JAVA CURRENCY CONVERTER

This currency converter program was built using the java.net library and an an external library JSON.ORG.

The app supports around 170 countries and it processes the user requests using ***live data.***

The data is fetched via an API for which we have acquired a personal key.
API LINK
https://apilayer.com/marketplace/exchangerates_data-api#documentation-tab

First we import all ther libraries required to make an API call.
All of them are part of the JAVA Standard library.

```
import java.net.http.HttpRequest;
import java.net.http.HttpClient;
import java.net.http.HttpResponse;
import java.net.URI;
import java.util.Scanner;
import org.json.simple.JSONObject;
import org.json.simple.parser.*;
```

Making a user-defined function called **APi_call();**

```
public static void APi_call() throws Exception {
        Scanner input = new Scanner(System.in);
        System.out.println("what do you want to convert to?");

        String ConTo = input.nextLine();
        System.out.println("what do you want to convert from?");
        String ConFrom = input.nextLine();
        System.out.println("enter the amount");

        int amount =  input.nextInt();
        var url =
"https://api.apilayer.com/exchangerates_data/convert?to=" + ConTo +
"&from=" + ConFrom + "&amount="
                + amount;
        var request = HttpRequest.newBuilder().GET().uri(URI.create(url))
                .header("apikey",
```

```
"EHMuqKYmp2QGX4JB5Fn72TmiHP1Pw4p0").build();
```

BREAKDOWN:-

Here we specify an url variable with the url making an api call.

```
  var url =
"https://api.apilayer.com/exchangerates_data/convert?to=" + ConTo +
"&from=" + ConFrom + "&amount="
```

We can insert the user requested countries in the url by accepting them in the form of strings and appending them in the request.

We use the inbuilt request builder of the java.net library ot build a request.

```
var request = HttpRequest.newBuilder().GET().uri(URI.create(url))

.header("apikey", "EHMuqKYmp2QGX4JB5Fn72TmiHP1Pw4p0").build();
```

Here we specify the type of the request “GET” , call the URI creator function to create an uri of the API request and add the header to it which is our personal API key allotted to us.

The api key is important since it allows the web host to track who has been accessing their server and how many request have been sent per day and other details.

```
var client = HttpClient.newBuilder().build();
```

Then we create a client object which receives the response once we send the request to the API.

```
var response = client.send(request,
HttpResponse.BodyHandlers.ofString());
```

Then we create an object of response,
We call the send() method of the client class to get the response in the form of a string.

```
 Object obj = new JSONParser().parse(response.body())
```

We create an object of the Object class containing the response body in the JSON format.

```
JSONObject jo= (JSONObject)obj;
```

Then we create a JSONObject of the object 'obj' but we cast it into a JSONOBject type. So that we can use the inbuilt methods of this class to extract required labelled data from the JSON response.

The JSON response looks something like this.
Its a label and data pair format.

```
{
    "success": true,
    "query": {
        "from": "INR",
        "to": "USD",
        "amount": 1234
    },
    "info": {
        "timestamp": 1665316623,
        "rate": 0.012073
    },
    "date": "2022-10-09",
    "result": 14.898082
}
```

We assign the result value from the JSON response by using the get() method of the object 'jo'to extract the data part of the label 'result'

```
  double result= (double) jo.get("result");
        System.out.println(result+" "+ConTo);
```

