

Muhammad Zeeshan Ahmad

Currency Exchange Tracking Application

Introduction:

The Currency Exchange Tracking Application is designed to fetch and store daily currency exchange rates from the European Central Bank (ECB) website. It provides a REST API to access current exchange rates and rate changes compared to the previous day for different currencies. This user guide will help you understand how to use and interact with the application effectively.

Getting Started:

1. Installation:

- Ensure you have Python 3 installed on your system.
- Navigate to the project directory.
- Clone the GitHub repository to your local machine using the following command:
`git clone git@github.com:zmalik789/currency_exchange_rate.git`
- Navigate to the project directory.

2. Dependencies:

- Install the required dependencies. All the dependencies are listed in the **requirements.txt**.

3. Database Setup:

- The application uses an SQLite database to store currency exchange rates. The database file is named '**currency_exchange.db**'.
- The database is automatically created when you run the application.

4. Running the Application:

- To start the currency exchange application, run the **currency_exchange.py**.
- The application will automatically fetch exchange rates from the ECB website and store them in the database
- The background job to fetch rates is scheduled to run at 16:05 and 17:35 Europe/Berlin Timezone daily.

API Endpoints:

- Current Exchange Rates:
 - Endpoint: '**current_rates**'
 - Method: GET
 - Parameters: currency (optional)
 - Description: This endpoint provides the current exchange rates for all tracked currencies. If the currency parameter is provided, it will return the rate for the specific currency.

- Example:
 - To get rates for all currencies: http://{host}:5000/current_rates
 - To get rates for a specific currency (e.g., USD):
http://{host}:5000/current_rates?currency=USD
- Rate Changes:
 - Endpoint: **'/rate_changes'**
 - Method: GET
 - Parameters: currency (optional)
 - Description: This endpoint calculates and provides the rate changes compared to the previous day's rates for all tracked currencies. If the currency parameter is provided, it will return the rate change for the specific currency.
 - Example:
 - To get rate changes for all currencies: http://{host}:5000/rate_changes
 - To get rate changes for a specific currency (e.g., EUR):
http://{host}:5000/rate_changes?currency=EUR

DEMO:

Let's assume we started our app on 14-July-2023.

Currency_exchange_rates table status when we first start our app.

id	currency	previous_rates	latest_rates
1	USD	None	1.1221
2	JPY	None	155.51
3	BGN	None	1.9558
4	CZK	None	23.777
5	DKK	None	7.4484
6	GBP	None	0.8559
7	HUF	None	374.55
8	PLN	None	4.449
9	RON	None	4.9439
10	SEK	None	11.493
11	CHF	None	0.9649
12	ISK	None	146.9
13	NOK	None	11.2105
14	TRY	None	29.3872
15	AUD	None	1.635
16	BRL	None	5.3847
17	CAD	None	1.4728
18	CNY	None	8.0076

19	HKD	None	8.7694
20	IDR	None	16797.87
21	ILS	None	4.0499
22	INR	None	92.1985
23	KRW	None	1423.03
24	MXN	None	18.9497
25	MYR	None	5.0786
26	NZD	None	1.7589
27	PHP	None	61.135
28	SGD	None	1.482
29	THB	None	38.864
30	ZAR	None	20.2032

Recorded_dates table stats

id	type	date
1	previous_date	None
2	latest_date	2023-07-14

In this scenario, If we request this endpoint ('https://{host}:5000/current_rates') without specifying any currency it will return this JSON which contains all currencies rates.

```
{'AUD': 1.635, 'BGN': 1.9558, 'BRL': 5.3847, 'CAD': 1.4728, 'CHF': 0.9649, 'CNY': 8.0076, 'CZK': 23.777, 'DKK': 7.4484, 'GBP': 0.8559, 'HKD': 8.7694, 'HUF': 374.55, 'IDR': 16797.87, 'ILS': 4.0499, 'INR': 92.1985, 'ISK': 146.9, 'JPY': 155.51, 'KRW': 1423.03, 'MXN': 18.9497, 'MYR': 5.0786, 'NOK': 11.2105, 'NZD': 1.7589, 'PHP': 61.135, 'PLN': 4.449, 'RON': 4.9439, 'SEK': 11.493, 'SGD': 1.482, 'THB': 38.864, 'TRY': 29.3872, 'USD': 1.1221, 'ZAR': 20.2032}
```

