

TymeBoxx Documentation

Overview:

TymeBoxx is a social media consolidation and Generative AI solution, first being launched as a B2B API service, then as a B2C solution in an app (progressive web app) called TymeBoxx. You can see an intro to the product [here](#) and the buildout/growth strategy [here](#) (early draft).

The tech stack is largely Python (data consolidation, normalization, table entry and consumption into MongoDB), and to date we have incorporated Twitter/X, Instagram and Facebook, with plans to add TikTok and several others. The Gen AI tweet generator piece is powered by ChatGPT 4 using OpenAI in Azure, and it has the ability to create new tweets / posts from your old content by analyzing the user's past tweets based on the sentiments, biases, tone & type of words/vocabulary, ideology & perspective.

It'll different than all the other social media solutions out there because it will not be ad-driven (paid for service), will allow people to consolidate all their content in one place, ask questions of their data (NLP/NLQ services in Power BI), have no harmful algorithms and be Web3 compliant (be able to support Decentralized IDs, digital wallets, etc.) as it'll be built on Jack Dorsey's [Blue Sky Web](#) / [AT Protocol](#). What we are building will also pave the way for data self-monetization.

Key Features and Components:

1. Data Download Instructions for Facebook, Instagram and Twitter/X

- To download data from Facebook: [Facebook In-App Data Request User Guide.pdf](#)
- To download data from Instagram: [Instagram In-App Data Request User Guide.pdf](#)
- To download data from Twitter: [Twitter In-App Data Request User Guide.pdf](#)

2. UniqueId creation

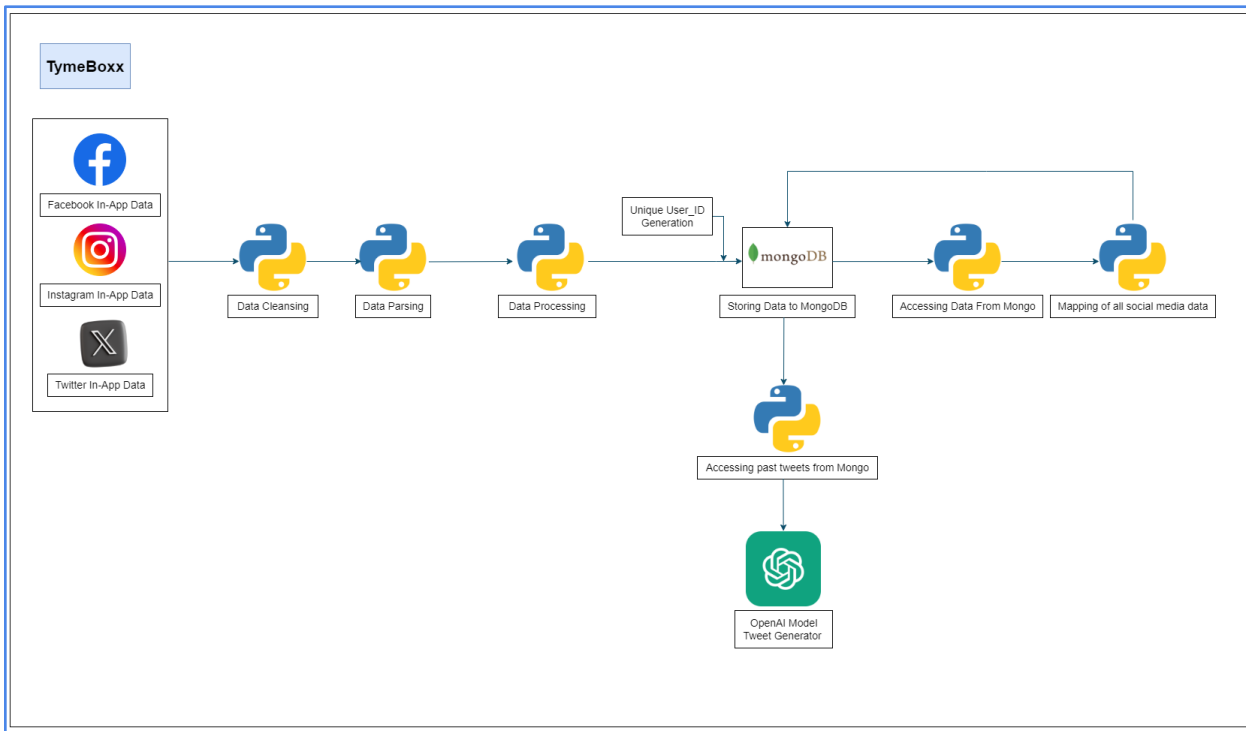
3. Data Extraction and Data Processing from these huge zip files

4. Storing Data to the Database

5. Mapping of the data from 3 social media platforms

6. Tweet Generator OpenAI model

Architecture Diagram:



Technical Specifications:

Our system will systematically download data from three distinct platforms to initiate the data-retrieving process. Subsequently, the coding phase includes the development of 13 Application Programming Interfaces (APIs).

The initial API is designed to generate a unique user ID for each user, providing a foundational identifier for further data processing. Following this, the subsequent three APIs are dedicated to extracting data from the zip files which contain the data from a user's data requests from the above-mentioned platforms (pulled from a designated file location). This extraction process includes automatic categorization of specific files, removal of unwanted fields, and conversion of .js files to .json format. The resulting data is then stored in a MongoDB database, with separate databases and collections for each platform (Facebook, Instagram, and Twitter/X).

An additional API is responsible for retrieving desired data from the MongoDB database. This involves accessing connection data from all three platforms. Subsequently, a separate API is employed to map and store the connection data from the three platforms into MongoDB.

The subsequent six APIs mirror the earlier mentioned process, focusing on the extraction of likes-based data, profile-based data, and text/post-based data from all three platforms. These APIs map and store the extracted data in the corresponding MongoDB databases.

Generative AI - Tweet Generator :

The final API integrates Generative AI capabilities to generate new tweets based on the user's historical twitter/X data with the sentiments, biases, tone & type of words/vocabulary, Ideology & perspective. This innovative feature adds a layer of dynamic content creation to the system, enhancing user engagement and interaction.

Usage and Operations:

There are a total of 13 API's

Method	Endpoint
GET	generate_tymeboxx_id
POST	extract_and_store_fb_files
POST	extract_and_store_insta_files
POST	extract_and_store_twitter_files
POST	extract_social_connection_data
GET	mapped_connection_data
POST	extract_social_media_likes
POST	mapped_likes_and_reactions
GET	get_profile_data_all_platforms
GET	mapped_profile_data
POST	extract_social_media_post_data
GET	text_post_mapped
POST	get_post_by_tweet_generator

API 1: generate_TymeBoxx_id

To create a unique ID (TymeBoxx_id) for the user.

The generate_tymeboxx_id API enables the creation of unique identification codes, known as "TymeBoxx IDs", providing a secure and efficient way to uniquely identify entities within a specific system or application.

GET request url : http://20.51.249.20/generate_tymeboxx_id/

The screenshot shows a REST client interface for the endpoint `generate_tymeboxx_id`. The request method is `GET` and the URL is `http://20.51.249.20/generate_tymeboxx_id/`. The response status is `200 OK` with a response time of `544 ms` and a size of `347 B`. The response body is displayed in JSON format, showing a single object with a `Tymeboxx_ID` field containing the value `"TB85440cb8-d5b5-45dc-a0c0-854551679308"`.

Key	Value	Description	...	Bulk Edit
Key	Value	Description		

```
1 {
2   "Tymeboxx_ID": "TB85440cb8-d5b5-45dc-a0c0-854551679308"
3 }
```

API 2: extract_and_store_fb_files

To extract the particular files from the zip file.

