Technical University of Moldova

Faculty of Computers, Informatics and Microelectronics

Department of Software Engineering and Automation

Web Programming

Laboratory work #2

HTTP, CACHING AND CONTENT NEGOTIATION

Author:
Daniela Afteni
std. gr. FAF-203

Supervisor:
Alexei Șerșun

Chișinău 2023

1 Task

The task for this lab is:

1. You have to write a command line program, using [go2web](go2web) executable as a starting point;
2. The program should implement at least the following, as in the Listing 1:

```
go2web -u <URL>
# make an HTTP request to the specified URL and print the response

go2web -s <search-term>
# make an HTTP request to search the term using your favorite search engine and print
    top 10 results

go2web -h
# show this help
```

Listing 1: Main tasks

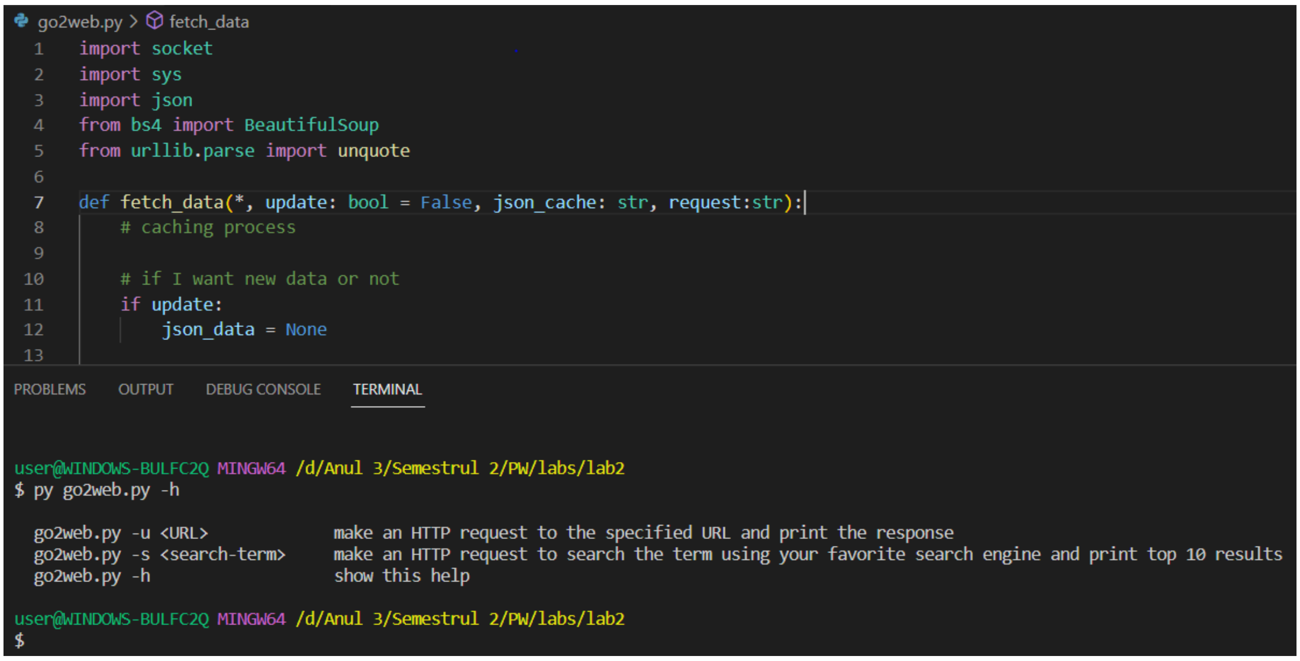
3. The responses from request should be human-readable (e.g. no HTML tags in the output).

2 Results

To implement this laboratory work, I made sure that in python there are right tools as CLI parser, HTML/JSON parser and support for TCP sockets.

In the laboratory work nr. 2 have been realised: executable with ‘-u‘ and ‘-s‘, results/links from search engine can be accessed, implementation of HTTP request redirects and HTTP cache mechanism.

In the Figure 4 is represented the help option that consists just of printing in a right form of the commands that were described in the readme file.