And, we finally print it out.

```
System.out.println(result+" "+ConTo);
```

COMPLETE CODE:-

```
import java.net.http.HttpRequest;
```

```
import java.net.http.HttpClient;
import java.net.http.HttpResponse;
import java.net.URI;
import java.util.Scanner;
import org.json.simple.JSONObject;
import org.json.simple.parser.*;
public class mainapp {
    public static void main(String[] args) throws Exception {
        Scanner sc = new Scanner(System.in);
        //System.out.println("do you want to make another conversion");
        //System.out.println("press 1 for Yes.");
        //System.out.println("press 0 for No");
        String a="no";
        //var;

//url="https://api.apilayer.com/exchangerates_data/convert?to=USD&from=I
NR&amount=140000";

        String exit="yes";
        while(true){

            if(exit.equalsIgnoreCase(a))
            break;
        APi_call();
        System.out.println("do you want to make another conversion");
        exit=sc.nextLine();


        }
    }

    public static void APi_call() throws Exception {
        Scanner input = new Scanner(System.in);
        System.out.println("what do you want to convert to?");

        String ConTo = input.nextLine();
        System.out.println("what do you want to convert from?");
        String ConFrom = input.nextLine();
        System.out.println("enter the amount");

        int amount =  input.nextInt();
        var url =
"https://api.apilayer.com/exchangerates_data/convert?to=" + ConTo +
"&from=" + ConFrom + "&amount="
                + amount;
        var request =
```

```
HttpRequest.newBuilder().GET().uri(URI.create(url))
                .header("apikey",
"EHMuqKYmp2QGX4JB5Fn72TmiHP1Pw4p0").build();

        var client = HttpClient.newBuilder().build();
        var response = client.send(request,
HttpResponse.BodyHandlers.ofString());
       // System.out.println(response.body());
        Object obj = new JSONParser().parse(response.body());
        JSONObject jo= (JSONObject)obj;
          double result= (double) jo.get("result");
        System.out.println(result+" "+ConTo);

    }
}
```

GIT REPOSITORY

You can find the code here:-
https://github.com/Ashtrobuff/JAVA-CURRENCY-CONVERTER
Make sure you download all the files in the same folder and unzip the jar files and store the org folder in the root directory of the mainapp.java file.

OUTPUT :-

```
what do you want to convert to?
USD
what do you want to convert from?
INR
enter the amount
3783747
45681.177531 USD
do you want to make another conversion
yes
what do you want to convert to?
JPY
what do you want to convert from?
PKR
enter the amount
1234343
811869.358762 JPY
do you want to make another conversion
yes
what do you want to convert to?
KWD
```

```
what do you want to convert from?
AED
enter the amount
123443
10431.180386 KWD
do you want to make another conversion
No
*END*
```

Code supports 170 countries ,all G7 and ASEAN countries.

Annex 2.5

ISO CURRENCY CODES

SL	Currency Name	CURCD
1	Afghanistan Afghani	AFA
2	Albanian Lek	ALL
3	Algerian Dinar	DZD
4	Angolan New Kwanza	AOA
5	Argentine Peso	ARS
6	Armenian Dram	AMD
7	Aruban Florin	AWG
8	Australian Dollar	AUD
9	Azerbaijanian Manat	AZM
10	Bahamanian Dollar	BSD
11	Bahraini Dinar	BHD
12	Bangladeshi Taka	BDT
13	Barbados Dollar	BBD
14	Belarussian Ruble	BYR
15	Belize Dollar	BZD
16	Bermudian Dollar	BMD
17	Bhutan Ngultrum	BTN
18	Bolivian Boliviano	BOB
19	Botswana Pula	BWP
20	Brazilian Real	BRL
21	Brunei Dollar	BND
22	Bulgarian Lev	BGL
23	Burundi Franc	BIF
24	Cambodian Riel	KHR
25	Canadian Dollar	CAD
26	Cape Verde Escudo	CVE
27	Cayman Islands Dollar	KYD

SL	Currency Name	CURCD
28	CFA Franc BCEAO	XOF
29	CFA Franc BEAC	XAF
30	CFP Franc	XPF
31	Chilean Peso	CLP
32	Chinese Yuan Renminbi	CNY
33	Colombian Peso	COP
34	Comoros Franc	KMF
35	Congolese Franc	CDF
36	Costa Rican Colon	CRC
37	Croatian Kuna	HRK
38	Cuban Peso	CUP
39	Cyprus Pound	CYP
40	Czech Koruna	CZK
41	Danish Kroner	DKK
42	Djibouti Franc	DJF
43	Dominican Repub. Peso	DOP
44	East Caribbean Dollar	XCD
45	Egyptian Pound	EGP
46	El Salvador Colon	SVC
47	Eritrea Nakfa	ERN
48	Estonian Kroon	EEK
49	Ethiopian Birr	ETB
50	Euro	EUR
51	Falkland Islands Pound	FKP
52	Fiji Dollar	FJD
53	Gambian Dalasi	GMD
54	Georgian Lari	GEL