Currency Converter

User Manual

University of St.Gallen

Programming – Introduction Level

Spring Semester 2022

Dr. Mario Silic

By

Nick Albin Elliot Hellström

(21-624-176)

Sebastian Sjursen Hoaas
(21-624-184)

Joakim Brunstrøm
(21-623-947)

Douglas Falk
(21-624-093)

Table of Content

Description

About the Program

The goal of this project was to create a program that displays the exchange rate between currencies, performs currency conversions, enables the user to check historical data and if currencies have appreciated or depreciated over a given period, and that gives the user an opportunity to search for currency codes. To summarize, the program is intended to be an all-in-one solution for currency conversions.

Functions

- Check currency code for any input country
- Display the exchange rate between two input currencies
- Convert a given amount of a currency to another currency
- Check historical exchange rate data for any given input date

Programming Language and Version

The program is written for Python 3.10.

Data

Exchangerate API

The data is accessed through an Exchange rate Application Programming Interface (API). This API uses over 15 exchange rate data sources to deliver exchange rates for 168 currencies.[1] Furthermore, the Exchangerates API also gives access to historical data, not only real-time and intraday data. The API is provided by APILayer and can be found here: https://apilayer.com/marketplace/description/currency_data-api

Currency Code

The currency code data is stored in the excel file: CURRENCY_CODES.xlsx

---

[1] https://apilayer.com/marketplace/description/exchangerates_data-api#details-tab

Installation

The Currency Converter is developed to run on Windows OS. To use Currency Converter the following environmental setup steps must be made.

Software Setup

To run the Currency Converter, Python 3.10 must be installed on your preferred computer. You can download Python using the following steps:

1. Use this link
   https://www.python.org/
2. Then click on "Download for Windows"
3. Install the program

Currency Converter Setup

Currency Converter is downloadable trough the repository in GitHub. To download the program, follow these steps:

1. Follow this link
   https://github.com/SHOAAS/Currency-Converter
2. Download the zipped file containing all necessary files to run the program by clicking the green "Code" button.
3. Save the folder at your desired location and unzip it. Make sure to store all files in the same folder, otherwise the program will not be able to access the necessary data.)
4. Open the program in your preferred terminal.
5. Make sure that your preferred terminal has the following Python libraries installed:
   - Pandas
   - Requests
   - Os