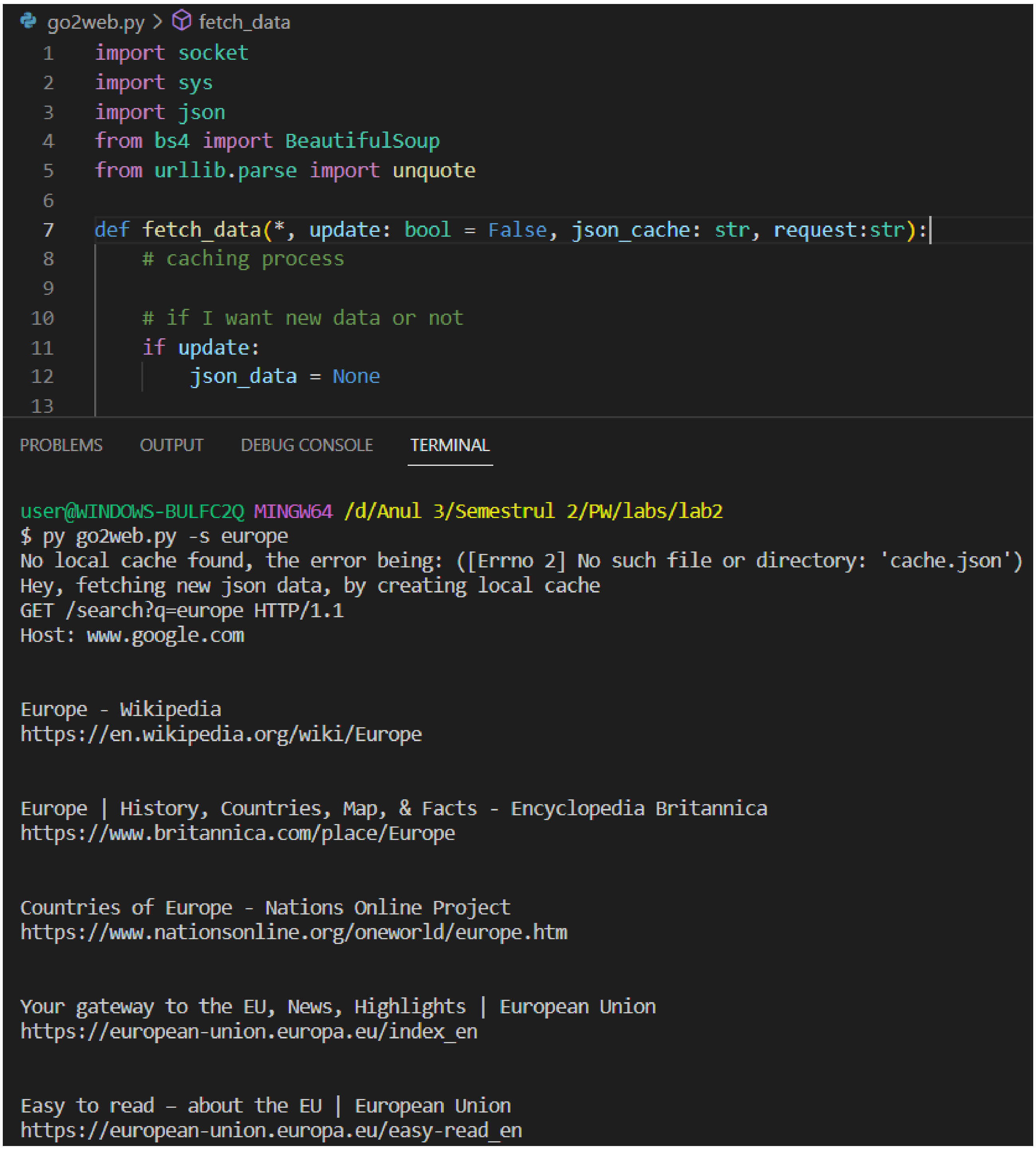
```
import socket
import sys
import json
from bs4 import BeautifulSoup
from urllib.parse import unquote

def fetch_data(*, update: bool = False, json_cache: str, request:str):
    # caching process

    # if I want new data or not
    if update:
        json_data = None

```

```
user@WINDOWS-BULFC2Q MINGW64 /d/Anul 3/Semestrul 2/PW/labs/lab2
$ py go2web.py -h

  go2web.py -u <URL>             make an HTTP request to the specified URL and print the response
  go2web.py -s <search-term>     make an HTTP request to search the term using your favorite search engine and print top 10 results
  go2web.py -h                   show this help

user@WINDOWS-BULFC2Q MINGW64 /d/Anul 3/Semestrul 2/PW/labs/lab2
$
```

Figure 1: ”go2web.py -h”

The next is Figure 2, which determines the searching a term option. There can be seen as well and the HTTP caching mechanism. The number of represented links is around 10, that can be accessed by clicking on it.

