**CIS 2286. Lab Assignment 2.**

**Conditions, Loops**

**Strings**

**Testing**

**Error handling**

Goal: Load currency quotes from external service. Add input validation methods and error handling. Add unit tests.

***Detailed requirements:***

1. Allow Currency Exchange service (CE) run in a loop to collect user input.
2. Add an “action” parameter. Display a menu to allow selection of the following actions. Depending on the action collect the following input fields

* save the new conversion rate

**action=’S’**

**currency\_from**

**currency\_to**

**conversion\_rate**

* display available quotes

**action=’D’**

* exchange an amount in one currency to an amount in the other currency

**action=’X’**

**currency\_from**

**currency\_to**

**amount**

* get currency quote for the two currencies

**action=’Q’**

**currency\_from**

**currency\_to**

* exit application

**action=’E’**

1. Currency quote is loaded from the external service <https://free.currencyconverterapi.com/>

“JSON support” link on this page is the REST API endpoint we are using. Please do not abuse this free service . Your goal is to extract value of **conversion\_rate** from the JSON string returned by the service

1. Implement validation of user input: amount, currencies, action. Normalize action. Details of the new functions are in **app.py** specs
2. Implement storage and display of quotes through a global list **quote**.**quote\_list**
3. \*Extra: implement additional logging of timing of the call and input parameters by adding decorator common.log to methods
4. Add 5 more success and fail unit tests for the validation functions. Unit tests are in

**tests/currency\_exchange\_tests**

to run all tests:

**python -m unittest tests.currency\_exchange\_tests**

to run group of tests:

**python -m unittest tests.currency\_exchange\_tests.TestConversionMethods**

1. While connecting to remote service always wrap the service call in **try–except** clauses
2. Run your application with the use cases below logging to a log file:

**python3 app.py 2>log**

**python3 app.py 2>>log appends to an existing file**

***Test use cases:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **action** | **currency\_from** | **currency\_to** | **amount** | **conversion\_rate** |
| 1 | Q | USD | EUR | ---- | ---- |
| 2 | S | USD | GBP | ---- | 0.74 |
| 3 | D | ---- | ---- | ---- | ---- |
| 4 | X | GBP | AUD | 12.12 | ---- |
| 5 | Q | AUD | EUR | ---- | ---- |
| 6 | X | AUD | GPB | 109.09 | ---- |
| 7 | S | EUR | AUD | ---- | 1.59 |
| 8 | D | ---- | ---- | ---- | ---- |
| 9 | S | JPY | AUD | ---- | 0.012 |
| 10 | Q | JPY |  | ---- | ---- |
| 11 | X | USD | AUD | 237 | ---- |
| 12 | D | ---- | ---- | ---- | ---- |
| 13 | E | ---- | ---- | ---- | ---- |

***My script produces the following output:***

LAPIN:currency-exchage-service-lab1-solution sab$ python3 app.py 2>log

Enter the amount to convert > 10.10

Enter the currency code to convert from, a 3-character string > USD

Enter the currency code to convert to, a 3-character string > EUR

{ "success" : true, "error" : "", "source" : "{"amount" : "10.10 ", "currency" : "USD"}", "target" : "{"amount" : "12.44 ", "currency" : "EUR"}" }

LAPIN:currency-exchage-service-lab1-solution sab$ python3 app.py 2>>log

Enter the amount to convert > 10.10

Enter the currency code to convert from, a 3-character string > USD

Enter the currency code to convert to, a 3-character string > GBP

{ "success" : true, "error" : "", "source" : "{"amount" : "10.10 ", "currency" : "USD"}", "target" : "{"amount" : "12.09 ", "currency" : "GBP"}" }

LAPIN:currency-exchage-service-lab1-solution sab$ python3 app.py 2>>log

Enter the amount to convert > 12.12

Enter the currency code to convert from, a 3-character string > GBP

Enter the currency code to convert to, a 3-character string > AUD

{ "success" : true, "error" : "", "source" : "{"amount" : "12.12 ", "currency" : "GBP"}", "target" : "{"amount" : "13.24 ", "currency" : "AUD"}" }

LAPIN:currency-exchage-service-lab1-solution sab$ python3 app.py 2>>log

Enter the amount to convert > 12.12

Enter the currency code to convert from, a 3-character string > EUR

Enter the currency code to convert to, a 3-character string > JPY

{ "success" : true, "error" : "", "source" : "{"amount" : "12.12 ", "currency" : "EUR"}", "target" : "{"amount" : "11.30 ", "currency" : "JPY"}" }

LAPIN:currency-exchage-service-lab1-solution sab$ python3 app.py 2>>log

Enter the amount to convert > 14.14

Enter the currency code to convert from, a 3-character string > EUR

Enter the currency code to convert to, a 3-character string > GBP

{ "success" : true, "error" : "", "source" : "{"amount" : "14.14 ", "currency" : "EUR"}", "target" : "{"amount" : "13.74 ", "currency" : "GBP"}" }

