MESIIN480323 : No SQL 19/01

PROJECT REPORT

On

ElasticSearch

reuters.json

GABRIEL CHAIX & MALO LEROY & CHARLES TERRIER

ÉCOLE SUPERIEURE D'INGENIEURS LEONARD DE VINCI

ANNEE 2023 / 2024

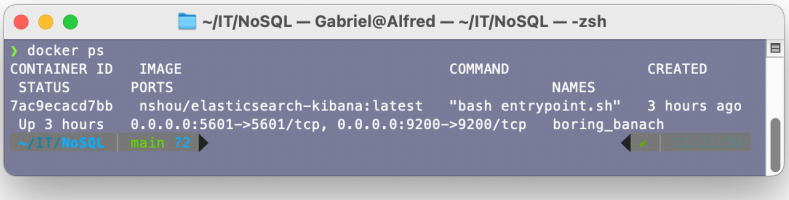
INDEX

INTRODUCTION AND PROBLEM STATEMENT

Reuters is a news agency founded in 1851 in London. It is one of the global and generalist news agencies, an activity that represents a part of its revenue, mainly devoted to financial information. It is one of the largest news agencies in the world. We want to create a MongoDB database with reuters articles and query it.

I. Data Importation

In a first manner, we'll setup the docker container with nshu image and make sure everything is working properly, we try to connect on port 5601.



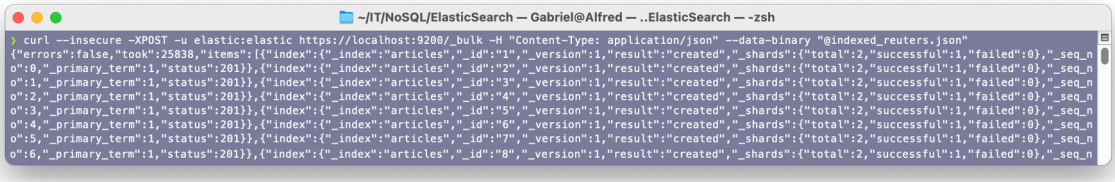
Now we must import our data using python to write a script. To do this, we run the following code modify our [json file] to match the indexing bulk format of elastic search.

```
import json

# Open the original file for reading
with open('reuters.json', 'r') as original_file:
    data = original_file.readlines()

with open('indexed_reuters.json', 'w') as indexed_file:
    for i, line in enumerate(data):
        obj = json.loads(line)
        # Remove the "_id" field
        obj.pop('_id', None)
        # Create the index line
        index_line = '{"index":{"_index":"articles","_id":' + str(i + 1) + '}}\n'
        indexed_file.write(index_line)
        # Write the JSON object to the indexed file
        json.dump(obj, indexed_file)
        indexed_file.write('\n')
```

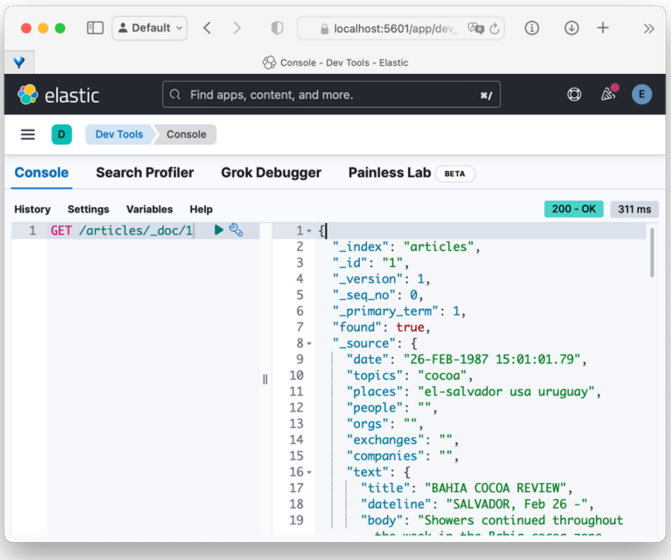
Then here is the formatting script we've wrote ←

After executing the script, goes to the format we needed.

We can now import the dataset.