This process is defined by calling a function, that is responsible for it. Firstly, that we do is determine the connection by connecting to ”www.google.com” at port 80 (main port), then comes the caching process (which is going to be discussed a bit later). Due to caching we will get response data and then start the web scraping [1], by extracting specific elements from the html. For this implementation there were used BeautifulSoup [2] and unquote.



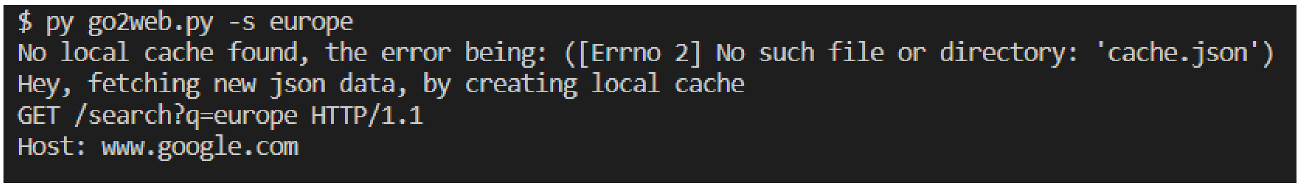
go2web.py > fetch_data

```
import socket
import sys
import json
from bs4 import BeautifulSoup
from urllib.parse import unquote

def fetch_data(*, update: bool = False, json_cache: str, request:str):
    # caching process

    # if I want new data or not
    if update:
        json_data = None

```

```
user@WINDOWS-BULFC2Q MINGW64 /d/Anul 3/Semestrul 2/PW/labs/lab2
$ py go2web.py -s europe
No local cache found, the error being: ([Errno 2] No such file or directory: 'cache.json')
Hey, fetching new json data, by creating local cache
GET /search?q=europe HTTP/1.1
Host: www.google.com


Europe - Wikipedia
https://en.wikipedia.org/wiki/Europe


Europe | History, Countries, Map, & Facts - Encyclopedia Britannica
https://www.britannica.com/place/Europe


Countries of Europe - Nations Online Project
https://www.nationsonline.org/oneworld/europe.htm


Your gateway to the EU, News, Highlights | European Union
https://european-union.europa.eu/index_en


Easy to read - about the EU | European Union
https://european-union.europa.eu/easy-read_en
```

Figure 2: ”go2web.py -s <search-term>”

Regarding the HTTP caching mechanism, there was used another function that initially needs a json file to store the cached. After that, in the fetching function we open the file json and read it, and if we do have json data and it was locally cached, then we are going to use it instead our new request. In case when we send the request for the first time and the cache.json is not created, we are going to get some errors (that shows us that we don’t have local cache) and create this file and store the request. Basically, if there is no json data, then we will write this cache file, by inserting in it, as it is shown in the Figure 3.