If we request this endpoint ('https://{host}:5000/current_rates') by specifying the currency in parameter params = {'currency': 'USD'} it will return this JSON which contains the specified currency rate.

```
{'USD': 1.1221}
```

If we request this endpoint ('https://{host}:5000/rate_changes') without specifying any currency it will return this JSON which contains all currencies latest_rates, previous_rates which is None in this case and rate_changes as We don't have enough data to show changes. Since the Json is too large, I can't paste it here. If you specify the currency in parameters like if it is USD it will return something like this.

```
{'USD': {'latest_rate': 1.1221, 'previous_rate': None, 'rate_changes': 'Not have enough data'}}
```

Since, on a daily basis it has to update its records on the next working day (i.e. 17-July-2023), the app will update its records.

Currency_exchange_rates table status after updating the records on 17-July-2023. The latest_rates will be copied to previous_rates and new currency rates will placed in the latest_rates column

id	currency	previous_rates	latest_rates
1	USD	1.1221	1.123
2	JPY	155.51	155.38
3	BGN	1.9558	1.9558
4	CZK	23.777	23.757
5	DKK	7.4484	7.4504
6	GBP	0.8559	0.85873
7	HUF	374.55	373.03
8	PLN	4.449	4.4508
9	RON	4.9439	4.9414
10	SEK	11.493	11.4873
11	CHF	0.9649	0.9652
12	ISK	146.9	146.9
13	NOK	11.2105	11.242
14	TRY	29.3872	29.5517
15	AUD	1.635	1.6487
16	BRL	5.3847	5.396
17	CAD	1.4728	1.4845
18	CNY	8.0076	8.0457
19	HKD	8.7694	8.775
20	IDR	16797.87	16842.79
21	ILS	4.0499	4.0897
22	INR	92.1985	92.1073
23	KRW	1423.03	1421.92
24	MXN	18.9497	18.8477
25	MYR	5.0786	5.099
26	NZD	1.7589	1.7727
27	PHP	61.135	61.125
28	SGD	1.482	1.4839
29	THB	38.864	38.856
30	ZAR	20.2032	20.2582

Recorded_dates table stats

id	type	date
1	previous_date	2023-07-14
2	latest_date	2023-07-17

In this scenario, If we request this endpoint ('https://{host}:5000/current_rates') this will behave the

same it returns a JSON containing all currencies and their latest_rates if no currency is specified in the request but it will return the response for only that specified currency if you mention the currency in the request parameters.

If we request this endpoint ('https://{host}:5000/rate_changes') without specifying any currency it will return this JSON which contains all currencies, latest_rates, previous_rates, and changes w.r.t previous day. JSON is too large. I can't paste it here but I can paste a response for only one currency if you mention the currency in the request parameters.

```
{'USD': {'latest_rate': 1.123, 'previous_rate': 1.1221, 'rate_changes': 0.00089999999999999009}}
```

Important:

- The application fetches the data from the ECB website once per day. If you run the application multiple times on the same day, the rate changes will remain the same until the data is updated the next day.
- The application runs on the default IP '127.0.0.1' if it fails to obtain the system's IP address. You can access the API using this default IP and port 5000.
- The application provides rate changes as the difference between the current rate and the previous day's rate. If there is no previous rate available (e.g., for the first day of running the application), it will display "Not have enough data" for rate changes
- The application uses the Flask web framework to serve the API endpoints. It automatically generates SSL certificates for secure connections.

Conclusion:

The Currency Exchange Tracking Application allows you to access up-to-date exchange rates and rate changes for various currencies. You can use the provided API endpoints to retrieve the information you need for currency exchange and tracking purposes. If you encounter any issues or have any questions, please refer to the application's code or documentation or reach out to the developer for assistance.