**Database Project 04**

**Deadline: May 7th (FRIDAY) 11:59pm**

**Overview**

In this assignment you are going to implement a news data crawler. News articles will be crawled from <https://newsapi.org>, an API service that gathers news from over 30K news sources and blogs from all over the world. The data crawler will send requests to the API to get the latest news articles of a topic of interest chosen by the user, saving the articles into a MongoDB collection.

The initial code for this project is available at:

<https://github.com/thyagomota/21SCS3810/tree/main/db04_news>

**Implementation**

Before you are able to use the News API service, you need to request a key. Go to <https://newsapi.org> and click on “Get API Key”. Complete the registration process and save your API key in the config.properties file. Requests to the API are sent using the customized URL below:

https://newsapi.org/v2/everything?apiKey=<apiKey>&sortBy=publishedAt&q=<topic>&page=<page>&pageSize=<pageSize>

Note that the following parameters need to be informed whenever you make calls to the API:

* <apiKey>: that's your API key
* <topic>: a string with the news topic to be searched
* <page>: the page number, starting at 1
* <pageSize>: the number of news article to be retrieved in each API call (we suggest using 10)

To avoid making having your API key locked out, you should limit the number of API calls per unit of time. We suggest a very conservative parameter for that: one API call every 2 minutes (see PREEMPTIVE\_BACKOFF\_TIME parameter). Also, avoid page sizes too large (the last time I checked the API was not accepting pages with values > 30).

To actually make the calls to the service provider you will use Unirest, a lightweight HTTP request client library. Read the documentation for the Java implementation of Unirest [here](http://javadox.com/com.mashape.unirest/unirest-java/1.4.5/com/mashape/unirest/http/Unirest.html) and figure it out how to get the JSON response's body as a String (highlights are hints).

Once you have the response as a JSON string, use Google's GSON library to map it to Java objects. The model that you need is given to you and it is represented by the classes Response, Article, and ArticleSource. Finish the toDocument methods in the Article and ArticleSource classes. Read the documentation about org.bson.Document [here](https://mongodb.github.io/mongo-java-driver/3.9/javadoc/org/bson/Document.html).

[This](http://tutorials.jenkov.com/java-json/gson.html) tutorial explains how to use GSON to parse a JSON string. Note that the NewsDataCrawler class already has a Gson instance variable (named gson) ready for you to use (no need to instantiate it). After you are able to extract all Article objects from the News API response, your next goal is to insert those articles into a MongoDB collection named articles.

Finish the program by asking the user to choose the topic of interest when the program starts. To avoid API call errors, make sure to replace any spaces in the topic string with %20. The replacement should be transparent to the user.

**Queries**

Once you are done collecting 100 news articles and saving them in a database, answer the following queries using MongoDB query language:

* The total number of documents in the collection.
* All articles from a given author (picked from your collection), sorted alphabetically.
* The total number of articles per article source (name), also alphabetically.
* All articles having a specific word (picked from your collection) in their content.
* A different query created by you.

Save the code for your queries in a file named db04.js.

**Deliverables**

A zip file containing the following files:

* db04.js
* Article.java
* ArticleSource.java
* NewsDataCrawler.java

**I only accept zip format!!!**

**It should be obvious, but please don't send your config.properties file!!!**

**Rubric**

+10 TO-DO in Article (toDocument method)

+10 TO-DO in ArticleSource (toDocument method)

+30 searchArticles method

+10 successfully sends API request using Unirest and with all of the required parameters

+10 successfully converts the response to a list of Article objects using Gson

+10 successfully inserts all of the Article objects into a MongoDB collection

+15 NewsDataCrawler's main method

+5 reads the topic of interest from the user, replacing any spaces with %20

+10 sends searchArticles request every 2m, incrementing the page number between calls

+35 MongoDB queries (+7 each one)

-5 didn't named participants in the team

-5 didn't use the deliverable format (zip and the exact files asked inside the zip)

Total: +100