News API Client-Server System

Overview

This project implements a client-server system that interacts with the News API to fetch headlines and sources. The server listens for client requests, retrieves data from the API, and sends it back to the client. The client can query headlines based on categories, countries, and other parameters, and view detailed information about specific results.

Features

Server:

- Handles multiple simultaneous connections from clients.
- Provides a main menu for users to select between searching headlines or listing sources.
- Allows the client to request headlines by category or search for sources by country, language, or category.
- Saves the fetched data in JSON format for testing and future analysis.

Client:

- Connects to the server and sends user input for various requests.
- Displays results and allows the user to select specific items to view detailed information.
- Continues to interact with the server until the user chooses to quit.

Prerequisites

- Python 3.10
- Requests library for fetching data from the News API.

To install the required dependencies:

```
pip install requests
```

Setup

1. Clone the repository:
2. Run the server:

   ```
   python server.py
   ```

3. Run the client:

   ```
   python client.py
   ```

Server Usage

Once the server is running, it will be waiting for client connections. The server listens on localhost at port 12345. It can handle up to 3 simultaneous connections.

Server Main Menu The server will display the following main menu options to the client:

```
Main Menu:
1. Search Headlines
2. List Sources
3. Quit
```

Headlines Menu If the user selects the "Search Headlines" option, the client will be prompted to choose one of the following categories:

- business
- general
- health
- science
- sports
- technology

The server will fetch the top headlines for the selected category and display them to the user. The user can then select a headline to view its details.

Sources Menu If the user selects the "List Sources" option, the client will be prompted to choose one of the following parameters:

- Country (e.g., au, ca, jp, etc.)
- Category (e.g., business, general, etc.)
- Language (e.g., en, ar)

The server will fetch sources from the selected country or category and display them. The user can then select a source to view its detailed information.

JSON Data Storage For every request, the server saves the results in a JSON file in the client_data folder. The filenames follow the format:

```
<client_name>_<choice>_<group_id>.json
```

The data includes a list of headlines or sources, which is then sent to the client.

Quit Option The client can choose to quit the session at any time. Upon quitting, the server closes the connection.

Client Usage

1. Run the client on your machine.
2. The client will automatically connect to the server and display the main menu.
3. You can interact with the menus by selecting options for headlines or sources.
4. The client will display the results and allow you to select specific headlines or sources for detailed information.

Example of Server Interaction

Main Menu

```
Main Menu:
1. Search Headlines
2. List Sources
3. Quit
```

Client selects "1" for Headlines

```
Choose a category: business, general, health, science, sports, technology.
```

The server will fetch and display the top headlines for the selected category.

Client selects a specific headline

```
Select a headline to view details:
```

The server responds with the full details of the selected headline, including the source, author, title, URL, description, and publishing time.

Example of Client Interaction

```
Main Menu:
1. Search Headlines
2. List Sources
3. Quit
```

- If you choose option 1, you can select a category and see the results.
- If you choose option 2, you can select a country or language and see the list of sources.

Error Handling The client and server both handle errors such as invalid selections, network issues, and API failures. The client will prompt the user to select a valid option if an invalid choice is made.

Contributing If you'd like to contribute to the project, feel free to fork the repository and submit a pull request. Here's how to do it:

1. Fork the repository.
2. Create a new branch for your feature or bug fix.
3. Commit your changes.
4. Push the changes to your fork.
5. Create a pull request.

Happy coding!