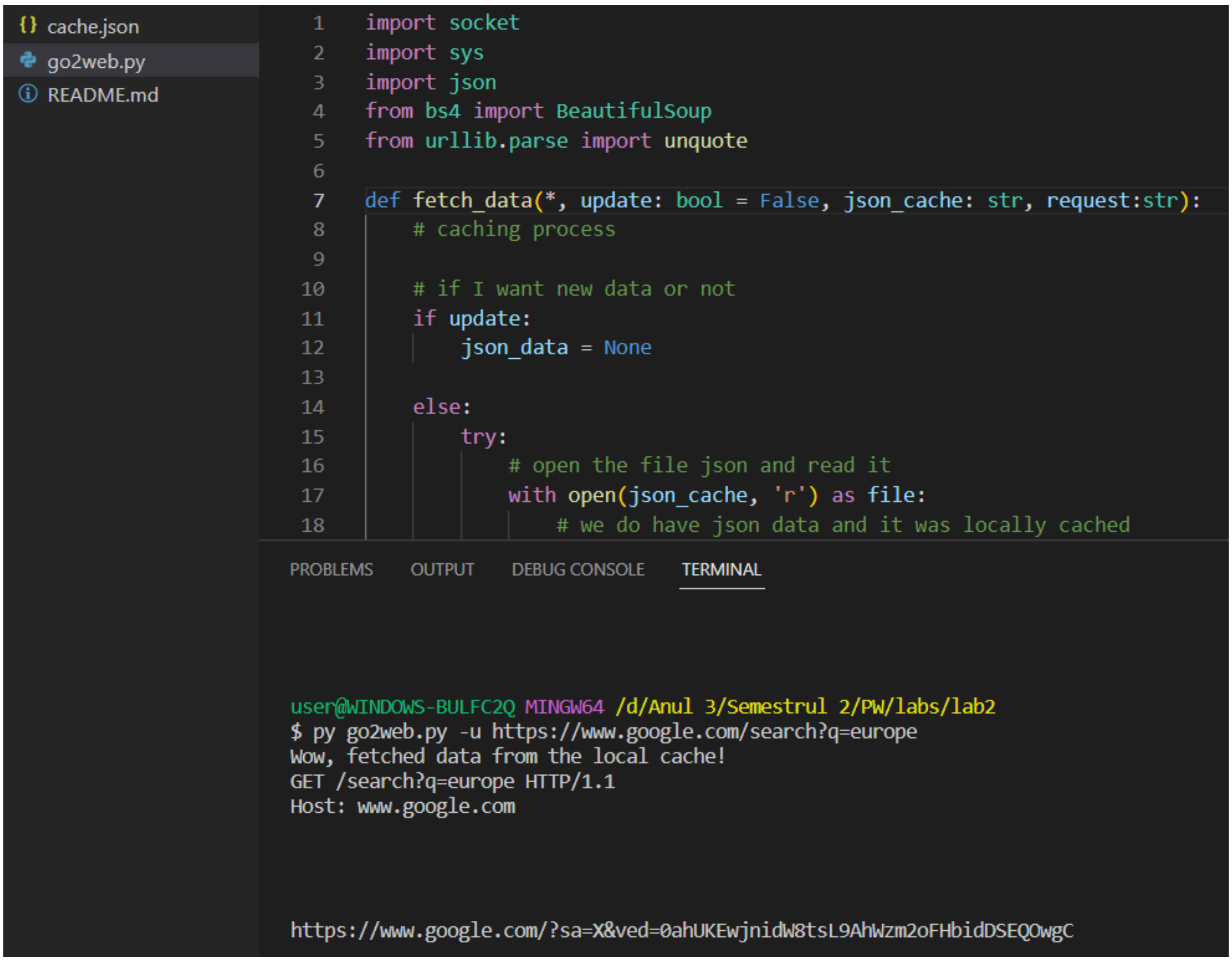


```
$ py go2web.py -s europe
No local cache found, the error being: ([Errno 2] No such file or directory: 'cache.json')
Hey, fetching new json data, by creating local cache
GET /search?q=europe HTTP/1.1
Host: www.google.com
```

Figure 3: Fetching new data by creating local cache

The last option is shown in Figure 4, Figure 5 and in Figure 6, which is related to the searching process by a specific URL. There are represented a lot of links, that can be accessed as well by clicking on it.