POST request url : http://20.51.249.20/extract_and_store_fb_files/

Request Body :

- form-data
 1. Key : zip_file (select 'File' from dropdown)
Value : "Upload your zip file here"
 2. Key : TymeBoxx_ID
Value : "Paste your TymeBoxx_ID generated via API 1"

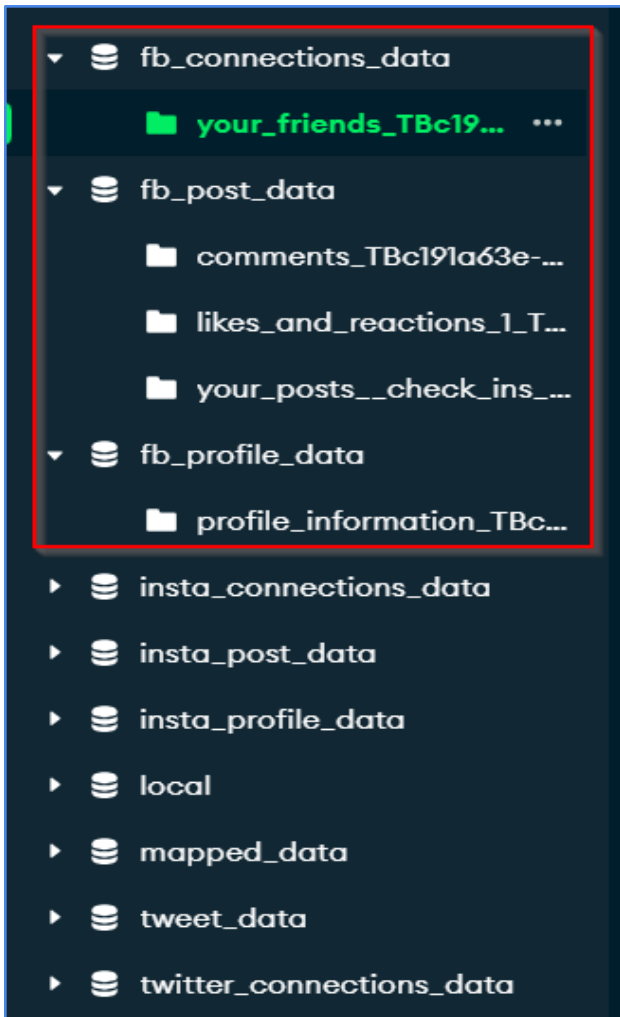
The screenshot shows a REST client interface for the endpoint `http://20.51.249.20/extract_and_store_fb_files/`. The request is a POST with form-data. The body contains two fields: `zip_file` (File) and `Tymeboxx_ID` (Text). The response is a JSON object with a message: `"message": "Files extracted and data inserted into MongoDB successfully."`

Key	Value	Description
zip_file	File	SteveLeMay-FacebookDownload-Jan132...
Tymeboxx_ID	Text	TB85440cb8-d5b5-45dc-a0c0-854551679308

```
1 {
2   "message": "Files extracted and data inserted into MongoDB successfully."
3 }
```

- After sending the request this API starts the extraction of specific files as mentioned below:
 1. profile_information.json
 2. your_hobbies.json
 3. your_freinds.json
 4. comments.json.
 5. your_videos.json
 6. likes_and_reactions_1.json
 7. Your_posts__check_ins__photos_and_videos_1.json
- We have given our MongoDB credentials in the codebase so that after extracting the files it will store them in MongoDB as different databases and collections.
- Before storing any file, it also cleans the file, and parses the data with the required formats and labels.
- Here it will store the collection name as a "collection name_TymeBoxxID."
-

This is how the database and collection will be stored in MongoDB.



API 3: extract_and_store_insta_files

To extract the particular files from the zip file.

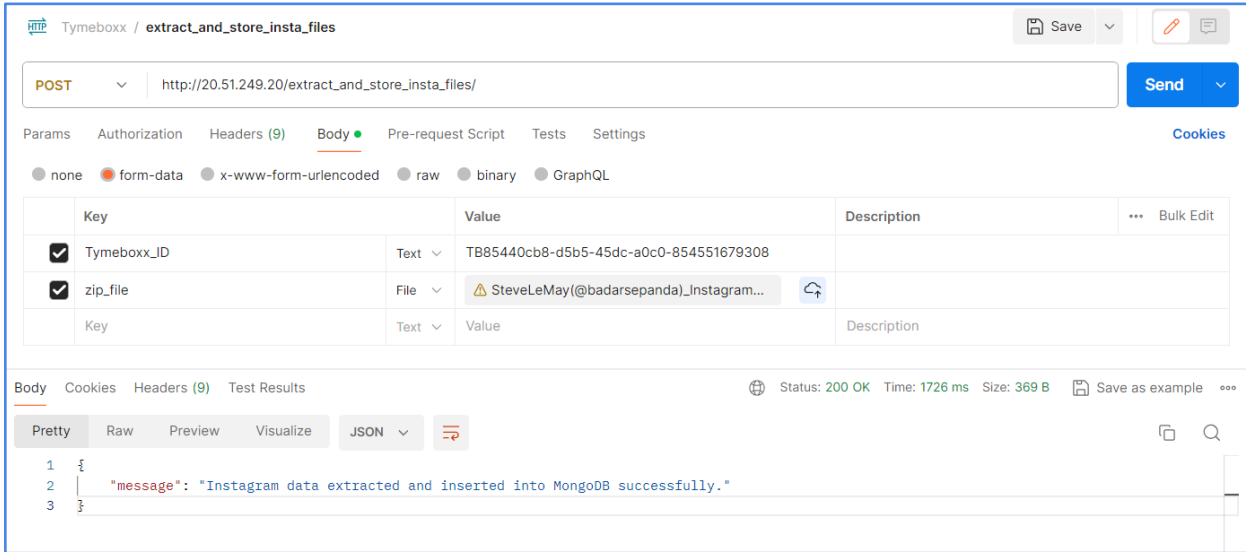
POST request url : http://20.51.249.20/extract_and_store_insta_files/

Request Body :

- form-data

1. Key : zip_file (select 'File' from dropdown)
Value : "Upload your zip file here"

2. Key : TymeBoxx_ID
Value : "Paste your TymeBoxx_ID generated via API 1"



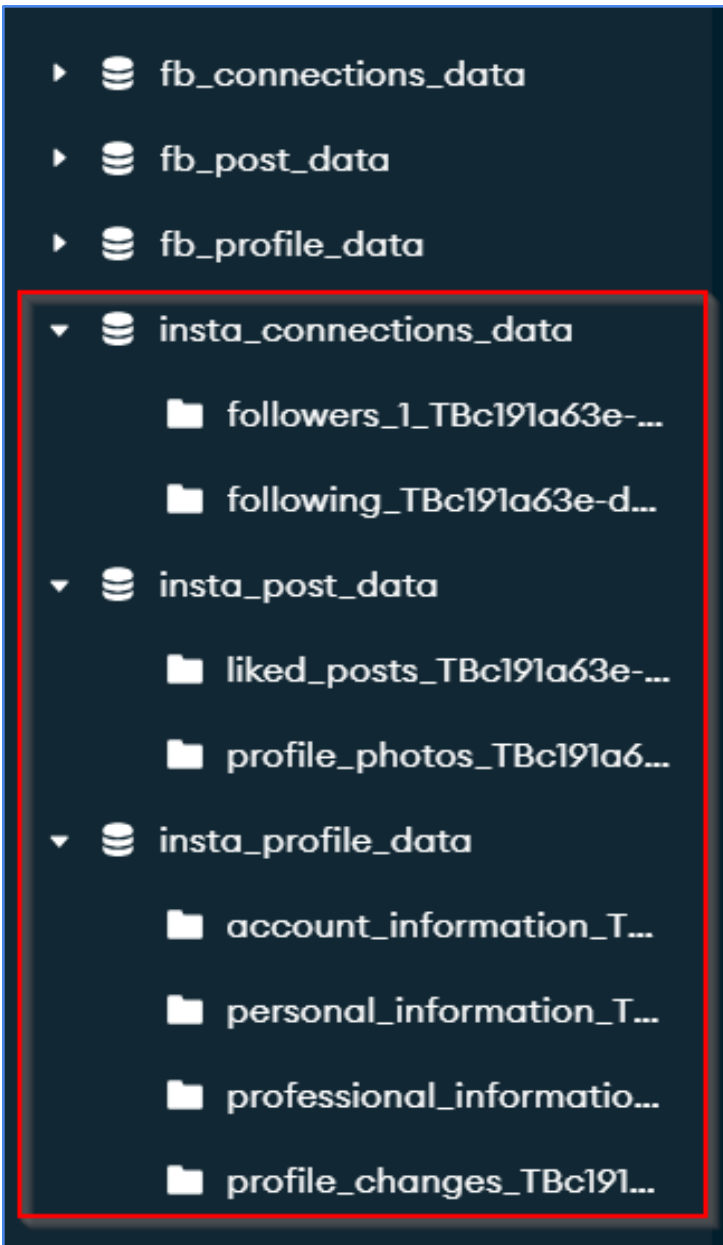
After sending the request this API starts the extraction of specific files i.e personal_information.json

1. Professional_information.json
2. Linked_meta_accounts.json
3. Profile_changes.json
4. Account_information.json
5. Followers_1.json
6. Following.json
7. Posts_1.json
8. Profile_photos.json
9. Reels.json
10. liked_posts.json
11. Liked_comments.json

We have given our MongoDB credentials in the code based on that after extracting the files it will store them in MongoDB as different databases and collections. Before storing any file, it also cleans the file and parses the data with the required formats and labels.

