Pythonist (https://soumilshah1995.blogspot.com/)

Tuesday, June 30, 2020

4 Ways to do Pagination or scrolling in Elastic Search Tutorials

Elastic Search Tutorials

4 Ways to do Pagination or scrolling in Elastic Search Tutorials

- Soumil Nitin Shah¶
- Bachelor in Electronic Engineering | Masters in Electrical Engineering | Master in Computer Engineering |
- Website : https://soumilshah.herokuapp.com (https://soumilshah.herokuapp.com)
- Github: https://github.com/soumilshah1995 (https://github.com/soumilshah1995)
- Linkedin: https://www.linkedin.com/in/shah-soumil/ (https://www.linkedin.com/in/shah-soumil/)
- Blog: https://soumilshah1995.blogspot.com/ (https://soumilshah1995.blogspot.com/)
- Youtube : https://www.youtube.com/channel/UC_eOodxvwS_H7x2uLQa-svw?view_as=subscriber (https://www.youtube.com/channel/UC_eOodxvwS_H7x2uLQ view_as=subscriber)
- Facebook Page : https://www.facebook.com/soumilshah1995/ (https://www.facebook.com/soumilshah1995/)
- Email : shahsoumil519@gmail.com

Method 1:

Step 1:

In [1]:

```
try:
    import os
    import sys

    import elasticsearch
    from elasticsearch import Elasticsearch
    import pandas as pd

    print("All Modules Loaded ! ")
except Exception as e:
    print("Some Modules are Missing {}".format(e))
```

```
All Modules Loaded !
```

Step 2:

In [2]:

```
def connect_elasticsearch():
    es = None
    es = Elasticsearch([{'host': 'localhost', 'port': 9200}])
    if es.ping():
        print('Yupiee  Connected ')
    else:
        print('Awww it could not connect!')
    return es
es = connect_elasticsearch()
```

```
Yupiee  Connected
```

Step 3: Define Query

In [3]:

```
myquey = {
   "_source": [],
   "size": 10,
   "query": {
      "bool": {
         "must": [],
         "filter": [
            {
               "exists": {
                  "field": "director"
               }
            }
         ],
         "should": [
            {
               "match_phrase": {
                  "director": "Richard "
               }
            }
         ],
         "must_not": []
      }
   }
}
```

Step 4:

Elastic Search

- index -> name of the index name
- Scroll -> How long you want the scroll to stay in his case 2m
- Size -> How many records you need in each cycle
- Body-> ELK Query

In [4]:

```
res = es.search(
  index = 'netflix',
  scroll = '2m',
  size = 10,
  body = myquey)
```

In [5]:

```
counter = 0
sid =  res["_scroll_id"]
scroll_size = res['hits']['total']
scroll_size = scroll_size['value']


# Start scrolling
while (scroll_size > 0):

    #print("Scrolling...")
    page = es.scroll(scroll_id = sid, scroll = '10m')

    #print("Hits : ",len(page["hits"]["hits"]))

    # Update the scroll ID
    sid = page['_scroll_id']

    # Get the number of results that we returned in the last scroll
    scroll_size = len(page['hits']['hits'])

    #print("Scroll Size {} ".format(scroll_size))

    # Do something with the obtained page
    counter = counter + 1

print("Total Pages  : {}".format(counter))
```

```
Total Pages  : 427
```

Method 2:

- the idea Goes Like this we need to map page number and we divide the search into parts
- say the size was 500
- Page 1 -> 0-10
- Page 2 -> 10-20
- All this time the size is same the query is same all that is changing is from and to word
- think this way you will only query once and create a hashmap key are page number and value would be sliced Records

In [6]:

```
res = es.search(
  index = 'netflix',
  size = 100,
  body = myquey)
```

In [7]:

```
data = res["hits"]["hits"]
```

In [17]:

```
hashmap = {}
```

In [ ]:

```
step = 2
hashmap = {}
for i in range(len(data)):
    if i==0:
        hashmap[i] = data[0:step]
    else:
        startIndex = step * i
        EndIndex =  ((i+1) * (step))
        sample = data[startIndex:EndIndex]
        hashmap[i] = sample
```

Method 3:

- The approach we are taking here is basically
- page 1 correspond to from 0
- page 2 correspond to from 10
- idea is eevry page keep increment the from varibale

First Time Query Becomes

In [ ]:

```
myquey = {
   "_source": [],
   "size": 10,
    "from":0
   "query": {
      "bool": {
         "must": [],
         "filter": [
            {
               "exists": {
                  "field": "director"
               }
            }
         ],
         "should": [
            {
               "match_phrase": {
                  "director": "Richard "
               }
            }
         ],
         "must_not": []
      }
   }
}
```

Secodn Time Query Becomes

In [20]:

```
myquey = {
   "_source": [],
   "size": 10,
    "from":10,
   "query": {
      "bool": {
         "must": [],
         "filter": [
            {
               "exists": {
                  "field": "director"
               }
            }
         ],
         "should": [
            {
               "match_phrase": {
                  "director": "Richard "
               }
            }
         ],
         "must_not": []
      }
   }
}
```

Method 4: Search After Query

In [19]:

```
myquey = {
   "_source": [],
   "size": 10,
   "query": {
      "bool": {
         "must": [],
         "filter": [
            {
               "exists": {
                  "field": "director"
               }
            }
         ],
         "should": [
            {
               "match_phrase": {
                  "director": "Richard "
               }
            }
         ],
         "must_not": []
      }
   }
}
```

In [21]:

```
res = es.search(
  index = 'netflix',
  size = 100,
  body = myquey)
```