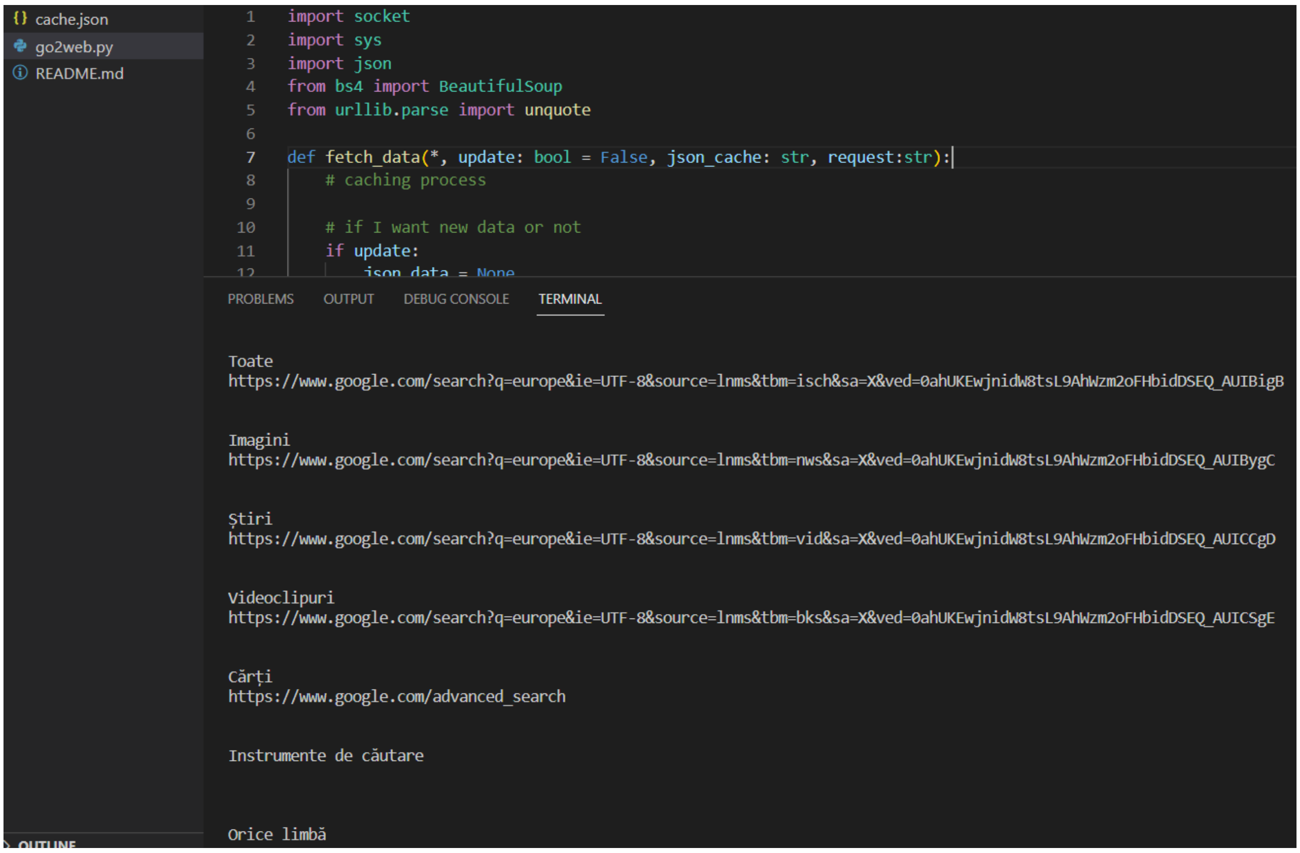
This process is defined by calling a function, that is responsible for it. Firstly, that we do is determine the connection by connecting to the host, "www.google.com" at port 80 (main port) as well as building the request inserting in it the host and path (path is /search?q=europe, and host is www.google.com). Then comes as well as in the searching term process, the caching (which is going to be discussed a bit later). Due to caching we will get response data and then start the web scraping, by extracting specific elements from the html. We get rid of any styles, heads and scripts using BeautifulSoup and work with descendants, so as to get from our child the links that we need.



```python
import socket
import sys
import json
from bs4 import BeautifulSoup
from urllib.parse import unquote

def fetch_data(*, update: bool = False, json_cache: str, request:str):
    # caching process

    # if I want new data or not
    if update:
        json_data = None

    else:
        try:
            # open the file json and read it
            with open(json_cache, 'r') as file:
                # we do have json data and it was locally cached
```

```
user@WINDOWS-BULFC2Q MINGW64 /d/Anul 3/Semestrul 2/PW/labs/lab2
$ py go2web.py -u https://www.google.com/search?q=europe
Wow, fetched data from the local cache!
GET /search?q=europe HTTP/1.1
Host: www.google.com



https://www.google.com/?sa=X&ved=0ahUKEwjnidW8tsL9AhWzm2oFHbidDSEQOwgC
```