Here it will store the collection name as a "collection name_TymeBoxxID".

This is how the database and collection will be stored in MongoDB.



API 4: extract_and_store_twitter_files.

To extract the particular files from the zip file.

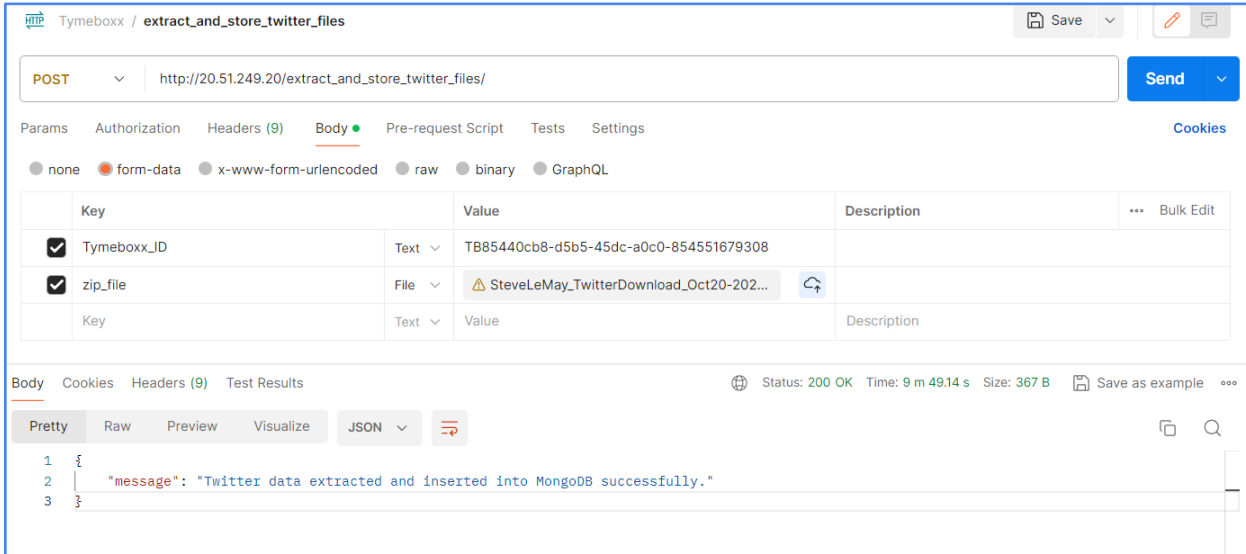
POST request url : http://20.51.249.20/extract_and_store_twitter_files/

Request Body :

- form-data

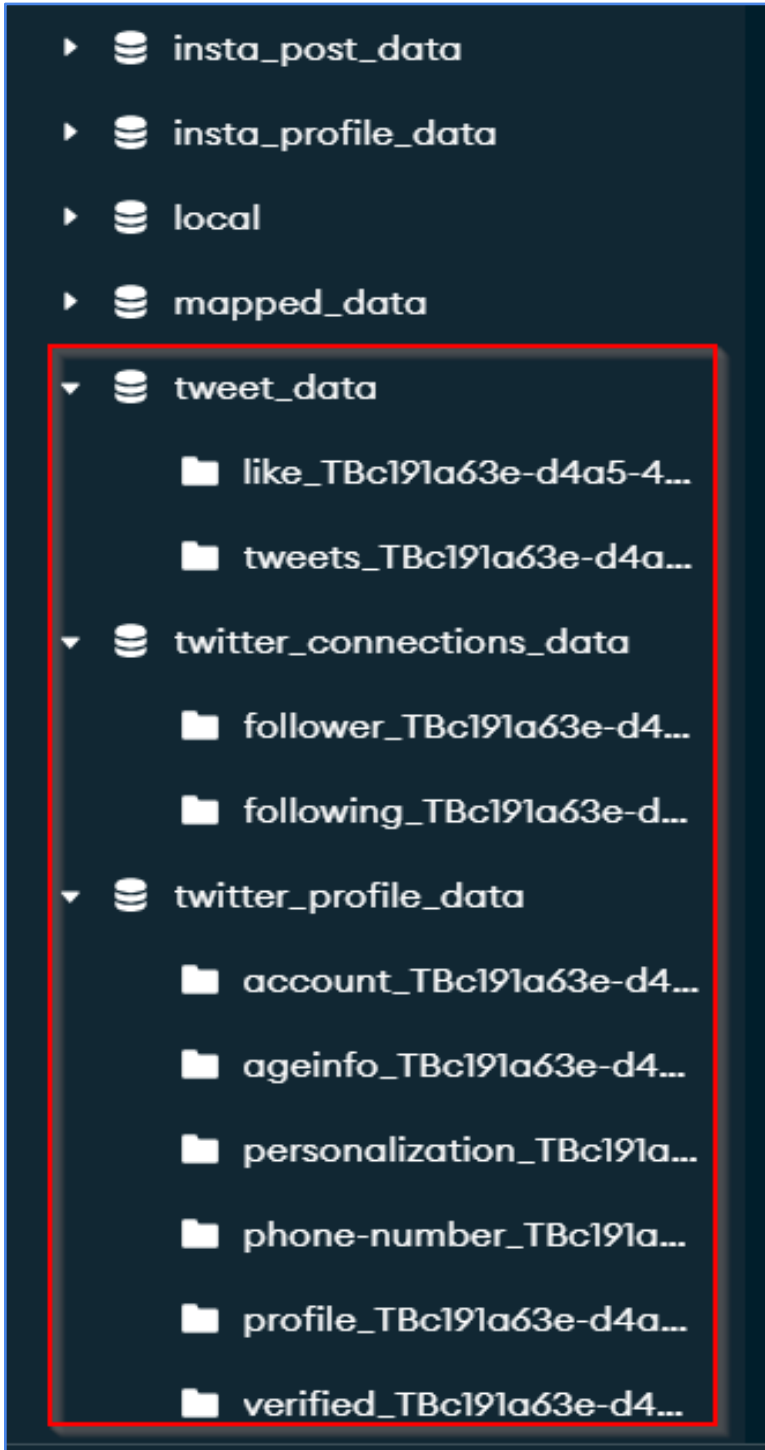
1. Key : zip_file (select 'File' from dropdown)
Value : "Upload your zip file here"

2. Key : TymeBoxx_ID
Value : "Paste your TymeBoxx_ID generated via API 1"



- After sending the request this API starts the extraction of specific files i.e
 1. account.js
 2. ageinfo.js
 3. Personalization.js
 4. Phone-number.js
 5. Profile.js
 6. Verified.js
 7. Follower.js
 8. Following.js
 9. like.js
 10. tweets.js.
- In this API first, we are converting the .js files into .json files because in Twitter zip files we will not get JSON files directly.
- After this API will do the preprocessing of the tweets.json file because here we need only particular fields from this file.
- We have given our MongoDB credentials in the code based on that after extracting the files it will store them in MongoDB as different databases and collections.
- Here it will store the collection name as a "collection name_TymeBoxxID".

This is how the database and collection will be stored in MongoDB.