In [22]:

```
def create_scroll(res):
    """
    :param res: json
    :return: string
    """

    try:
        data = res.get("hits", None).get("hits", None)
        data = data[-1]
        score = data.get("_score", None)
        scroll_id_ = data.get("_id", None)
        unique_scroll_id = "{},{}".format(score, scroll_id_)
        return unique_scroll_id
    except Exception as e:
        return "Error,scroll error "
```

In [23]:

```
scroll = create_scroll(res)
```

This is out unique Scroll

In [24]:

```
scroll
```

Out[24]:

```
'0.0,8URc93IB135PBBnB55dH'
```

Next time we will pass this scroll

In [25]:

```
score, scroll_id = scroll.split(",")
```

In [26]:

```
myquey["search_after"] = [score, scroll_id]
myquey["sort"] = [{"_score": "desc", "_id": "desc"}]
```

New Query for next page becomes

In [ ]:

```
new_query ={
   "_source":[

   ],
   "size":10,
   "query":{
      "bool":{
         "must":[

         ],
         "filter":[
            {
               "exists":{
                  "field":"director"
               }
            }
         ],
         "should":[
            {
               "match_phrase":{
                  "director":"Richard "
               }
            }
         ],
         "must_not":[

         ]
      }
   },
   "search_after":[
      "0.0",
      "8URc93IB135PBBnB55dH"
   ],
   "sort":[
      {
         "_score":"desc",
         "_id":"desc"
      }
   ]
}
```

Now perfrom the search on Elastic search and you will get the result

- make sure again create a scroll and then next page keep reperating the process

Please Dont Forget to Like and Share the Article if found useful

at June 30, 2020 (2020-06-30T16:17:00-07:00) (https://soumilshah1995.blogspot.com/2020/06/elk-pre-margin-0px-border-none-padding.html)

No comments:

Post a Comment

(https://www.blogger.com/comment-iframe.g?blogID=2397361725226431430&postID=3373859464234133983&blogspotRpcToken=7799317)

Enter your comment...

Comment as: hemanth22he

Publish Preview

Newer Post (https://soumilshah1995.blogspot.com/2020/07/computing-similarity-on-images-using.html) Home (https://soumilshah1995.blogs
Older Post (https://soumilshah1995.blogspot.com/2020/05/gensim-word2vec-visua

Subscribe to: Post Comments (Atom) (https://soumilshah1995.blogspot.com/feeds/3373859464234133983/comments/default)

Power of Semantics Search combined with Elastic Search | ML on ELK (https://soumilshah1995.blogspot.com/2022/01/power-of-semantics-search-combined-with.html)

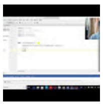
master Power of Semantics Search combined with Elastic Search | ML on ELK ¶ Soumil ...

Project: Data Analysis and Visualizations and Predicting Future Energy Consumption using LSTM Predicting Values 2 month Later Accurately RNN (https://soumilshah1995.blogspo data-analysis-and.html)
Energy Hourly Energy Consumption ¶ Step 1: ¶ Import Library ¶ I...

(https://soumilshah1995.blogspot.com/2019/04/server-and-client-send-actual-sensor.html) Server and Client Send Actual Sensor Data over Network using Raspberry (https://soumilshah1995.blogspot.com/2019/04/server-and-client-send-actua
Lab 3 Server and Client Send Actual Sensor Data over Network using Rasp

(https://soumilshah1995.blogspot.com/2019/05/smart-proxy-library-to-get-random-proxy.html) Smart Proxy library to get random proxy using Python [Hide your Identity (https://soumilshah1995.blogspot.com/2019/05/smart-proxy-library-to-ge
Smart Library that's fetch Random Proxy using Python Smart Proxy libra

Name Entity Recognition on PDF Resume using NLP and spacy python (https://soumilshah1995.blogspot.com/2020/05/name-entity-recognition-on-pdf-resume.html)
NamedEntity Name Entity Recognition on PDF Resume using NLP and spacy ¶ In [22...


(https://soumilshah1995.blogspot.com/2019/04/upload-any-sensor-data-to-thingspeak.html) Upload any Sensor data to ThingSpeak using Raspberry/Arduino Python (Examples (https://soumilshah1995.blogspot.com/2019/04/upload-any-sen thingspeak.html)
Lab 4 (ThingSpeak) Getting started with Open Source Cloud Server Uploa

Getting started with Elastic Search and Python (https://soumilshah1995.blogspot.com/2020/01/getting-started-with-elastic-search-and.html)
Getting started with Elastic Search Getting started with Elastic Search and Python ¶...

4 Ways to do Pagination or scrolling in Elastic Search Tutorials (https://soumilshah1995.blogspot.com/2020/06/elk-pre-margin-0px-border-none-padding.html)
ELK Elastic Search Tutorials ¶ 4 Ways to do Pagination or scrolling in Elastic Searc...

Using BERT with Scikit Learn to do Text classification¶ (https://soumilshah1995.blogspot.com/2021/04/using-bert-with-scikit-learn-to-do-text.html)
BERT Using BERT with Scikit Learn to do Text classification ¶ Soumil Nitin Shah ¶ Ba...

KNN Machine learning Algorithm on ElasticSearch (https://soumilshah1995.blogspot.com/2020/05/knn-machine-learning-algorithm-on.html)
Untitled KNN Machine learning Algorithm on ElasticSearch ¶ Step 1 ¶ Import th...

Entity Recognition Extract information from Job posting using Spacy Machine learning Model (https://soumilshah1995.blogspot.com/2021/04/entity-recognition-extract-information.ht
Untitled Entity Recognition Extract information from Job posting ¶ Soumil Nitin Sha...

Followers

Followers (17)

Follow

Contact Form

Name

Email *

Message *

Send