LAPIN:currency-exchage-service-lab1-solution sab$ python3 app.py 2>>log

Enter the amount to convert > 14.14

Enter the currency code to convert from, a 3-character string > JPY

Enter the currency code to convert to, a 3-character string > AUD

{ "success" : true, "error" : "", "source" : "{"amount" : "14.14 ", "currency" : "JPY"}", "target" : "{"amount" : "16.10 ", "currency" : "AUD"}" }

LAPIN:currency-exchage-service-lab1-solution sab$

***The content of the log file:***

LAPIN:currency-exchage-service-lab1-solution sab$ cat log

2018-04-26 16:24:24.416186 currency-exchange 58397 37b1763f-1542-430a-a6e6-dc0a061e9271 LAPIN.local started

2018-04-26 16:24:42.769042 currency-exchange 58397 37b1763f-1542-430a-a6e6-dc0a061e9271 LAPIN.local USD EUR 10.10

2018-04-26 16:24:42.770572 currency-exchange 58397 37b1763f-1542-430a-a6e6-dc0a061e9271 LAPIN.local {"USD\_EUR:{"val":1.2318840579710144}}

2018-04-26 16:24:42.770663 currency-exchange 58397 37b1763f-1542-430a-a6e6-dc0a061e9271 LAPIN.local 1.2318840579710144 12.44

2018-04-26 16:24:42.770706 currency-exchange 58397 37b1763f-1542-430a-a6e6-dc0a061e9271 LAPIN.local { "success" : true, "error" : "", "source" : "{"amount" : "10.10 ", "currency" : "USD"}", "target" : "{"amount" : "12.44 ", "currency" : "EUR"}" }

2018-04-26 16:24:42.770782 currency-exchange 58397 37b1763f-1542-430a-a6e6-dc0a061e9271 LAPIN.local ended

2018-04-26 16:24:49.839359 currency-exchange 58403 ed24a524-63b0-403a-9c82-c4c51da8bebe LAPIN.local started

2018-04-26 16:25:02.088816 currency-exchange 58403 ed24a524-63b0-403a-9c82-c4c51da8bebe LAPIN.local USD GBP 10.10

2018-04-26 16:25:02.088919 currency-exchange 58403 ed24a524-63b0-403a-9c82-c4c51da8bebe LAPIN.local {"USD\_GBP:{"val":1.1971830985915493}}

2018-04-26 16:25:02.088967 currency-exchange 58403 ed24a524-63b0-403a-9c82-c4c51da8bebe LAPIN.local 1.1971830985915493 12.09

2018-04-26 16:25:02.088995 currency-exchange 58403 ed24a524-63b0-403a-9c82-c4c51da8bebe LAPIN.local { "success" : true, "error" : "", "source" : "{"amount" : "10.10 ", "currency" : "USD"}", "target" : "{"amount" : "12.09 ", "currency" : "GBP"}" }

2018-04-26 16:25:02.089050 currency-exchange 58403 ed24a524-63b0-403a-9c82-c4c51da8bebe LAPIN.local ended

2018-04-26 16:25:05.805738 currency-exchange 58406 f893c638-3cd1-4e4b-8d17-f480cdca4a5d LAPIN.local started

2018-04-26 16:25:19.355804 currency-exchange 58406 f893c638-3cd1-4e4b-8d17-f480cdca4a5d LAPIN.local GBP AUD 12.12

2018-04-26 16:25:19.355898 currency-exchange 58406 f893c638-3cd1-4e4b-8d17-f480cdca4a5d LAPIN.local {"GBP\_AUD:{"val":1.0923076923076922}}

2018-04-26 16:25:19.355942 currency-exchange 58406 f893c638-3cd1-4e4b-8d17-f480cdca4a5d LAPIN.local 1.0923076923076922 13.24

2018-04-26 16:25:19.355969 currency-exchange 58406 f893c638-3cd1-4e4b-8d17-f480cdca4a5d LAPIN.local { "success" : true, "error" : "", "source" : "{"amount" : "12.12 ", "currency" : "GBP"}", "target" : "{"amount" : "13.24 ", "currency" : "AUD"}" }

2018-04-26 16:25:19.356026 currency-exchange 58406 f893c638-3cd1-4e4b-8d17-f480cdca4a5d LAPIN.local ended

2018-04-26 16:25:22.303676 currency-exchange 58410 5606c2f6-3a0c-4b08-8cda-82c8a7053d54 LAPIN.local started

2018-04-26 16:25:35.971553 currency-exchange 58410 5606c2f6-3a0c-4b08-8cda-82c8a7053d54 LAPIN.local EUR JPY 12.12