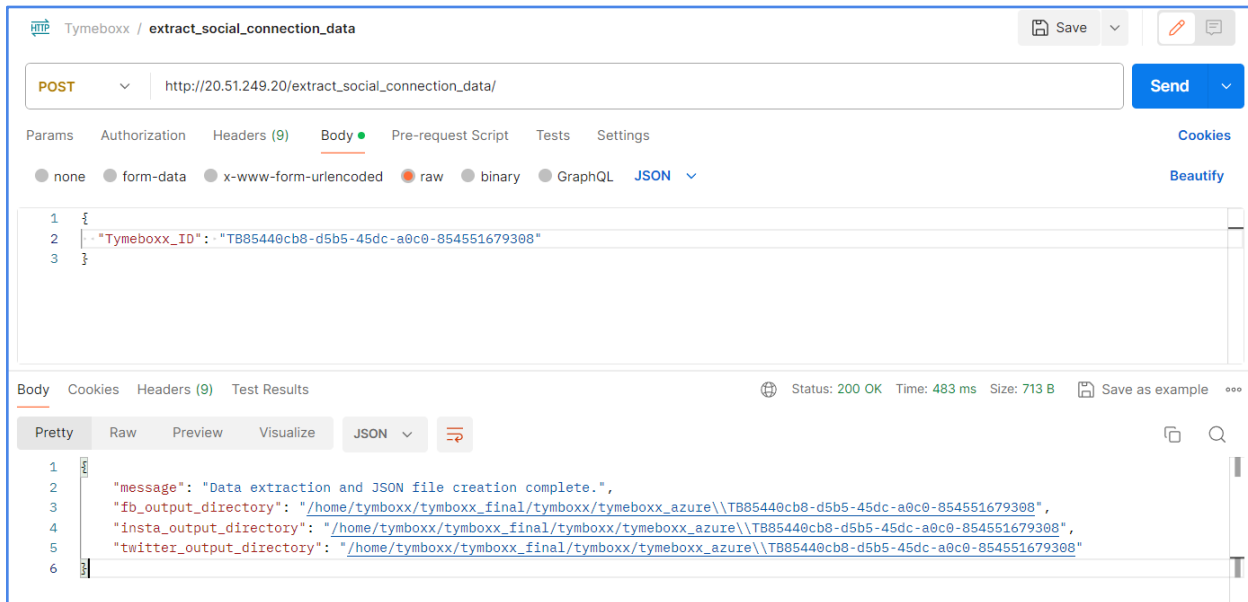
API 5: extract_social_media_connection_data

To extract the desired data from the MongoDB database.

POST request url : http://20.51.249.20/extract_social_connection_data/

Request Body :

```
{
  "Tymeboxx_ID": "Paste your TymeBoxx_ID generated via API 1"
}
```



The screenshot shows a REST client interface for a request to `http://20.51.249.20/extract_social_connection_data/`. The request is a POST with a JSON body containing a `Tymeboxx_ID`. The response is a 200 OK status with a JSON body containing a success message and output directories for Facebook, Instagram, and Twitter.

```
POST http://20.51.249.20/extract_social_connection_data/

{
  "Tymeboxx_ID": "TB85440cb8-d5b5-45dc-a0c0-854551679308"
}
```

```
{
  "message": "Data extraction and JSON file creation complete.",
  "fb_output_directory": "/home/tymboxx/tymboxx_final/tymboxx/tymeboxx_azure\\TB85440cb8-d5b5-45dc-a0c0-854551679308",
  "insta_output_directory": "/home/tymboxx/tymboxx_final/tymboxx/tymeboxx_azure\\TB85440cb8-d5b5-45dc-a0c0-854551679308",
  "twitter_output_directory": "/home/tymboxx/tymboxx_final/tymboxx/tymeboxx_azure\\TB85440cb8-d5b5-45dc-a0c0-854551679308"
}
```

- MongoDB credentials are mentioned.
- Here we are going to extract the data from your_friends_{TymeBoxx_id} from **fb_connections_data** database following_{TymeBoxx_id} followers_1_{TymeBoxx_id} from insta_connections_data database following_{TymeBoxx_id} follower_{TymeBoxx_id} from **twitter_connections_data** database.
- After this API will extract the specific fields from the collection and create separate JSON files for each platform.

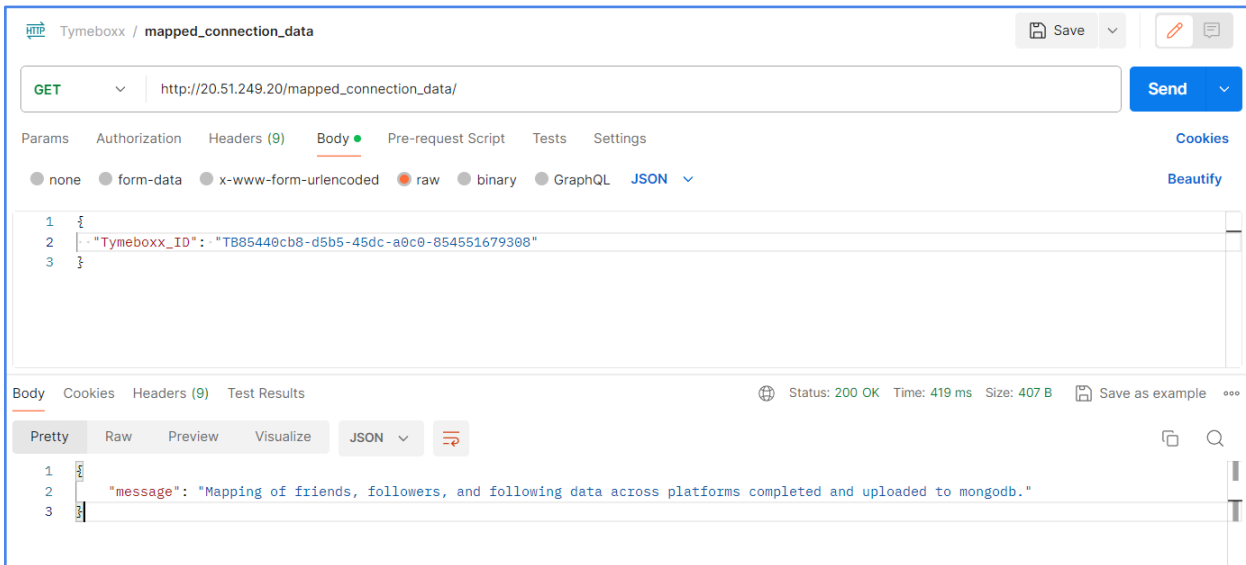
API 6: mapped_connection_data

To map all 3 platforms' connection data.

GET request url : http://20.51.249.20/mapped_connection_data/

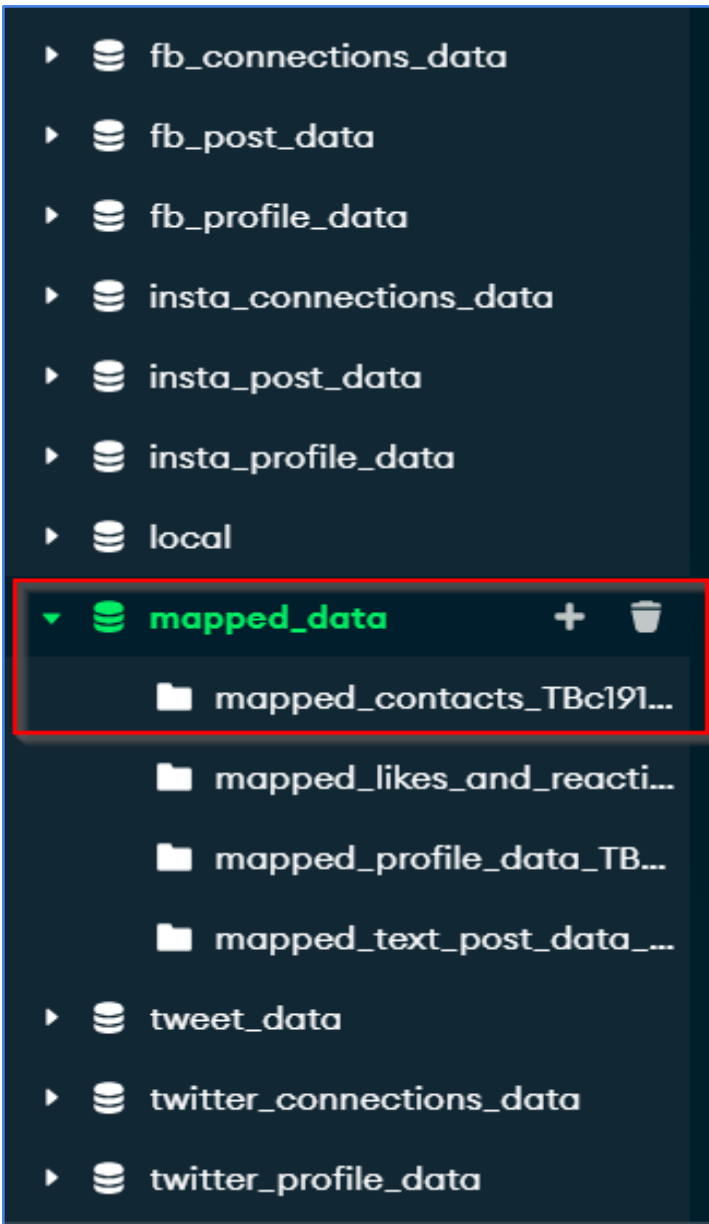
Request Body :

```
{
  "Tymeboxx_ID": "Paste your TymeBoxx_ID generated via API 1"
}
```



