

API Documentation

Installing the API:

For the time being to use the API:

1. Download the API files:
`git clone https://github.com/omkmorendha/chatbot/tree/api`
2. Install the requirements:
`pip install -r requirements.txt`
3. Make an .env file and add OPENAI_API_KEY
4. Run the file by using
`python main.py`

Base URL:

`http://127.0.0.1:5000`

Authorisation Headers:

Authorisation : API_KEY

Authorisation is done via checking if the SHA-256 hash of the API_KEY exists in the `api_keys.txt` file, currently we can use "abcd", "efgh", "ijkl" as the API key.

API Functions:

1. **query_route:**
Takes a query as an input and gives a response to the query using the custom-trained chat-bot

Endpoint:

`/query`

API Call Method: POST

Parameters:

```
{  
  
    "query_input" : "Insert the query here" (required)  
  
}
```

Output:

```
{  
  
    "query_response" : "Response to the query"  
  
}
```

2. upload_webhook_route:

Takes a URL as an input and uploads its information to the data_webhooks folder as “file_name”, if the filename parameter is not found then it is automatically extracted from the URL.

Endpoint:

```
/upload-webhook
```

API Call Method: POST

Parameters:

```
{  
  
    "url" : "Insert the url here" (required)  
  
    "file_name" : "Insert the file name to be saved as here" (optional)  
  
}
```

Output:

```
{  
  
    "message" : "Upload successful"  
  
}
```

3. upload_text_route:

Takes text as an input and uploads its information to the data folder as “file_name”, if the filename parameter is not found then it is appended to the “uploaded_text.txt” file.

Endpoint:

```
/upload-text
```

API Call Method: POST

Parameters:

```
{  
  
  "text" : "Insert the text here" (required)  
  
  "file_name" : "Insert the file name to be saved as here" (optional)  
  
}
```

Output:

```
{  
  
  "message" : "Text uploaded successfully"  
  
}
```

4. upload_direct_route:

Accepts a list of files as an input and uploads it to the data/ folder

Endpoint:

```
/upload-direct
```

API Call Method: POST

Parameters:

```
files (required): List of files directly uploaded
```

Output:

```
{  
  
  "message" : "Files uploaded successfully"  
  
}
```

5. show_upload_files:

Shows all the files uploaded in the /data folder

Endpoint:

```
/show-upload-files
```

API Call Method: GET

Output:

```
{  
  
  "message" : "Files in the directory data/: ",  
  
  "files" : [list of file names]  
  
}
```

6. **show_webhook_files:**

Shows all the files uploaded in the /data_webhooks folder

Endpoint:

```
/show-webhook-files
```

API Call Method: GET

Output:

```
{  
  
  "message" : "Files in the directory data_webhooks/: ",  
  
  "files" : [list of file names]  
  
}
```

7. **delete_upload_file_route:**

Deletes an uploaded file using the saved file's name in the data folder

Endpoint:

```
/delete-upload-file
```

API Call Method: POST

Output:

Parameters:

```
{  
  
  "file_name" : "Insert the filename to be deleted" (required)  
  
}
```

Output:

```
{  
  
  "message" : "File deleted successfully"  
  
}
```

8. delete_webhook_route:

Deletes an uploaded file using the saved file's name in the data_webhooks folder

Endpoint:

```
/delete-webhook
```

API Call Method: POST

Output:

Parameters:

```
{  
  
  "file_name" : "Insert the filename to be deleted" (required)  
  
}
```

Output:

```
{  
  
  "message" : "Webhook file deleted successfully"  
  
}
```

9. delete_all_upload_files_route:

Deletes all the saved files in the data folder

Endpoint:

```
/delete_all_upload_files
```

API Call Method: GET

Output:

Output:

```
{  
  
  "message" : "File deleted successfully"  
  
}
```

10. delete_all_webhooks_route:

Deletes all the saved files in the data_webhooks folder

Endpoint:

```
/delete_all_webhooks
```

API Call Method: GET

Output:

Output:

```
{  
  
  "message" : "All webhook files deleted successfully"  
  
}
```

Status Code:

200/201 : success

400 : Bad Request

401: Authorisation Failed