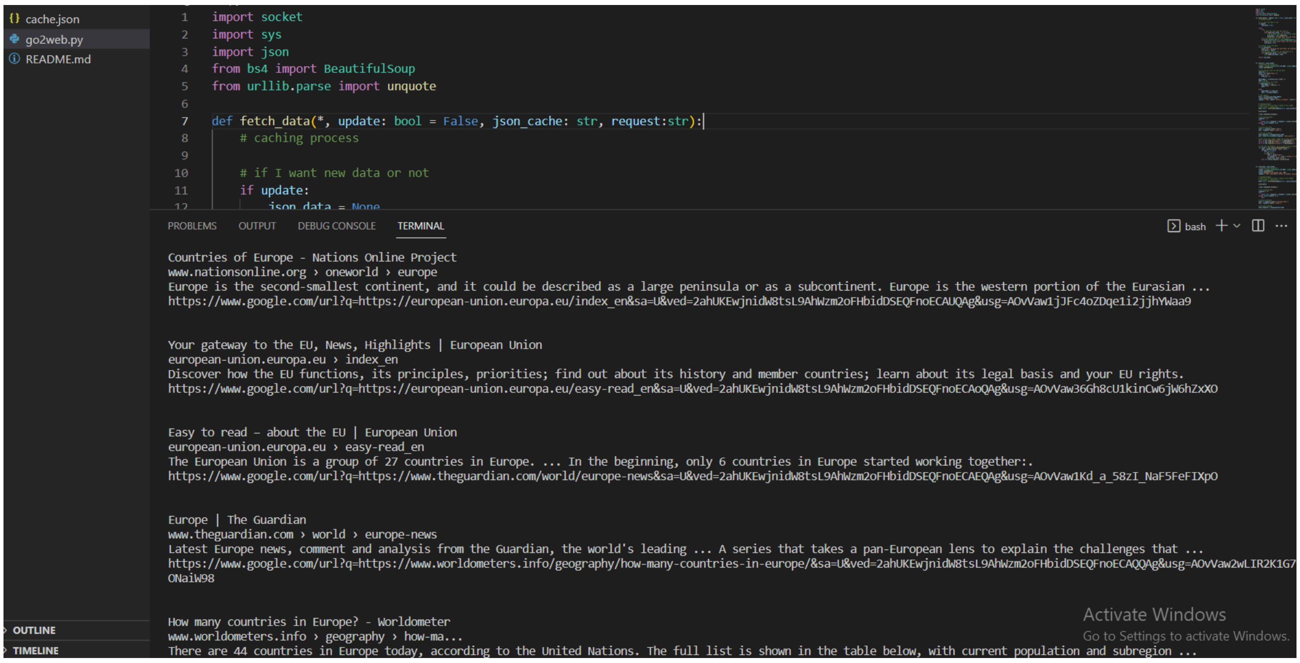
Figure 4: "go2web.py -u <URL>" part 1



```python
import socket
import sys
import json
from bs4 import BeautifulSoup
from urllib.parse import unquote

def fetch_data(*, update: bool = False, json_cache: str, request:str):
    # caching process

    # if I want new data or not
    if update:
        json_data = None
```

```
Toate
https://www.google.com/search?q=europe&ie=UTF-8&source=lnms&tbm=isch&sa=X&ved=0ahUKEwjnidW8tsL9AhWzm2oFHbidDSEQ_AUIBigB


Imagini
https://www.google.com/search?q=europe&ie=UTF-8&source=lnms&tbm=nws&sa=X&ved=0ahUKEwjnidW8tsL9AhWzm2oFHbidDSEQ_AUIBygC


Știri
https://www.google.com/search?q=europe&ie=UTF-8&source=lnms&tbm=vid&sa=X&ved=0ahUKEwjnidW8tsL9AhWzm2oFHbidDSEQ_AUICCgD


Videoclipuri
https://www.google.com/search?q=europe&ie=UTF-8&source=lnms&tbm=bks&sa=X&ved=0ahUKEwjnidW8tsL9AhWzm2oFHbidDSEQ_AUICSgE


Cărți
https://www.google.com/advanced_search


Instrumente de căutare


Orice limbă
```