- Here the 1st step is to load the data from JSON files for each platform.
- And then merging the data from Facebook, Instagram and Twitter.
- After merging it will create one JSON file and save it in our output directory.
- At the end, the merged JSON file will be stored in MongoDB as a `mapped_data` database & `mapped_contacts_{TymeBoxx_id}` collection.

This is how the database and collection will be stored in MongoDB.



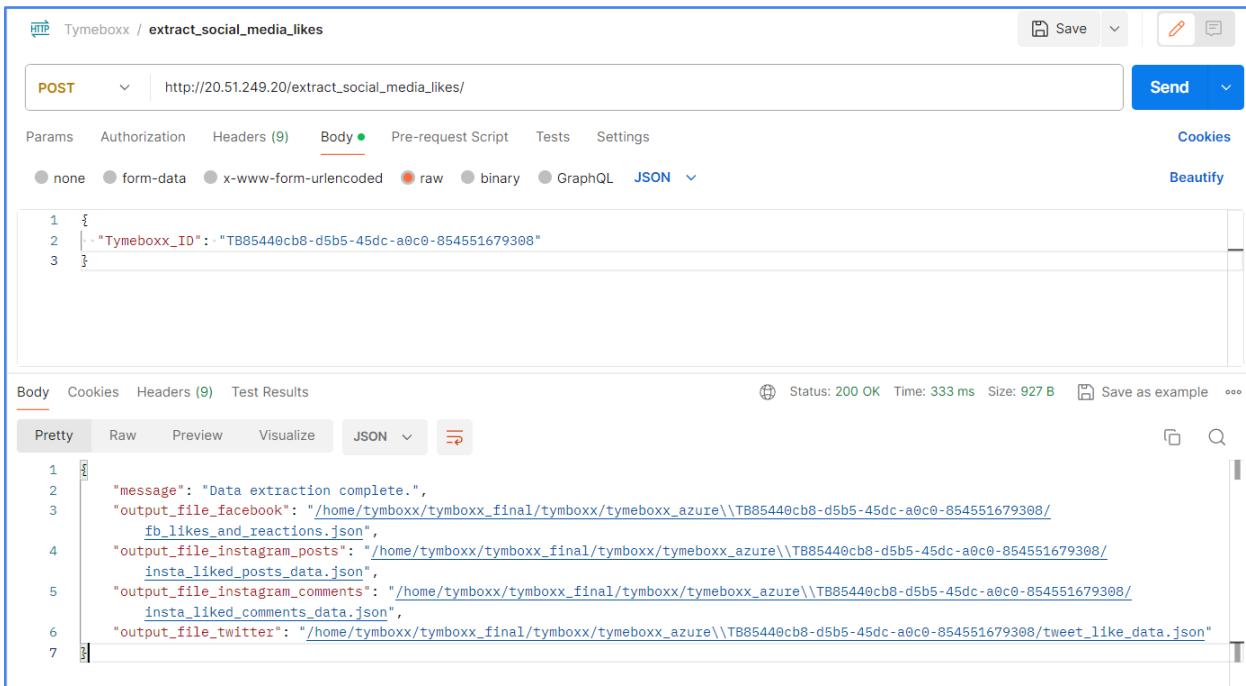
API 7: extract_social_media_likes

To extract the desired data from the MongoDB database.

POST request url : http://20.51.249.20/extract_social_media_likes/

Request Body :

```
{
  "Tymeboxx_ID": "Paste your TymeBoxx_ID generated via API 1"
}
```



- MongoDB credentials are mentioned.
- Here we are going to extract the data from `likes_and_reactions_1_{TymeBoxx_id}` from `fb_post_data` database, `liked_posts_{TymeBoxx_id}`, `liked_comments_{TymeBoxx_id}` from `insta_post_data` database, `like_{TymeBoxx_id}` from `tweet_data` database.
- After this API will extract the specific fields from the collection and create separate JSON files for each platform.

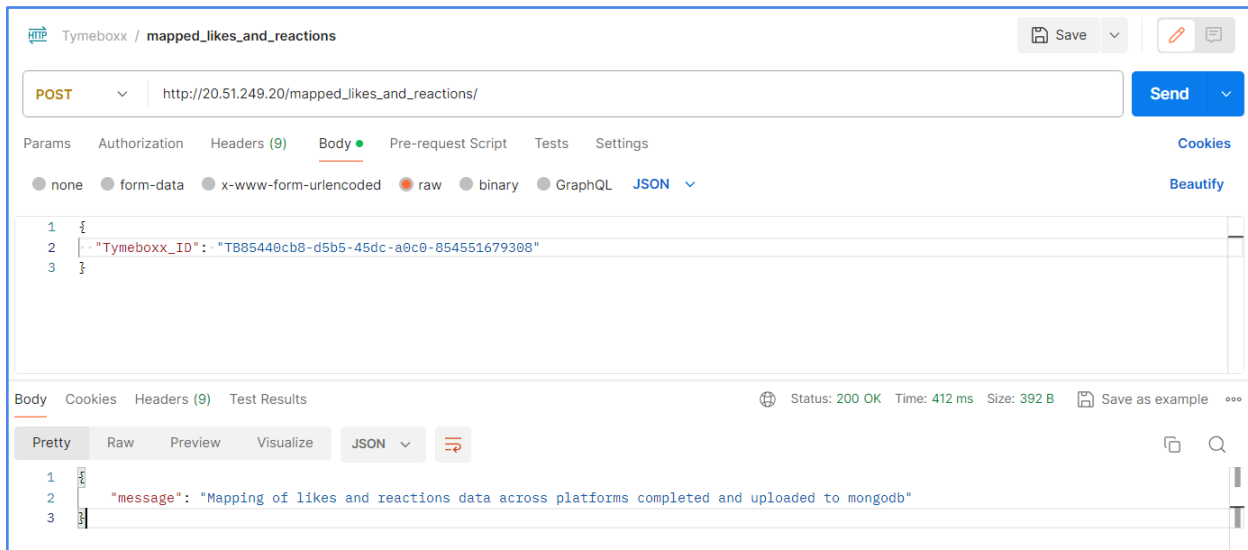
API 8: mapped_likes_and_reactions

To map all 3 platforms like data.