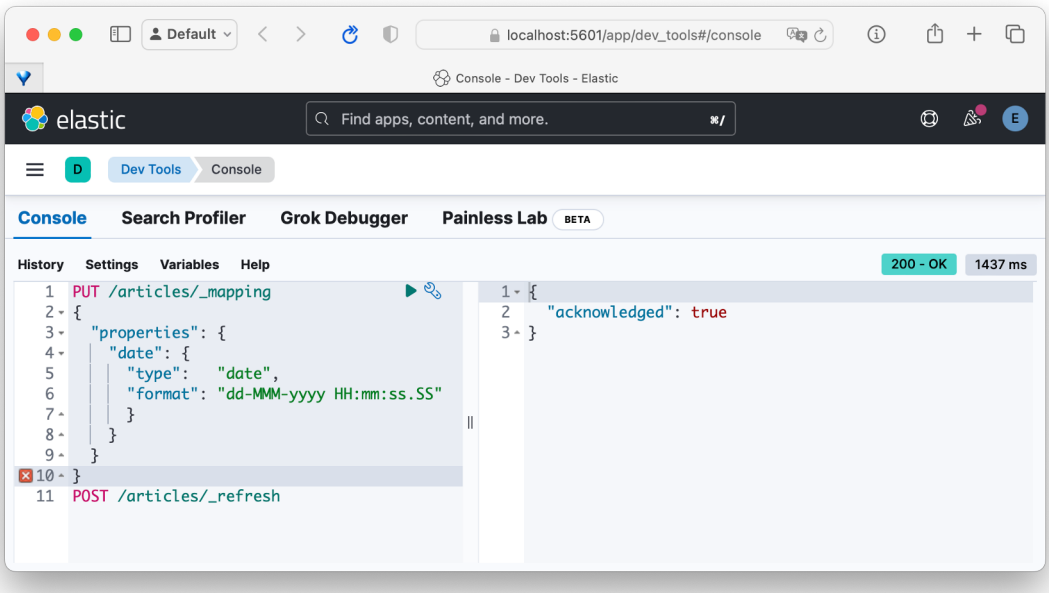
After importing all the rows, we make sure that we have the count and that each row is correct.



Using Kibana compass, we can see everything looks alright.

II. Data cleaning

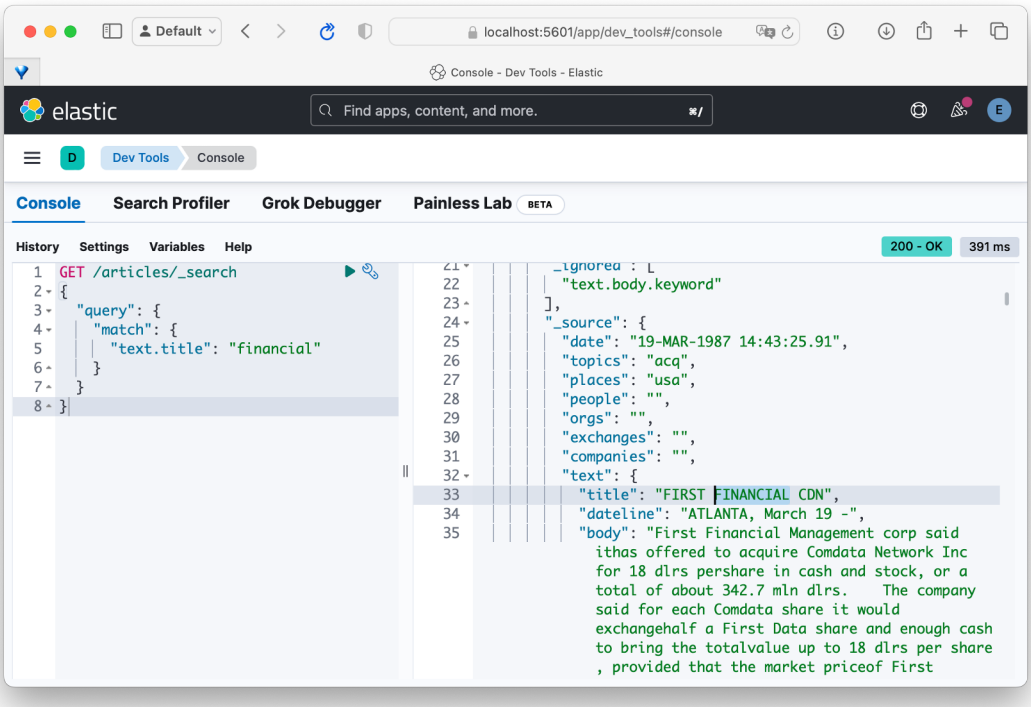
The only cleaning, we success to do had been to update all date entry to the valid date type. Compared to other noSQL language, we found hard to word with elastic search.