2018-04-26 16:25:35.971650 currency-exchange 58410 5606c2f6-3a0c-4b08-8cda-82c8a7053d54 LAPIN.local {"EUR\_JPY:{"val":0.9324324324324325}}

2018-04-26 16:25:35.971694 currency-exchange 58410 5606c2f6-3a0c-4b08-8cda-82c8a7053d54 LAPIN.local 0.9324324324324325 11.30

2018-04-26 16:25:35.971720 currency-exchange 58410 5606c2f6-3a0c-4b08-8cda-82c8a7053d54 LAPIN.local { "success" : true, "error" : "", "source" : "{"amount" : "12.12 ", "currency" : "EUR"}", "target" : "{"amount" : "11.30 ", "currency" : "JPY"}" }

2018-04-26 16:25:35.971777 currency-exchange 58410 5606c2f6-3a0c-4b08-8cda-82c8a7053d54 LAPIN.local ended

2018-04-26 16:25:38.147635 currency-exchange 58414 821f69b8-960d-4cc8-8750-e561734371c5 LAPIN.local started

2018-04-26 16:25:51.579589 currency-exchange 58414 821f69b8-960d-4cc8-8750-e561734371c5 LAPIN.local EUR GBP 14.14

2018-04-26 16:25:51.579682 currency-exchange 58414 821f69b8-960d-4cc8-8750-e561734371c5 LAPIN.local {"EUR\_GBP:{"val":0.971830985915493}}

2018-04-26 16:25:51.579737 currency-exchange 58414 821f69b8-960d-4cc8-8750-e561734371c5 LAPIN.local 0.971830985915493 13.74

2018-04-26 16:25:51.579764 currency-exchange 58414 821f69b8-960d-4cc8-8750-e561734371c5 LAPIN.local { "success" : true, "error" : "", "source" : "{"amount" : "14.14 ", "currency" : "EUR"}", "target" : "{"amount" : "13.74 ", "currency" : "GBP"}" }

2018-04-26 16:25:51.579821 currency-exchange 58414 821f69b8-960d-4cc8-8750-e561734371c5 LAPIN.local ended

2018-04-26 16:25:53.599849 currency-exchange 58417 56fea091-5180-484a-8e0a-1ac4f97fe6c9 LAPIN.local started

2018-04-26 16:26:05.955286 currency-exchange 58417 56fea091-5180-484a-8e0a-1ac4f97fe6c9 LAPIN.local JPY AUD 14.14

2018-04-26 16:26:05.955378 currency-exchange 58417 56fea091-5180-484a-8e0a-1ac4f97fe6c9 LAPIN.local {"JPY\_AUD:{"val":1.1384615384615384}}

2018-04-26 16:26:05.955436 currency-exchange 58417 56fea091-5180-484a-8e0a-1ac4f97fe6c9 LAPIN.local 1.1384615384615384 16.10

2018-04-26 16:26:05.955466 currency-exchange 58417 56fea091-5180-484a-8e0a-1ac4f97fe6c9 LAPIN.local { "success" : true, "error" : "", "source" : "{"amount" : "14.14 ", "currency" : "JPY"}", "target" : "{"amount" : "16.10 ", "currency" : "AUD"}" }

2018-04-26 16:26:05.955520 currency-exchange 58417 56fea091-5180-484a-8e0a-1ac4f97fe6c9 LAPIN.local ended

LAPIN:currency-exchage-service-lab1-solution sab$

***Code structure for lab2***

The project code contains several files with functions prototypes for Lab1 solution. The goal is to implement these methods.

**currency-exchange**

**├── \_\_init\_\_.py**

**├── app.py**

**├── common**

**│   ├── \_\_init\_\_.py**

**│   └── utils.py**

**├── log**

**├── resources**

**│   ├── \_\_init\_\_.py**

**│   ├── quote.py**

**│   └── transaction.py**

**└── tests**

**└── currency\_exchange\_tests.py**

The structure of this tree is designed for the future move from a command line script to a Flask application. We will start outlining our project as a RESTful application from the beginning.

**currency-exchange/** is the root directory for this project and the name of the service (application)

**app.py** is the main entry point of the application. The logic is outlined in app.main\_command\_line()

**common**/ consists of common classes and functions that are useful for all the modules. Some examples are global constants, global helper functions or logging utilities. Currently we have util.py module with log function

**resources/** - collection of modules representing resources. Resource is the fundamental concept in any RESTful application. A resource is a unit of a REST API that you can call. It has some type, associated data, methods operating on it and relationships with other resources. In our project we will be defining two resources:

- **quote**: an exchange rate between two currencies

- **transaction**: a transfer of a given amount from one currency to another

**\_\_init\_\_.py** - files are required to make Python treat the directories as containing packages; this is done to prevent directories with a common name, such as string, from unintentionally hiding valid modules that occur later (deeper) on the module search path

**tests/currency\_exchange\_tests.py -** unit tests