POST request url : http://20.51.249.20/mapped_likes_and_reactions/

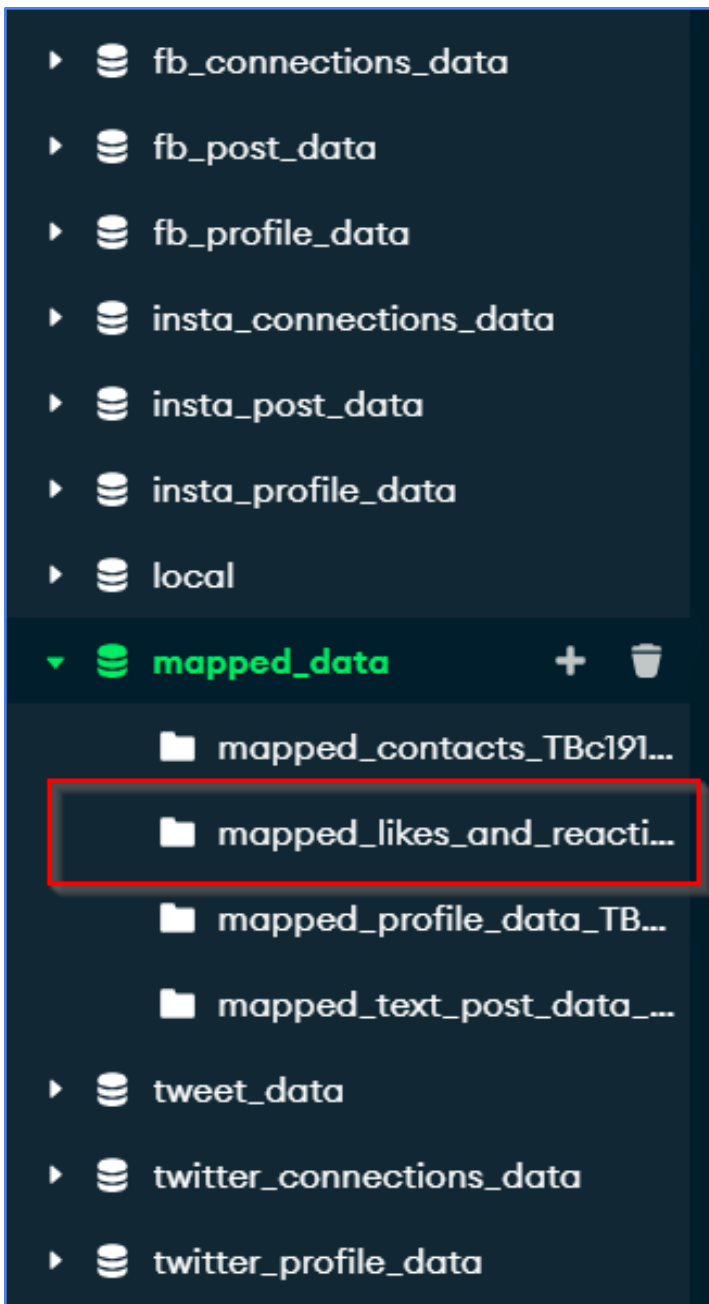
Request Body :

```
{
  "Tymeboxx_ID": "Paste your TymeBoxx_ID generated via API 1"
}
```



- Here the 1st step is to load the data from JSON files for each platform.
- And then merging the data from Facebook, Instagram and Twitter.
- After merging it will create one JSON file and save it in our output directory.
- At the end, the merged JSON file will be stored in MongoDB as a `mapped_data` database, `mapped_likes_and_reactions_{TymeBoxx_id}` collection.

This is how the database and collection will be stored in MongoDB.



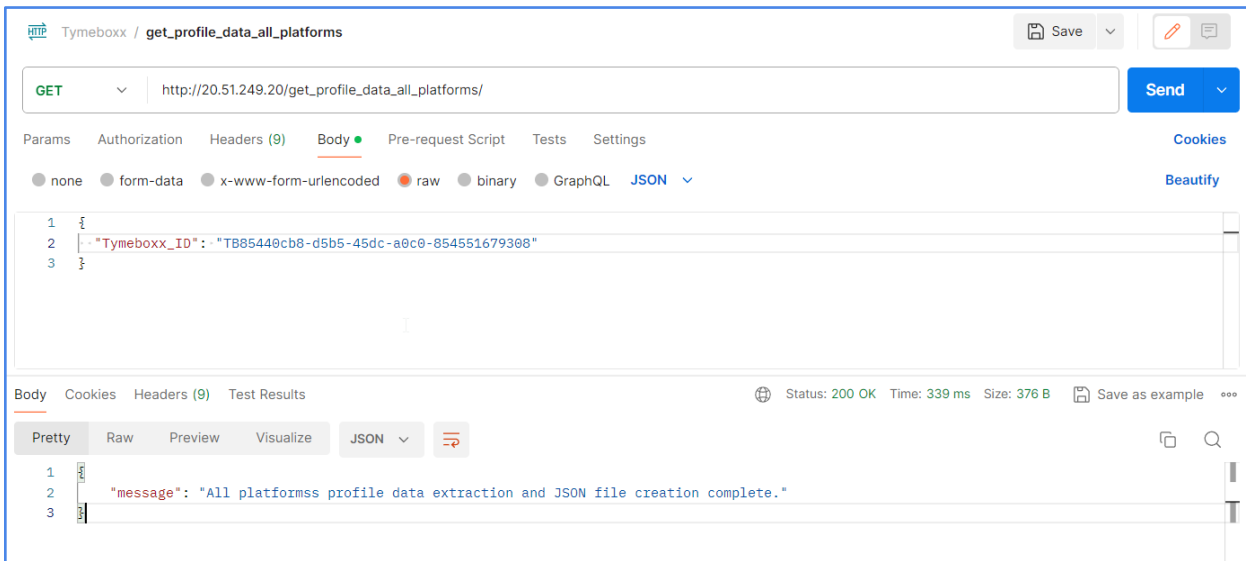
API 9: get_profile_data_all_platforms

To extract the desired data from the MongoDB database.

GET request url : http://20.51.249.20/get_profile_data_all_platforms/

Request Body :

```
{
  "Tymeboxx_ID": "Paste your TymeBoxx_ID generated via API 1"
}
```



- MongoDB credentials are mentioned.
- Here we are going to extract the data from
 1. `profile_information_{TymeBoxx_id}` from **fb_profile_data** database
 2. `personal_information_{TymeBoxx_id}`
 3. `account_information_{TymeBoxx_id}`
 4. `linked_meta_accounts_{TymeBoxx_id}` from **insta_profile_data** database
 5. `account_{TymeBoxx_id}`
 6. `ageinfo_{TymeBoxx_id}`
 7. `personalization_{TymeBoxx_id}`
 8. `profile_{TymeBoxx_id}`
 9. `verified_{TymeBoxx_id}` from **twitter_profile_data** database.
- After this API will extract the specific fields from the collection and create separate JSON files for each platform.

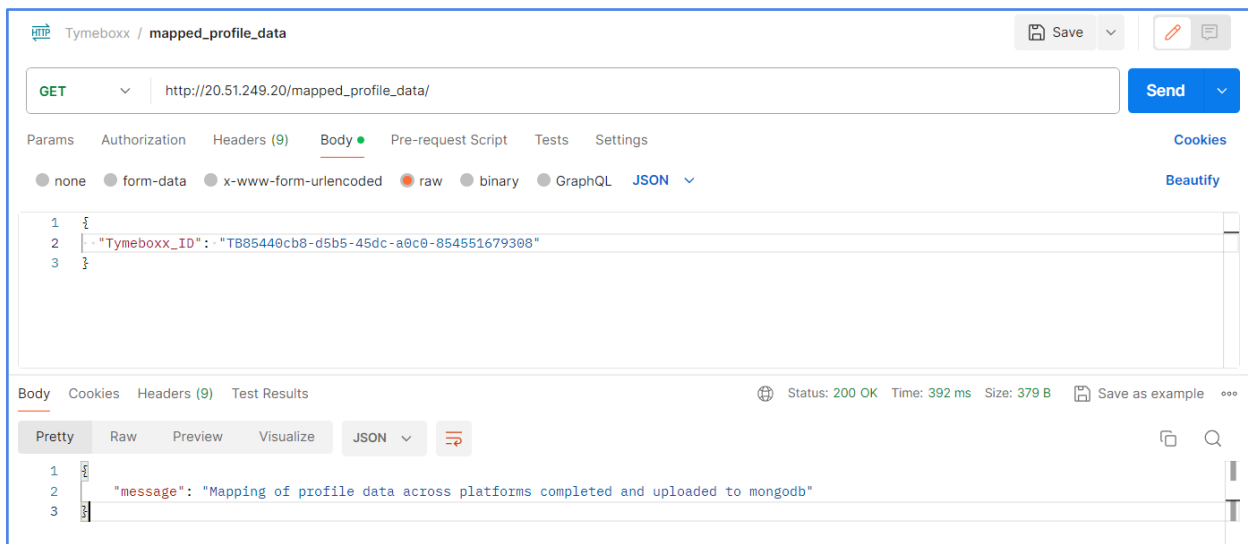
API 10: mapped_profile_data

To map all 3 platforms' profile data.