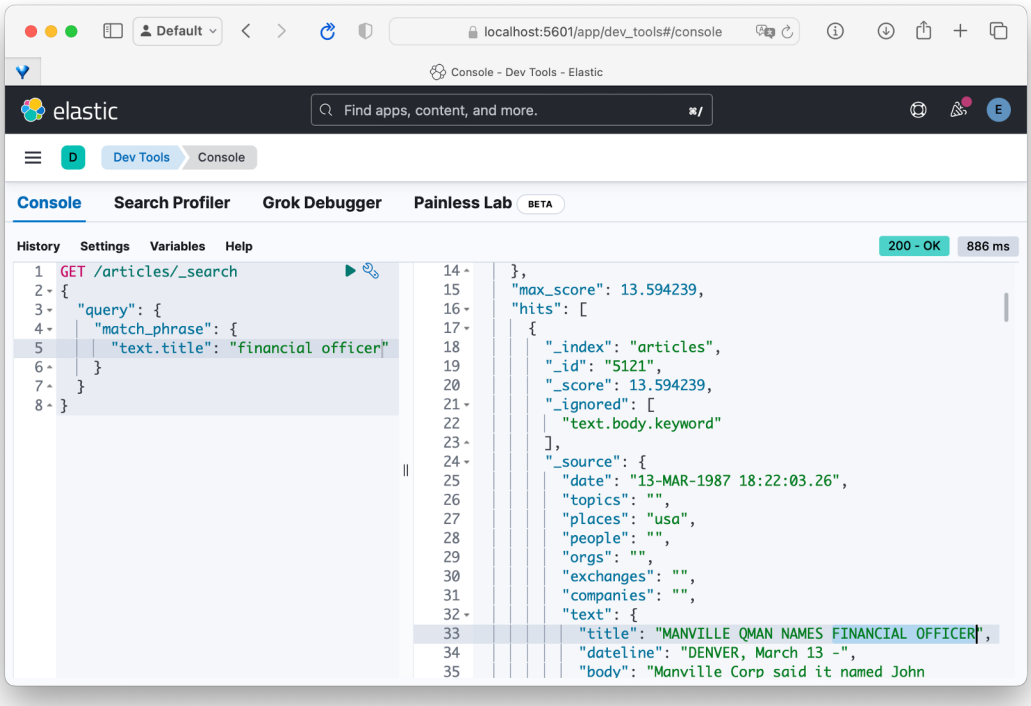
III. Querys

A. Simple querys (6)

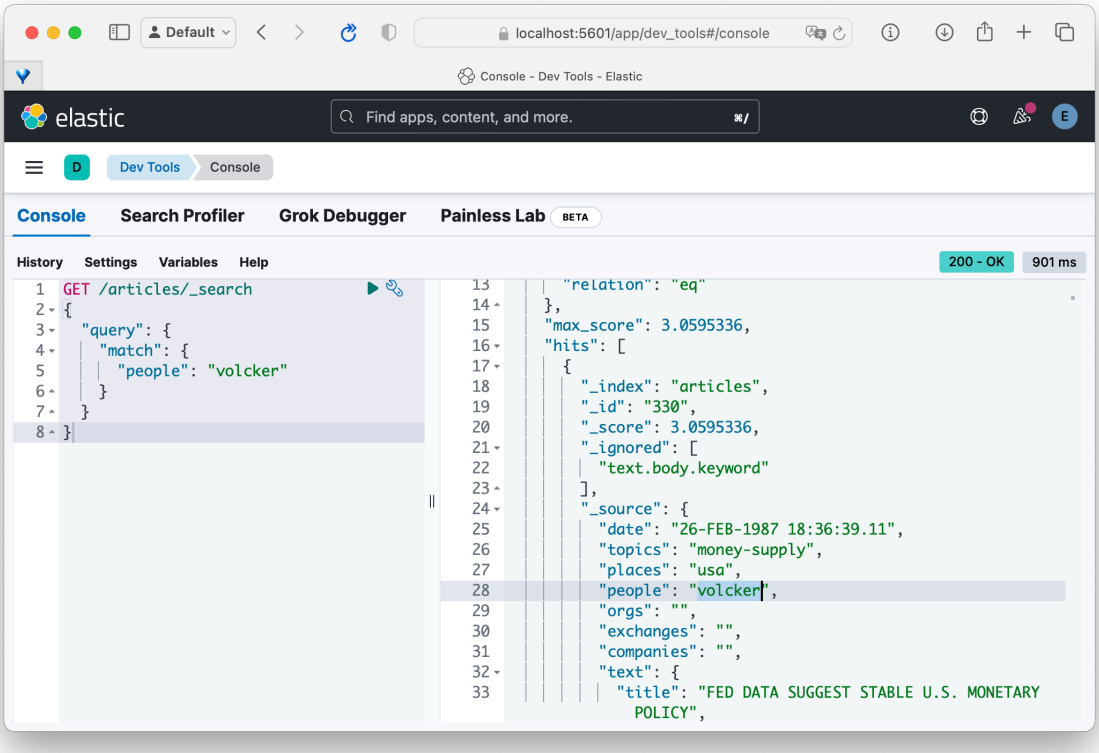
1. Every article which title match “financial”