   -

Code With Comments

```python
#this program utilizes the following python libraries
#pandas to read and interpret excel files
#requests to retrieve data from an url
#os to configure custom pathing
import pandas as pd
import requests as rq
import os

#the program imports the latest currency rates (USD)
#stores the rates under the name 'newrates' as a json file
#the json format allows us to easily use the data
api = "https://api.exchangerate-api.com/v4/latest/USD"
response = rq.get(api)
newrates = response.json()

#using pandas to read the excel file
#using os to secure proper pathfinding independent of location
ccodes = pd.read_excel ((os.path.dirname(__file__)+"\CURRENCY_CODES.xlsx"),index_col=0)


#the program uses functions to limit the amount of lines of codes

#this function gives the exchange rate between to currencies using USD as the base
def exchangerate(x,y):
    #adds the upper() so the input does not need to be uppercase
    #this is used throughout the program
    currencyfrom = (newrates["rates"][x.upper()])
    currencyto   = (newrates["rates"][y.upper()])
    exchangerate.finalrate = (1/currencyfrom)*currencyto

#uses the previous function to convert a selected amount
def convert(x,y,z):
    exchangerate(x,z)
    convert.converted= y*exchangerate.finalrate

#returns the currency code of input country
def check():
    #uses 'try except' to counter any invalid inputs
    try:
        cur=input("What country's currency code do you want to know?")
        #pandas converted the data into an array so it can be simply looked up
        ccode= (ccodes.loc[cur.upper(),'Code'])
        print("The curreny code of the currency used in %s is %s"%(cur, ccode))
    except KeyError:
        print("You input does not match any country in our database.")
```

```python
#the 'while True' loop is used throughout the program
#to loop back if input is invalid
#or to loop back to start the program over again
while True:
    #Currency Converter's introduction interface
    #presents the available functions
    print("Welcome to this currency information program!")
    print("These are the currently supported functions:")
    print(" [1] Find Currency Code")
    print(" [2] Find Exchange Rate")
    print(" [3] Convert Currency")
    print(" [4] Compare Exchange Rate Against Historical Rates")

    while True:
        selection=int(input("Please enter the number of the function you want to use:"))
        if selection == 1:
            print("You have selected: Find Currency Code")
            check()
            break

        if selection == 2:
            print("You have selected: Find Exchange Rate")
            while True:
                cur1=input("What is the currency you want to find the exchange rate for?")
                cur2=input("Against what currency?")
                #since the function booth looks up values from an array
                #and uses a mathematical equation
                #it is possible to get two different errors
                #hence try except looks for both key and attribute errors
                try:
                    exchangerate(cur1,cur2)
                    print("The current exchange rate from %s and %s is"%(cur1, cur2),
                     exchangerate.finalrate)
                    break
                except (KeyError, AttributeError) as error:
                    print("Your input is not a valid currency code, please try again.")
                    #allows the usage of check() since any mistake would root from invalid currency
                     code
                    ifcheck=input("You can check for currency codes by typing [?]")
                    if ifcheck == "?":
                        check()
            break

        elif selection == 3:
            print("You have selected: Convert Currency")
            while True:
                cur1=input("Which currency would you like to convert?")
                while True:
                    #checks if the amount is compatible with the function before running it
```

```python
            #to distinguish errors from currency codes and amounts
            try:
                amt=float(input("What amount?"))
                break
            except ValueError:
                print("please input a valid amount")
        cur2=input("To what currency?")
        try:
            convert(cur1,amt,cur2)
            print("For %d %s you get %d %s"%(amt, cur1, convert.converted, cur2))
            break
        except (KeyError, AttributeError) as error:
            print("Your input is not a valid currency code, please try again.")
            ifcheck=input("You can check for currency codes by typing [?]")
            if ifcheck == "?":
                check()
    break


elif selection == 4:
    print("You have selected: Compare Exchange Rates Against Historical Rates")
    cur1=input("What currency do you want to check historical rates for?")
    cur2=input("What currency do you want to compare it to?")
    date=input("what date do you want to compare the current exchange rate to? [YYYY-MM-DD]")
    print("This process may take some time")

    #this section retrieves historical exchange rate data
    #has to be placed after the 'date' input
    oldurl = "https://api.apilayer.com/exchangerates_data/"+date+"&base=USD"
    payload = {}
    headers= {"apikey": "npyPeVq9GUNvyU5gkGjBE0dK6VazKsvE"}
    oldresponse = rq.request("GET", oldurl, headers=headers, data = payload)
    oldrates=oldresponse.json()

    try:
        #finds the new exchange rates and the old ones
        #then calculates the change in percent
        exchangerate(cur1,cur2)
        oldcurrencyfrom = (oldrates["rates"][cur1.upper()])
        oldcurrencyto   = (oldrates["rates"][cur2.upper()])
        oldrate = (1/oldcurrencyfrom)*oldcurrencyto
        change = float((exchangerate.finalrate-oldrate)/oldrate)
        change = change*100
        #checks if the change is positive or negative
        #in order to respond correctly
        if change <= 0:
            change=change*-1
            print("%s has depriciated %.4f percent to %s since %s"%
                (cur1.upper(),change,cur2.upper(),date))
        elif change >= 0:
```

```python
                    print("The %s has appreciated %.4f percent to %s since
%s"%(cur1,change,cur2,date))
                elif change == 0:
                    print("The rate is more or less the same")

            except (KeyError, AttributeError) as error:
                print("Your input is invalid. Make sure your currency codes are correct, and the date
                 has the format [YYYY-MM-DD].")
                ifcheck=input("You can check for currency codes by typing [?]")
                if ifcheck == "?":
                    check()
            break
        else:
            print("please enter a valid input")
    #allows the user to go back to the start of the program
    while True:
        quit=input("Would you like to return to the main menu? (Yes/No)")
        if quit.upper() in ("YES", "NO"):
            break
    if quit.upper() == "NO":
        break
print("Thank you for using our program!")
```

Sample Output

*(Words in **bold** represent user input.)*

Welcome to this currency information program!

These are the currently supported functions:

[1] Find Currency Code

[2] Find Exchange Rate

[3] Convert Currency

[4] Compare Exchange Rate Against Historical Rates

Please enter the number of the function you want to use. **1**

You have selected: Find Currency Code

Which country's currency code do you want to know? **Norway**

The currency code of the currency used in Norway is NOK

Would you like to return to the main menu? (Yes/No) **Yes**

Welcome to this currency information program!

These are the currently supported functions:

[1] Find Currency Code

[2] Find Exchange Rate

[3] Convert Currency

[4] Compare Exchange Rate against Historical Rates

Please enter the number of the function you want to use. **2**

You have selected: Find Exchange Rate

What is the currency you want to find the exchange rate for?
NOK

Against what currency? **CHF**

The current exchange rate between NOK and CHF is 0.10

Would you like to return to the main menu? (Yes/No) **Yes**

Welcome to this currency information program!

These are the currently supported functions:

[1] Find Currency Code

[2] Find Exchange Rate

[3] Convert Currency

[4] Compare Exchange Rate against Historical Rates

Please enter the number of the function you want to use. **4**

You have selected: Compare Exchange Rates Against Historical Rates

What currency do you want to look at? **NOK**

What currency do you want to compare it to? **CHF**

What date do you want to compare the current exchange rate to? [YYYY-MM-DD] **2015-08-23**

This process may take some time…

NOK has depreciated 11.5345 percent to CHF since 2015-08-23

Would you like to return to the main menu? (Yes/No) **No**

Thank you for using our program!