**Project Overview**

This project is a **News Aggregator REST API** built with Flask. It scrapes, categorizes, and serves news articles through several API endpoints. The project supports retrieving articles, filtering them by date or category, and searching articles by keywords.

**Table of Contents**

* [Features](#features)
* [Tech Stack](#tech-stack)
* [Setup Instructions](#setup-instructions)
* [API Endpoints](#api-endpoints)
* [Usage Examples](#usage-examples)
* [Postman Collection](#postman-collection)
* [Project Files](#project-files)
* [License](#license)

**Features**

1. Scrapes news from external sources like BBC.
2. Categorizes news articles into various categories (e.g., Sports, Technology, Politics).
3. Exposes a REST API to retrieve and filter articles.
4. Allows keyword-based search on news articles.

**Tech Stack**

* **Python** (Flask, Pandas)
* **Postman** (for API testing and collection)
* **JSON** (for data serialization)
* **HTML scraping libraries** (like BeautifulSoup, requests)
* **CORS** (Cross-Origin Resource Sharing)

**Setup Instructions**

**Prerequisites**

* Python 3.10+
* Pip (Python package installer)

**Install Dependencies**

1. Clone the repository:

git clone https://github.com/your-username/news-aggregator.git

cd news-aggregator

1. Install the required packages using the requirements.txt file:

pip install -r requirements.txt

**Running the Project**

1. Start the Flask server:

Python3 Rest\_API.py

1. The server will be running on <http://127.0.0.1:5000/>.

**API Endpoints**

**1. Get All Articles**

* **Endpoint**: /articles
* **Method**: GET
* **Description**: Retrieve all articles with optional filtering by date or category.
* **Parameters**:
  + start\_date: Optional (Filter by start date)
  + end\_date: Optional (Filter by end date)
  + category: Optional (Filter by category)
* **Response**: List of articles in JSON format.

**2. Get Article by ID**

* **Endpoint**: /articles/<id>
* **Method**: GET
* **Description**: Retrieve an article by its unique ID.
* **Response**: Single article in JSON format.

**3. Search Articles by Keyword**

* **Endpoint**: /search
* **Method**: GET
* **Description**: Search for articles by keywords in the title.
* **Parameters**:
  + q: Required (Keyword to search for in article titles)
* **Response**: List of matching articles in JSON format.

**Usage Examples**

**1. Get All Articles**

**Request**:

curl -X GET <http://127.0.0.1:5000/articles>

**Response** (sample):

[

{

"id": 0,

"Title": "Israel carrying out 'extensive' strikes in Lebanon",

"Category": "Other",

"Source": "BBC",

"URL": "https://www.bbc.com/news/live/c5y32qew9z2t"

},

{

"id": 1,

"Title": "Bowen: Israel is gambling Hezbollah will crumple",

"Category": "Other",

"Source": "BBC",

"URL": "https://www.bbc.com/news/articles/c93pg1qpxxzo"

}

]

1. **Get Specific Article by ID**

curl -X GET <http://127.0.0.1:5000/articles/1>

**Response**:

{

"id": 1,

"Title": "Bowen: Israel is gambling Hezbollah will crumple",

"Category": "Other",

"Source": "BBC",

"URL": "https://www.bbc.com/news/articles/c93pg1qpxxzo"

}

1. **Search Articles by Keyword**

curl -X GET <http://127.0.0.1:5000/search?q=Israel>

**Response**:

[

{

"id": 0,

"Title": "Israel carrying out 'extensive' strikes in Lebanon",

"Category": "Other",

"Source": "BBC",

"URL": "https://www.bbc.com/news/live/c5y32qew9z2t"

}

]

**Postman Collection**

For testing and usage, import the Postman collection into your Postman workspace. The collection provides pre-configured requests for each of the API endpoints.

1. Download the postman\_collection.json file.
2. Import it into Postman.
3. Modify base URLs if required (http://127.0.0.1:5000 by default).

**Project Files**

1. **app.py**: Main Flask application code.
2. **requirements.txt**: List of dependencies.
3. **postman\_collection.json**: Postman collection for testing API.
4. **README.md**: Project documentation.
5. **data/articles.csv**: Dataset containing news articles (if any).

**Sample requirements.txt**

Flask==2.1.1

pandas==1.3.3

Flask-Cors==3.0.10

**README File for Git**

Save this content as README.md in your project root to provide full documentation.

If you need anything else or further changes, feel free to ask!