Figure 5: ”go2web.py -u <URL>” part 2



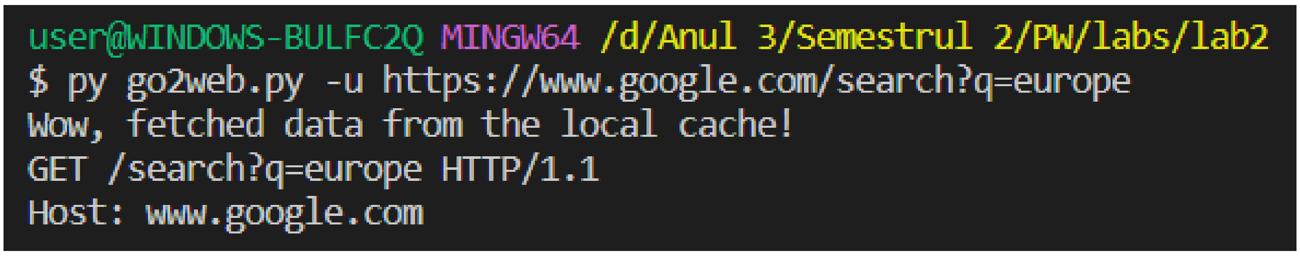
```python
import socket
import sys
import json
from bs4 import BeautifulSoup
from urllib.parse import unquote

def fetch_data(*, update: bool = False, json_cache: str, request:str):
    # caching process

    # if I want new data or not
    if update:
        json_data = None
```

```
Countries of Europe - Nations Online Project
www.nationsonline.org › oneworld › europe
Europe is the second-smallest continent, and it could be described as a large peninsula or as a subcontinent. Europe is the western portion of the Eurasian ...
https://www.google.com/url?q=https://european-union.europa.eu/index_en&sa=U&ved=2ahUKEwjnidW8tsL9AhWzm2oFHbidDSEQFnoECAUQAg&usg=AOvVaw1jJFc4oZDqe1i2jjhYWaa9


Your gateway to the EU, News, Highlights | European Union
european-union.europa.eu › index_en
Discover how the EU functions, its principles, priorities; find out about its history and member countries; learn about its legal basis and your EU rights.
https://www.google.com/url?q=https://european-union.europa.eu/easy-read_en&sa=U&ved=2ahUKEwjnidW8tsL9AhWzm2oFHbidDSEQFnoECAoQAg&usg=AOvVaw36Gh8cU1kinCw6jW6hZxXO


Easy to read – about the EU | European Union
european-union.europa.eu › easy-read_en
The European Union is a group of 27 countries in Europe. ... In the beginning, only 6 countries in Europe started working together:.
https://www.google.com/url?q=https://www.theguardian.com/world/europe-news&sa=U&ved=2ahUKEwjnidW8tsL9AhWzm2oFHbidDSEQFnoECAEQAg&usg=AOvVaw1Kd_a_58zI_NaF5FeFIXpO


Europe | The Guardian
www.theguardian.com › world › europe-news
Latest Europe news, comment and analysis from the Guardian, the world's leading ... A series that takes a pan-European lens to explain the challenges that ...
https://www.google.com/url?q=https://www.worldometers.info/geography/how-many-countries-in-europe/&sa=U&ved=2ahUKEwjnidW8tsL9AhWzm2oFHbidDSEQFnoECAQQAg&usg=AOvVaw2wLIR2K1G7
ONaiW98


How many countries in Europe? - Worldometer
www.worldometers.info › geography › how-ma...
There are 44 countries in Europe today, according to the United Nations. The full list is shown in the table below, with current population and subregion ...
```

Figure 6: ”go2web.py -u <URL>” part 3

There can be easily seen the implementation of HTTP request redirects, which gives us responses from different sections as: images, news, video, books, information, urls. As well we receive specific articles in english, in romanian and other languages. At the end we receive a lot of long urls that are related to the searched url, but is not exact like it.

