



Національний технічний університет України  
«Київський політехнічний інститут»  
Фізико-Технічний Інститут  
Кафедра математичних методів захисту інформації

## **Аналітика Big Data**

### **Практичне заняття №3**

**Встановлення системи Elasticsearch. Завантаження інформації  
ElasticSearch в базу даних із JSON-файлу.**

Виконав:  
студент групи ФІ-73  
Чіхладзе Вахтанг

Перевірив:  
Ланде Д.В.

Київ 2021

## Завдання на самостійну роботу

- 1)Написати програму формування пакетного файлу для завантаження в Elasticsearch на основі сформованого на попередньому занятті файлу в форматі JSON.
- 2)Встановити на комп'ютері утиліту curl (<http://curl.se>) і вивчити склад і значення її основних параметрів.
- 3)Ознайомитися зі складом і змістом CRUD-операцій в Elasticsearch.

- 1)З другої практичної роботи в нас з'явився файл, який містить в собі item'и в json форматі. Функція `prepare_items()` зчитує з файлу всі item'и та робить масив з item'ів.
- 2)Дана утиліта вже була встановлена у моїй операційній системі Linux. Значення її параметрів можна побачити в map сторінці.
- 3)В Elasticsearch є операції CreateReadUpdateDelete (що і позначає CRUD). Дані створюються флагом `-X PUT`, зчитуються `-X GET`, обновлюються `-X POST`, видаляються `-X DELETE`.

Код проекту доступен на github: <https://github.com/Pozzitrion1337/BigData-Labs>

```
import os
import subprocess
import time
import re

def prepare_items(filepath):
    file=open(filepath,'r')
    text=file.read()
    items=text.split('}',\n')
    for i in range(len(items)):
        items[i]=items[i]+'}'
    return items

def start_elastic():
    logs=open("logs.txt","w")
    elastic=subprocess.Popen(["../elasticsearch-7.11.1/bin/elasticsearch"],stdout=logs)
    return elastic

def stop_elastic(elastic):
    elastic.kill()
    time.sleep(5)

def check_elastic_server():
    out=os.popen('curl -X GET http://localhost:9200/').read()
    while out.find('Failed to connect') > -1:
        print("Connection failed,trying again...")
        time.sleep(1)
        out=os.popen('curl -X GET http://localhost:9200/').read()
        print(out)

def init_all_items(filepath):
```

```

#CRUD Operation: update
items=prepare_items(filepath)
for item in items:
    command="curl -X POST 'http://localhost:9200/test/_doc/' -H 'Content-Type:
application/json' -d '"+item+"'"
    os.system(command)

def get_items(url):
    command="curl -X GET "+url;
    os.system(command)

def delete_item(url):
    command="curl -X DELETE "+url;
    os.system(command)

if __name__ == "__main__":
    menu="""0.exit
1.start elastic
2.check elastic
3.stop elastic
4.write rss_feeds.json
""";
    elastic=None
    while True:
        print(menu)
        chose=int(input("Enter your choose: "))
        if chose == 0:
            exit()
        if chose == 1:
            elastic = start_elastic()
        if chose == 2:
            check_elastic_server()
        if chose == 3:
            stop_elastic(elastic)
        if chose == 4:
            init_all_items('rss_feeds.json')

```

Висновок: отже, я познайомився з деяким інтерфесом для роботи з інструментом зберігання та пошуку даних elasticsearch. Також було реалізована програма для взаємодії з сервером та було поглиблено знання з автоматизації та системного програмування.