

# Building Currency Converter in Python


[Start Assignment](#)

- Due 30 Sep by 23:59
- Points 100
- Submitting a file upload

## ***Building Currency Converter in Python***




### **The Brief:**

An API (Application Programming Interface) is a software program that provides communication channels following the HTTP protocol between 2 applications. It is usually used for allowing a client to request or update information from a server.

You are tasked to develop a Web App in Python that will perform currency conversion using data fetched from an open-source API: <https://www.frankfurter.app/>  (<https://www.frankfurter.app/>).

The goal of your program is to display the current conversion rate between 2 currency codes at a specific date or for the latest date. It will also calculate the inverse conversion rate between these 2 currencies.

To do so, you will need to call 3 different API endpoints from the Frankfurter app:

- Extracting the list of available currency codes (documentation: <https://www.frankfurter.app/docs/#currencies>  (<https://www.frankfurter.app/docs/#currencies>))
- Extracting the latest conversion rate for the specified currency codes (documentation: <https://www.frankfurter.app/docs/#latest>  (<https://www.frankfurter.app/docs/#latest>))
- Extracting the historical conversion rate for the specified currency codes and a given date (documentation: <https://www.frankfurter.app/docs/#historical>  (<https://www.frankfurter.app/docs/#historical>))

### **Description:**

In this individual assignment, you will develop a Web App using Streamlit where users can select 2 currencies and an amount to be converted.

# FX Converter

Enter the amount to be converted:

 - +

From Currency:

 ▼

To Currency:

 ▼

Get Latest Rate

Select a date for historical rates:

Conversion Rate

After selection the app will display the latest conversion rate, the converted amount and the inverse conversion rate.

# FX Converter

Enter the amount to be converted:

- +

From Currency:

AUD

▼

To Currency:

USD

▼

Get Latest Rate

## Latest Conversion Rate

The conversion rate on 2025-08-22 from AUD to USD was 0.64246. So 50.0 in AUD correspond to 32.12 in USD. The inverse rate was 1.5565.

## Rate Trend Over the Last 3 years



Additionally users can select a date in the past in order to get the conversion for this day.

# FX Converter

Enter the amount to be converted:

50.00

- +

From Currency:

AUD

▼

To Currency:

USD

▼

Get Latest Rate

Select a date for historical rates:

2024/09/01

Conversion Rate

## Conversion Rate

The conversion rate on 2024-09-01 from AUD to USD was 0.68014. So 50.0 in AUD correspond to 34.01 in USD. The inverse rate was 1.4703.

The displayed text in both cases should follow this convention:

The conversion rate on <date> from <from currency> to <to currency> was <rate> So <from amount> in <from currency> correspond to <to amount> in <to currency> The inverse rate was <inverse rate>.

For example: The conversion rate on 2023-07-10 from AUD to BGN was 118.62. So 100.0 in AUD correspond to 11862.0 in BGN The inverse rate was 0.0084

Your program will be composed of multiple files:

- **app.py**: main Streamlit python script used for managing users' inputs and displaying results
- **api.py**: python script that will contain the code for making API calls
- **frankfurter.py**: python script that will contain the functions used for calling relevant Frankfurter endpoints and extracting information.
- **currency.py**: python script that will contain the function used for formatting the results to be displayed in the Streamlit app.
- **README.md**: a markdown file containing your details (full name, student id), a description of this project, listing of all Python functions and instructions for running your web app

Each of these files have been pre-populated. You will need to fill the defined functions with your code. You are allowed to add more custom Python elements if you wish but they need to be compatible with

the original defined functions.

The Streamlit Web App should have the following element:

- A number input where user can enter the amount to be converted
- A select box listing all the currencies available on Frankfurter
- A second select box listing all the currencies available on Frankfurter
- A button that will fetch the latest conversion rate for the selected currencies
- A text box that will display the expected text described previously
- A date input where user can select a date in the past
- A text box that will display the expected text described previously

## Submission:

You will submit a zip file containing your python scripts and documentation. The name of the zip file **SHOULD** follow this convention: `dsp_at2_<student_id>.zip`

You can find the structure template here: [link](#) 

[https://drive.google.com/file/d/1aNMBnnFR85l77QbrF\\_6QPD4WwXNqxoD2/view?usp=drive\\_link](https://drive.google.com/file/d/1aNMBnnFR85l77QbrF_6QPD4WwXNqxoD2/view?usp=drive_link)

The zip file needs to contain the following files (there shouldn't be any folder):

- app.py
- api.py
- frankfurter.py
- currency.py
- README.md

All assignments need to be submitted before the due date on Canvas. Penalties will be applied for late submission.

## Assessment Criteria:

- Readability and consistency of coding style
- Comprehensibility and quality of code
- Level of clarity for documentation of pseudo code and code
- Reliability and robustness of program

| Some rubric  |              |             |               |             |                |        |
|--|--------------|-------------|---------------|-------------|----------------|--------|
| Criteria   | Ratings      |             |               |             |                | Pts    |
| Readability and consistency of coding style                | 25 Pts<br>HD | 21 Pts<br>D | 18.5 Pts<br>C | 16 Pts<br>P | 12.25 Pts<br>F | 25 pts |
| Comprehensibility and quality of code                      | 25 Pts<br>HD | 21 Pts<br>D | 18.5 Pts<br>C | 16 Pts<br>P | 12.25 Pts<br>F | 25 pts |
| Level of clarity for documentation of pseudo code and code | 25 Pts<br>HD | 21 Pts<br>D | 18.5 Pts<br>C | 16 Pts<br>P | 12.25 Pts<br>F | 25 pts |
| Reliability and robustness of program                      | 25 Pts<br>HD | 21 Pts<br>D | 18.5 Pts<br>C | 16 Pts<br>P | 12.25 Pts<br>F | 25 pts |
| Total points: 100  |              |             |               |             |                |        |