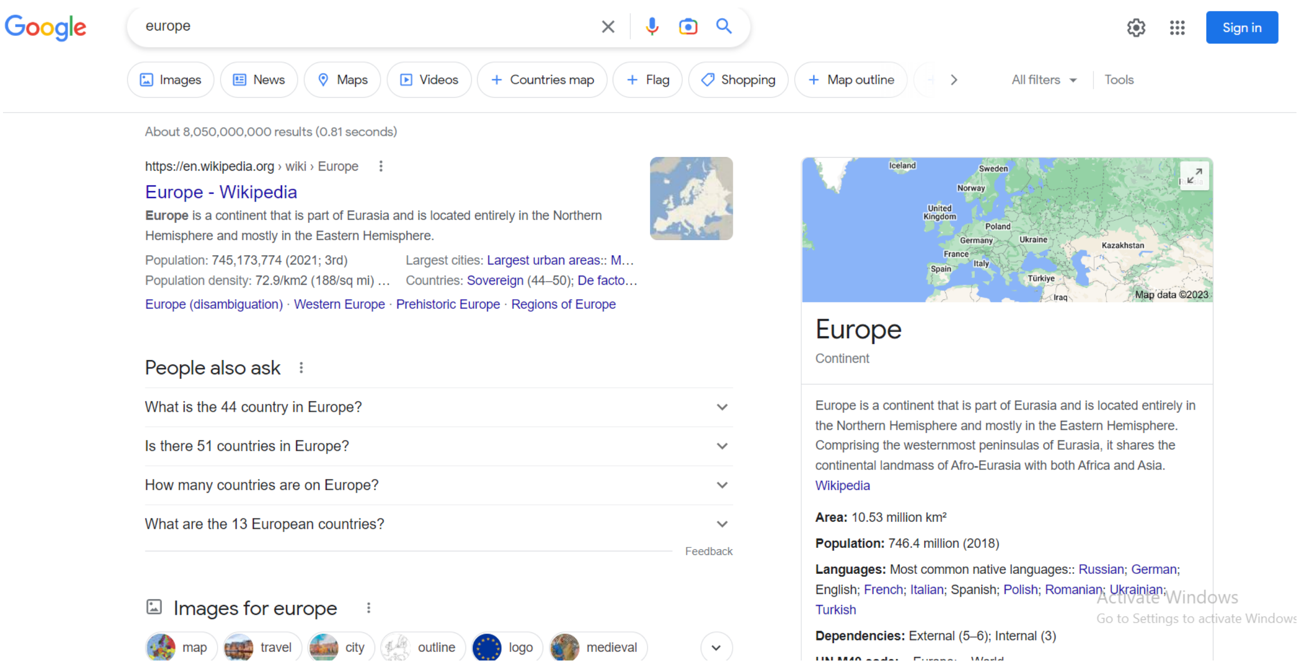
Here, regarding the HTTP caching mechanism, there was used the same function that initially needs a json file to store the cached. In this case, we requested the same information (about Europe), thus in the fetching function we open the file cache.json and read it, and if we do have json data and it was locally cached, then we are going to use it instead our new request, as it is shown in the Figure 7.



```
user@WINDOWS-BULFC2Q MINGW64 /d/Anul 3/Semestrul 2/PW/labs/lab2
$ py go2web.py -u https://www.google.com/search?q=europe
Wow, fetched data from the local cache!
GET /search?q=europe HTTP/1.1
Host: www.google.com
```

Figure 7: Fetched data from the local cache

In Figure 8 is represented that the urls can be aceessed.



About 8,050,000,000 results (0.81 seconds)

https://en.wikipedia.org › wiki › Europe

Europe - Wikipedia

Europe is a continent that is part of Eurasia and is located entirely in the Northern Hemisphere and mostly in the Eastern Hemisphere.

Population: 745,173,774 (2021; 3rd) Largest cities: Largest urban areas:: M...

Population density: 72.9/km2 (188/sq mi) ... Countries: Sovereign (44–50); De facto...

Europe (disambiguation) · Western Europe · Prehistoric Europe · Regions of Europe

People also ask

What is the 44 country in Europe?

Is there 51 countries in Europe?

How many countries are on Europe?

What are the 13 European countries?

Feedback

Images for europe

Europe

Continent

Europe is a continent that is part of Eurasia and is located entirely in the Northern Hemisphere and mostly in the Eastern Hemisphere. Comprising the westernmost peninsulas of Eurasia, it shares the continental landmass of Afro-Eurasia with both Africa and Asia. Wikipedia

Area: 10.53 million km²

Population: 746.4 million (2018)

Languages: Most common native languages:: Russian; German; English; French; Italian; Spanish; Polish; Romanian; Ukrainian; Turkish

Dependencies: External (5–6); Internal (3)

Figure 8: Accessed url

3 Conclusion

Due to this laboratory work, I practiced building and working with HTTP requests, as well with parsing HTML responses. I was also able to implement HTTP caching mechanism. Thus I learned about new libraries that helped a lot in parsing process and considerably eased the creation of structured final output.

References

[1] Beautiful Soup 4 Tutorial 1 - Web Scraping With Python, *https://www.youtube.com/watch?v=gRLHr664tXA*

Accessed on February 18, 2023.

[2] Beautiful Soup Documentation, *https://www.crummy.com/software/BeautifulSoup/bs4/doc/*

Accessed on February 25, 2023.