

Multithreaded News Client/Server Information System

Project Description

A client-server system that exchanges information about current news where the user requests for news headlines and sources from the server through filters such as keywords, category, and country. The server then fetches the data from the API and send the requested information to the client.

Semester

First semester of 2024/2025.

Group

- Group Number: A19
 - Course Code: ITNE352/ITCE320
 - Section Number: 01
 - Students Names: HASHEM SAEED ABDULLA ALKHANAIZI, ABDULLAH MOHAMMAD ISHTIAQ AFRIDI
 - Students IDs: 202203100, 202200767
-

Table of Contents

1- [Requirements](#)

2- [How to](#)

3- [The Scripts](#)

4- [Additional Concepts](#)

5- [Acknowledgments \(aAcknowledgments\)](#)

6- [Conclusion](#)

Requirements

1- Install the latest version of Python from Python's official website, which is: <https://www.python.org/downloads/>
(<https://www.python.org/downloads/>).

2- Install the required packages to run the project by running the following command in command prompt:

```
pip install requests
```

3- Clone the repository from GitHub and run it on Virtual Studio code

How to

1- Open two terminal windows, one for the client and one for the server

2- Navigate to the directory where the client and server files are in terminal

3- Start the server side first to make it ready to receive requests by running the script:

```
python server.py
```

4- Start the client side next and connect to the already running server using the script:

```
python client.py
```

5- The client side of the terminal window would ask you to input some information and choose from the menu, interact with it and retrieve the data.

The Scripts

Client side

The packages utilized in the client script are:

```
import socket
import json
import signal
import sys
```

Create a socket for the client and ask to input the client's name

```
client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
clientname = input('Enter Your Name: ')
```

Connect to the server after creating the socket

```
client_socket.connect((HOST, PORT))
client_socket.sendall(clientname.encode())
print("Connected to the server.")
```

The main menu that the user has to select from to get to the information wanted

```
print("1 - search by headlines")
print("2 - List of sources")
print("3 - Quit")
x = int(input("Choose your option "))
```

Displays the received list of results and details for a specified result that the user chooses

```
received = client_socket.recv(4000)
results = json.loads(received.decode('utf-8'))
```

Server side

The packages utilized in the server script are:

```
import requests
import json
import socket
import threading
```

Define URLs for the fetched sources and headlines

```
headline = "https://newsapi.org/v2/top-headlines?apiKey=4411d3bdab63427c91fcef22bed1b3f0"
sources = 'https://newsapi.org/v2/top-headlines/sources?apiKey=4411d3bdab63427c91fcef22bed1b3f0'
```

The following script is a function that handles the headline data and return a brief list:

```
def handle_headline(data):
```

The following script is a function that handles the source data and return a brief list:

```
def handle_sources(data):
```

Defines a socket and starts a TCP connection

```
server_socket=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
server_socket.bind(('localhost', 12345))
server_socket.listen(5)
```

Additional concepts

- 1- Threading.local: Each and every thread is separated from the other thread and with their own parameters that the users can borrow whenever they want to make a request from the server side.
 - 2- Multithreading: The server script uses the threading module to handle several client connections at once. The server may handle requests from several clients simultaneously thanks to this multithreading technique. By giving every client connection its own thread. Using multithreading, the server increases its efficiency in sending the requested data to clients
 - 3- Interrupt: There is a function defined in the client side of the code that interrupts the signal and closes the terminal whenever the user tries to copy something written within the terminal using Ctrl+c. The server disconnects with the client and client side of the terminal prints a message "Caught interrupt signal. Closing connection".
-

Acknowledgments

- News provided and used in this project was taken from NewsAPI
 - The usage of GitHub made sharing of the project very easy
 - A big thanks to Dr. Mohammad A. Almeer for teaching us and guiding us on the project and throughout the course as a whole.
-

Conclusion

By demonstrating effective network programming, the Multithreaded News Client/Server System makes it possible to retrieve news in real time using NewsAPI. The server manages several customers with ease by utilizing multithreading, guaranteeing quickness and dependability. Our knowledge of client-server architectures, API integration, and collaborative development has grown as a result of this project, giving us a solid basis for upcoming technological difficulties.