GET request url : http://20.51.249.20/mapped_profile_data/

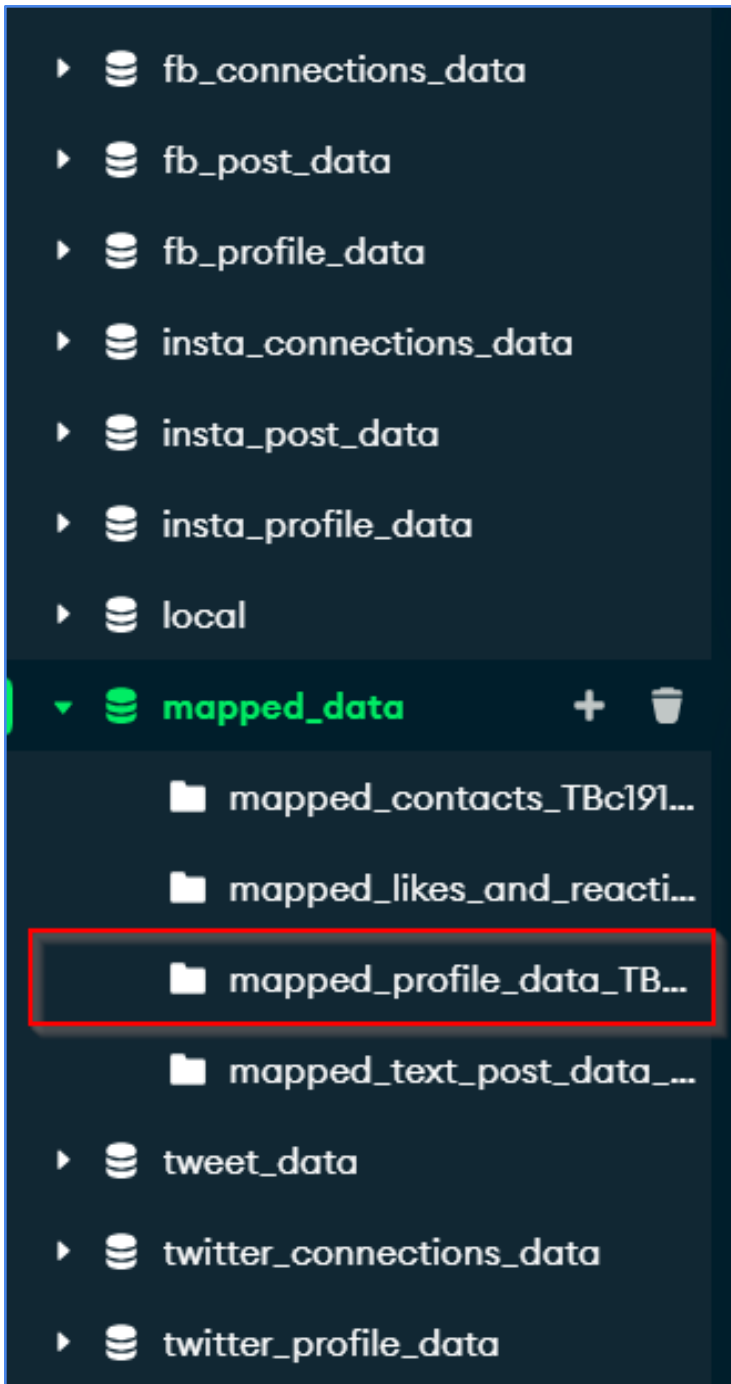
Request Body :

```
{
  "Tymeboxx_ID": "Paste your TymeBoxx_ID generated via API 1"
}
```



- Here the 1st step is to load the data from JSON files for each platform.
- And then merging the data from Facebook, Instagram and Twitter.
- After merging it will create one JSON file and save it in our output directory.
- At the end, the merged JSON file will be stored in MongoDB as a
 1. mapped_data database
 2. mapped_profile_data_{TymeBoxx_id}collection.

This is how the database and collection will be stored in MongoDB.



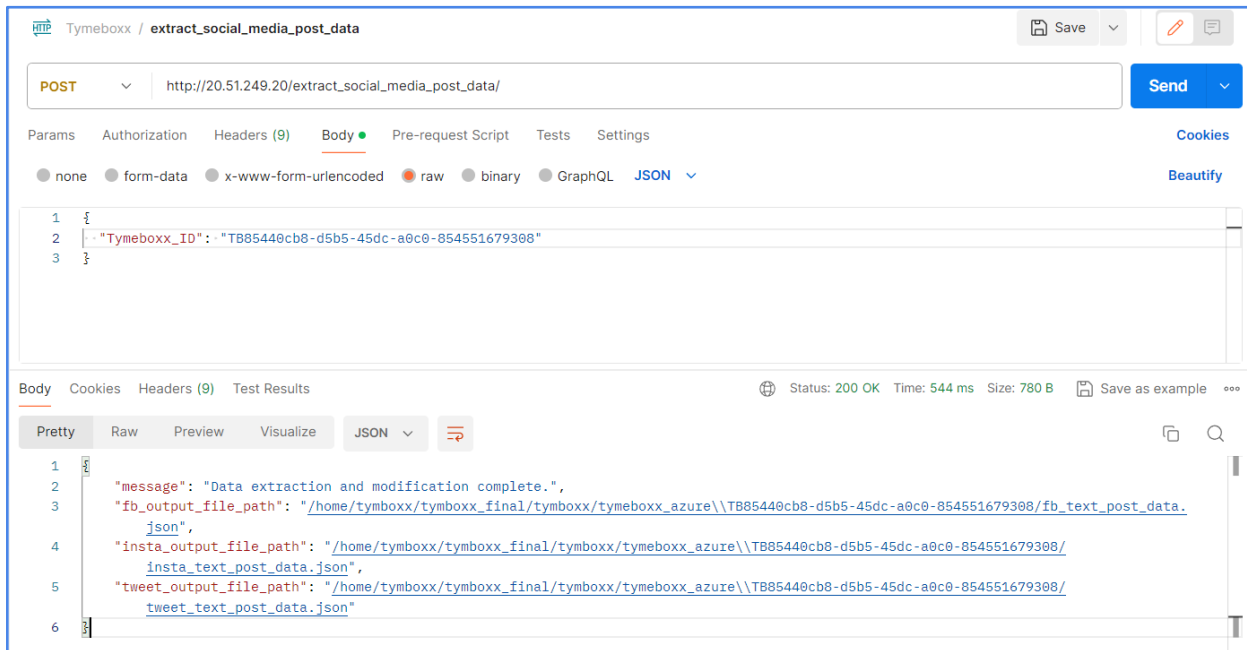
API 11: extract_social_media_post_data

To extract the desired data from the MongoDB database.

POST request url : http://20.51.249.20/extract_social_media_post_data/

Request Body :

```
{
  "Tymeboxx_ID": "Paste your TymeBoxx_ID generated via API 1"
}
```



- MongoDB credentials are mentioned.
- Here we are going to extract the data from
 1. your_posts__check_ins__photos_and_videos_1_{TymeBoxx_id} from fb_post_data database.
 2. posts_1_{TymeBoxx_id} from **insta_post_data** database.
 3. tweets_{TymeBoxx_id} from **tweet_data** database.
- After this API will extract the specific fields from the collection and create separate JSON files for each platform.