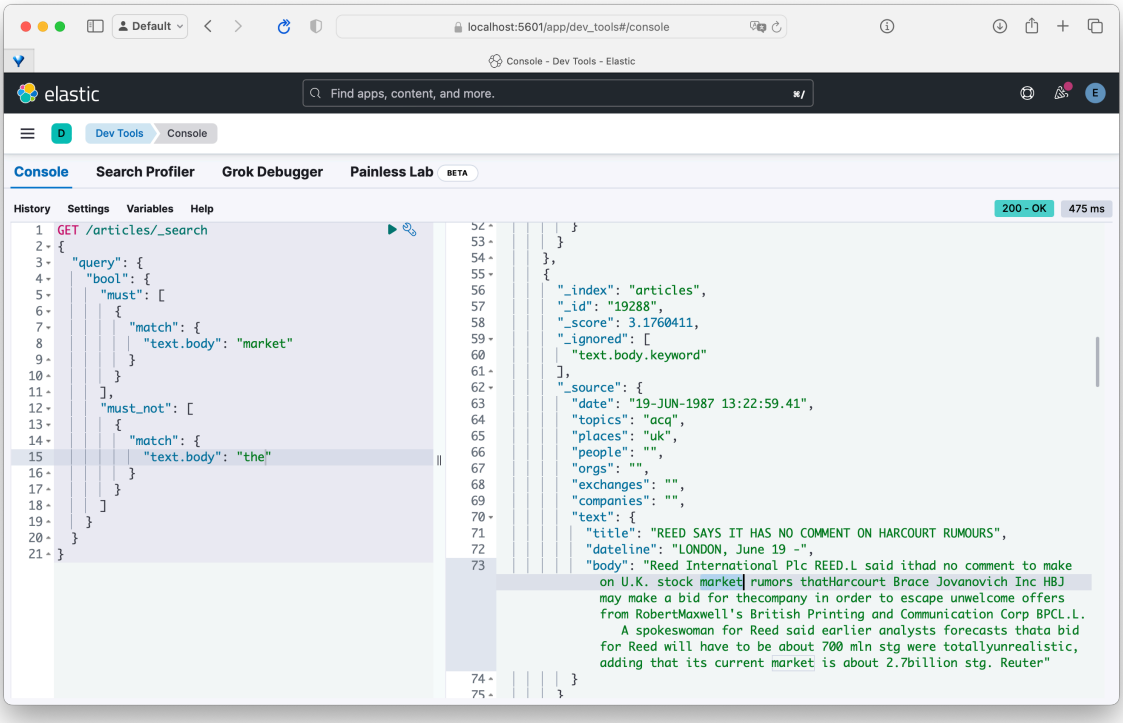
2. Every article which exact match “financial officer”



