0
2	parent_id	1813	1813	0
3	Retailer_Detail	3848	3848	0
4	Product_description	3848	3848	0

Remarks:

- File Delivery Path: path/to/upload

This is system generated mail - Do Not Reply

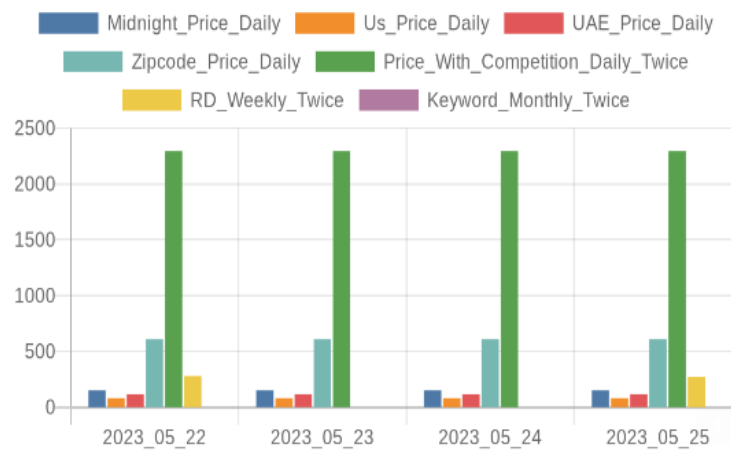
Delta Table Graph



Project_Name

Client_Name
26 May 2023

Project_Name



No	Date	Midnight_Price_Daily	Us_Price_Daily	UAE_Price_Daily	Zipcode_Price_Daily	Price_With_Competition_Daily_Twice	RD_Weekly_Twice	Keyword_Monthly_Twice
1	2023_05_22	152	80	116	608	2293	279	
2	2023_05_23	152	80	116	608	2293		
3	2023_05_24	152	80	116	608	2293		
4	2023_05_25	152	80	116	608	2293	272	

3. Conclusion

- In this project, we successfully developed the Central Mail API using FastAPI in Python, aiming to streamline and automate mail-related operations within our organization. The project objectives were to create a robust and efficient API that would handle various mail-related tasks.

4. Reference

- Herewith attached a Dropbox folder link that contains two “.py” file that refers to the Post request to the API using given Payload
 - **Comparison Graph Py file: -**
https://www.dropbox.com/scl/fi/lm2mykp2yfwjd0hu15mj/api_comparison_graph_request.py?rlkey=y18nzperaxpc007qike4ns2z6&dl=0
 - **Composite Graph Py file: -**
https://www.dropbox.com/scl/fi/j14un9jqhwr06uuqo1wfi/api_composite_graph_request.py?rlkey=227ateidutiftpdo6y8qvdjyr&dl=0
 - **Delta Table Py file: -**
https://www.dropbox.com/scl/fi/hdur4fso5e5do3vgdwa76/api_delta_table_request.py?rlkey=i4r15f1amjine10zjxumfaroa&dl=0
 - **Exclusive Table Py file: -**
https://www.dropbox.com/scl/fi/rbw6wbld60mmwbzre6iop/api_exclusive_graph_request.py?rlkey=uiyugat6o5kb6slc1xen05wrk&dl=0