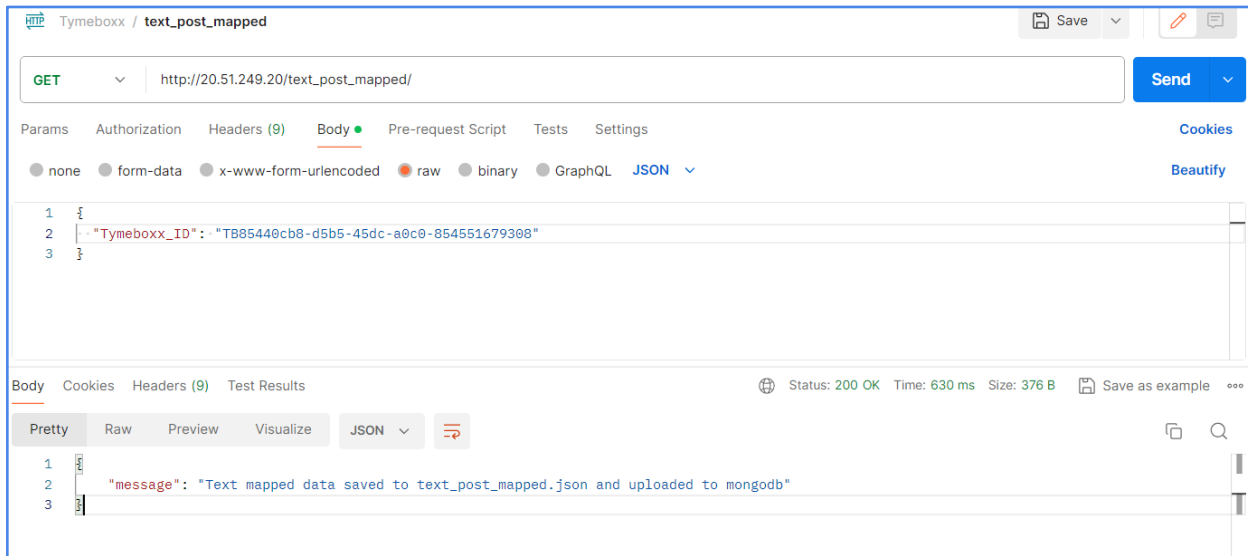
API 12: text_post_mapped

To map all 3 platforms' text_post data.

POST request url : http://20.51.249.20/text_post_mapped/

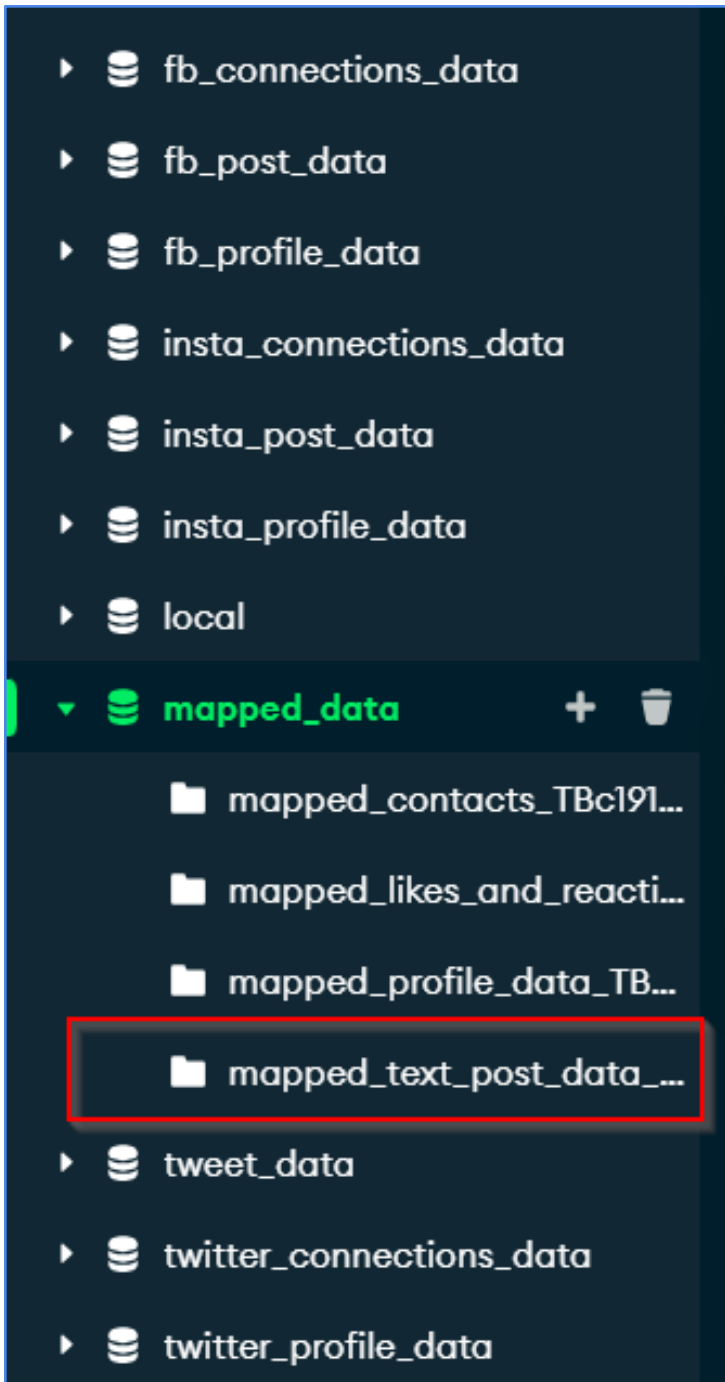
Request Body :

```
{
  "Tymeboxx_ID": "Paste your TymeBoxx_ID generated via API 1"
}
```



- Here the 1st step is to load the data from JSON files for each platform.
- And then merging the data from Facebook, Instagram and Twitter.
- After merging it will create one JSON file and save it in our output directory.
- At the end, the merged JSON file will be stored in MongoDB as a
 1. mapped_data database.
 2. mapped_text_post_data_{TymeBoxx_id} collection.

This is how the database and collection will be stored in MongoDB.



API 13: get_post_by_tweet_generator

To generate new tweets.

POST request url : http://20.51.249.20/get_post_by_tweet_generator/

Request Body :

```
{
  "Tymeboxx_ID": "Paste your TymeBoxx_ID generated via API 1",
  "tweet_timestamp": "last 6 months",
  "n": 2,
  "topic" : "Social change platforms"
}
```

1.TymeBoxx_ID.

2.tweet_timestamp - Here we can give any number of weeks, months and years.

For example - the last 5 week, last 6 month, last 2 year

3. n (Number of Tweets)

4. Topic

- This API will generate the n number of new tweets based on the topic given and past tweets of the user.
- And based on the tweet_timestamp given in the request it will access the past tweets of the user during that duration.

The screenshot shows a REST client interface for the endpoint `http://20.51.249.20/get_post_by_tweet_generator/`. The request is a POST with a JSON body:

```
{
  "Tymeboxx_ID": "TB85440cb8-d5b5-45dc-a0c0-854551679308",
  "tweet_timestamp": "last 6 month",
  "n": 2,
  "topic": "Social change platforms"
}
```

The response is a JSON object with the following structure:

```
{
  "generated_tweets": "[\"@UNWebTV Time for a paradigm shift! Decentralized IDs & blockchain could lead us to the next gen democracy. Excited for the possibilities with UN 2030 agenda! Let's empower people & bring generational wealth. Visit: https://t.co/RRsHM7WKX1 #Blockchain4Good #UN2030 🌐🌍🌎,\\n\\\"@amyschumer Ready to see democracy post-trump era? We're on it! Seeking angel investors for a revolution. Check out: https://t.co/nphqXOR19q Let's embark on a new journey together! #WhatsNext #DemocracyRedefined 🤖🚀\\\"]",
  "posts": [
    {
      "tweet_text": "@UNWebTV Hi, looking for a solution to the Palestinian plight? Go to my site https://t.co/RRsHM7WKX1 and learn about web3, Decentralized IDs, blockchain data governance, and building your own democracy + earning generational wealth from those systems. Aligns w/ UN 2030 agenda!!"
    },
    {
      "tweet_text": "@amyschumer Nice pic; you sexy B with an itch.. 🤩 I respect ur work.. wanna see what the next generation of democracy is gonna look like, post the trump idiocracy world we're currently living in?? then go to https://t.co/nphqXOR19q I'm looking for angel investors.. 🤩 #whatsnext"
    }
  ]
}
```

The client shows a status of 200 OK, a time of 11.16 s, and a size of 8.58 KB. The response is displayed in a 'Pretty' JSON view.