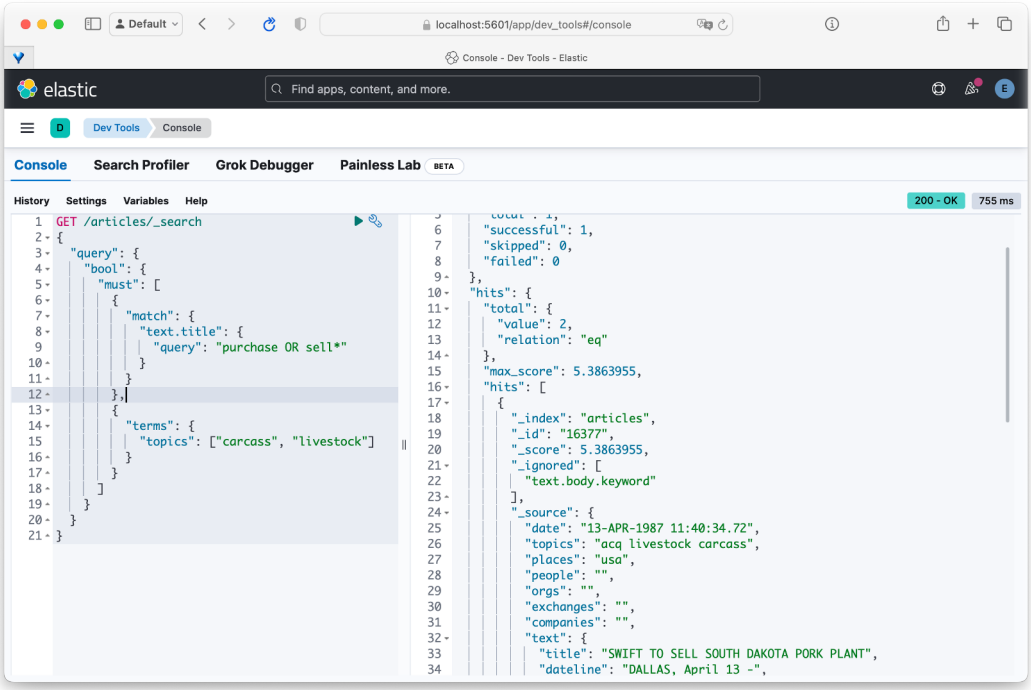
3. Articles written by volcker



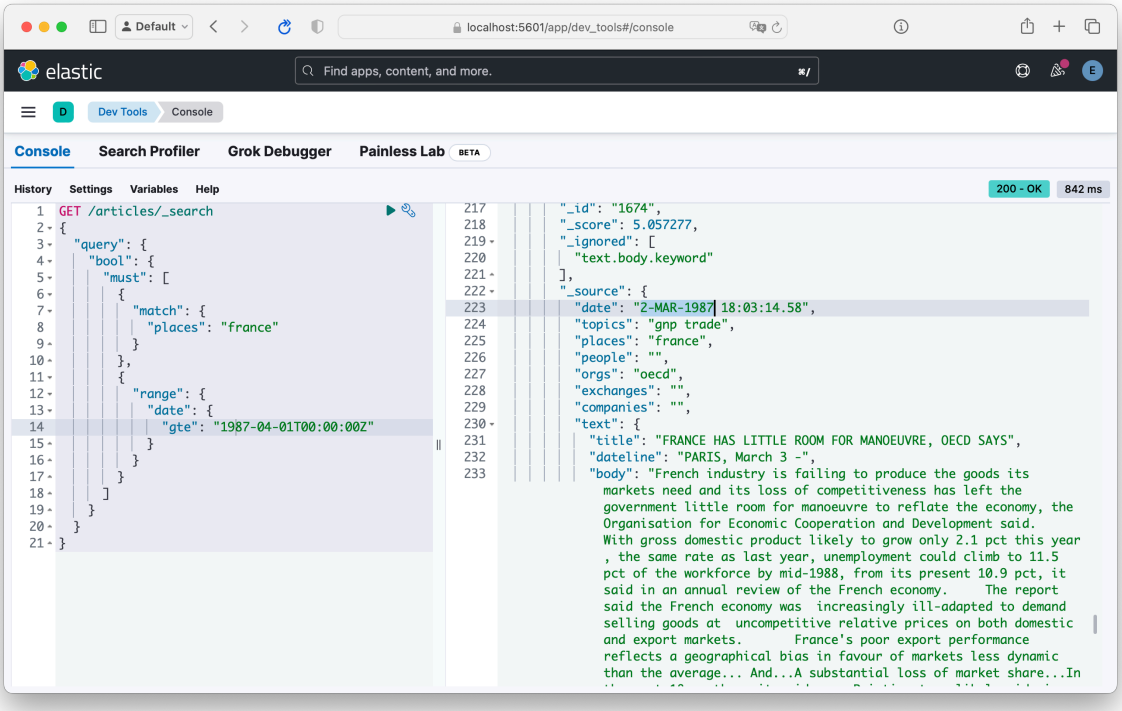
4. Article with a body containing “market” but not “the”



5. Find all articles that mention a purchase or sale of stocks by individuals or organizations, in particular articles related to agricultural commodities such as pork or livestock

