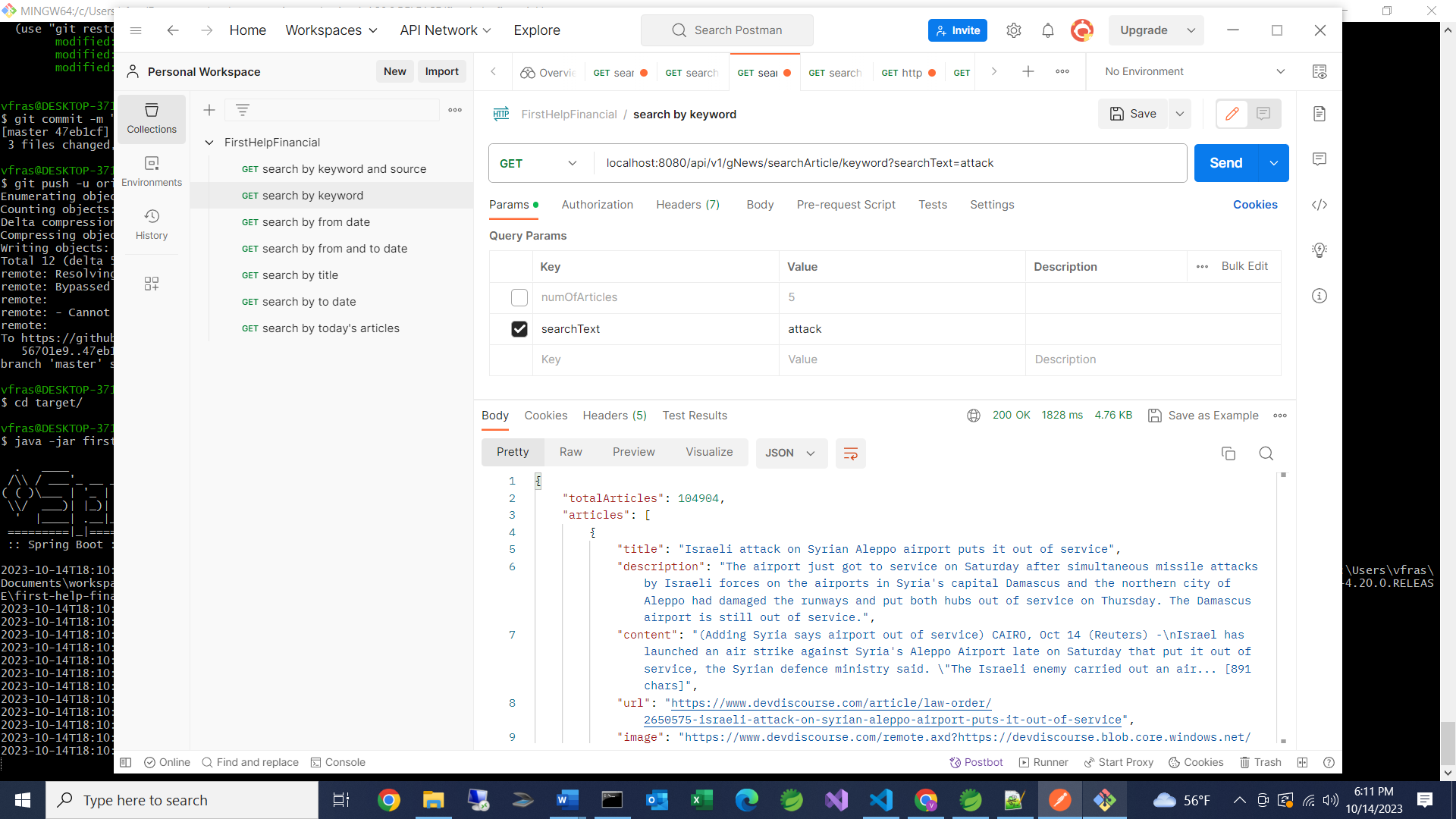
**How to use the gNewsAPI service**

Start the service:

1. You can start the server by running the executable or running it on your local IDE. Refer to the README file for instructions on how to do this.

API calls using postman:

1. In the project folder, there should be a folder named “Postman.” Inside that folder, there is a .json file FirstHelpFinancial.postman\_collection.
2. You will need to import that file into Postman which should load all of the different API calls. There is one call for each different API that I have implemented in the code. (Note: When it comes to the Date APIs there are 3 separate postman GET calls, that go to just one backend API endpoint).
3. You can refer to this link on any questions/difficulties with Postman: <https://docs.tink.com/entries/articles/postman-collection-for-account-check>
4. Here is a screenshot of Postman running the search by keyword API:



1. Once you have the calls uploaded, you can modify the values of the “Params” section to your liking.

Calling API service without Postman:

1. Once the service runs, you can open a web browser and paste the URLs directly.
2. I will provide URL’s below (in these examples you are looking for articles with the word “attack”):

localhost:8080/api/v1/gNews/searchArticle/keyword?searchText=attack

localhost:8080/api/v1/gNews/searchArticle/source?searchText=attack&source=The Australian Financial Review

localhost:8080/api/v1/gNews/searchArticle/date?searchText=attack&numOfArticles=8&from=07/01/2023

localhost:8080/api/v1/gNews/searchArticle/date?searchText=attack&numOfArticles=8&from=07/01/2023&to=07/04/2023

localhost:8080/api/v1/gNews/searchArticle/title?numOfArticles=5&searchText=attack

localhost:8080/api/v1/gNews/searchArticle/date?searchText=attack&numOfArticles=8&to=10/01/2023

localhost:8080/api/v1/gNews/searchArticle/today?searchText=attack

Information about the API:

1. The base URL for every call will be “localhost:8080/api/v1/gNews/searchArticle”
2. From this base you have 3 possible different paths:
3. /keyword - here you will be only searching articles by the value defined in the searchText parameter
4. /title – you will search for articles where the searchText appears in the title
5. /date – you will search for articles where the searchText appears in articles with date parameters
6. /today - you will search for articles where the searchText appears anywhere in the article/title that is on the current date.
7. There is another parameter called numOfArticles. You can include this in all of the API calls you make. The value must be 1 or greater. If the value is greater than 10, you will only get back 10 articles. This is due to the limitation of the apiKey used to make the API calls. Due to only having a free account, you are limited in the number of articles you can return. If this field is omitted from the API call, it will default to 5.
8. The date parameters in path “/date” look for two additional fields in the API call.
9. There is a “from” field and a “to” field. By specifying the “from” field, you will get articles published most recently and back in time until the “from” date. (max 10)
10. By specifying the “to” field, you will get all articles starting on the “to“ date and other articles that are further back in time. (max 10)
11. By specifying both a “from” and “to” date, you will get articles between the two dates, starting with the “to” date and going backward until the “from” date (max 10).
12. When providing dates, multiple date formats work, as long as Java can take the string provided. For example, 07/01/2023 and 2023/07/01 are both acceptable values for those fields.

Caching:

Each time you make a unique request, you will make the API call to the client server and return the information. Depending on which call you will get one of the following log statements in your console:

***log***.info("making api call to get articles containing " + searchText);

***log***.info("making api call to get articles containing " + searchText + " in the title");

***log***.info("making api call to get articles containing " + searchText + " from the source " + source);

***log***.info("making api call to get articles containing " + searchText+ " from date "+ from + " and to date "+to);

***log***.info("making api call to get articles containing " + searchText + " form today");

If you make another API call with the same information, you will grab the information from the cache and you won’t see the same print statement a second time.