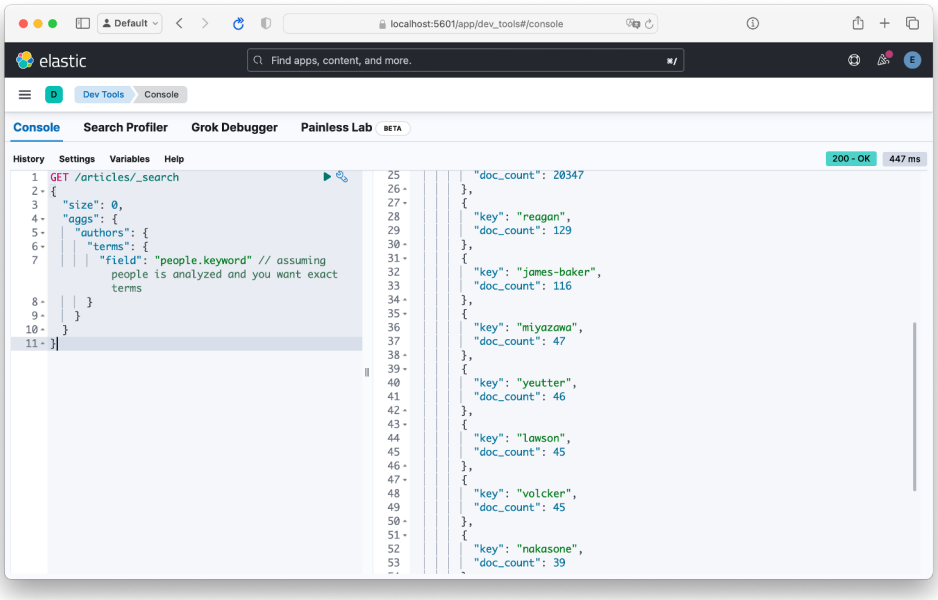


6. Articles whose place is France whose date is after mars-1987



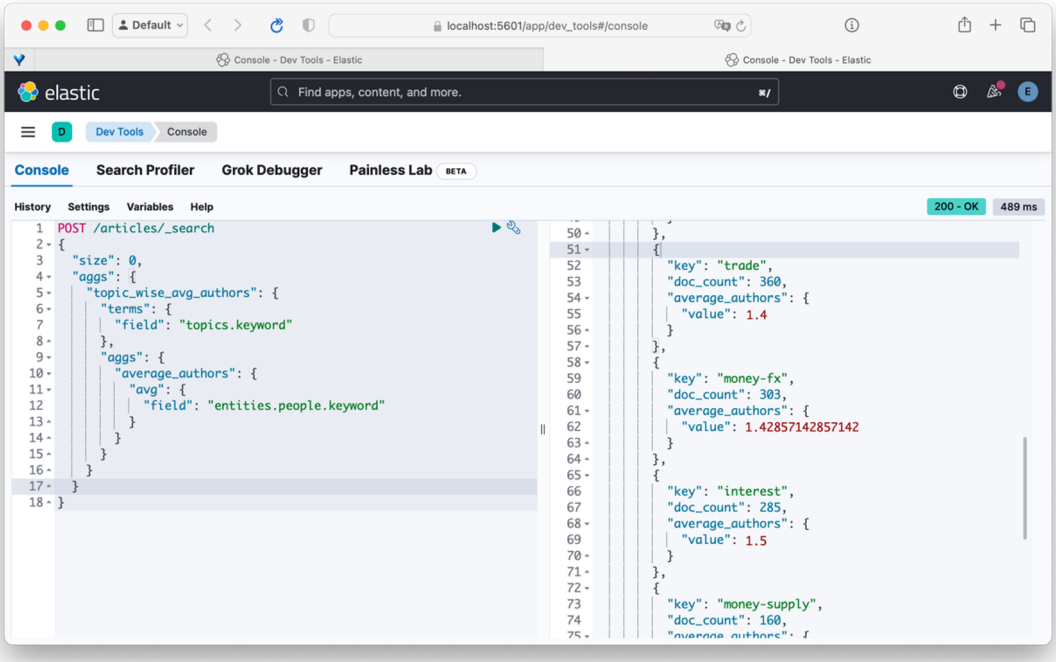
B. Complex querys (2)

1. Give the number articles written by each author?



Within the authors *aggregation*, we specify another aggregation type called *terms*, which creates a separate bucket for each unique term found within the specified field. In this case, we want to create one bucket per unique author, so we choose the people field.

2. For each topic, give the average number of people that wrote on. Sort the result decreasingly.



C.Hard query (1)

1. How much articles have been published per month?