

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ІМЕНІ ІГОРЯ  
СІКОРСЬКОГО»

Факультет прикладної математики  
Кафедра програмного забезпечення комп'ютерних систем

**ЗВІТ**  
з лабораторної роботи №3  
«Пошук документів за метаданими за  
допомогою Elasticsearch»

**Виконав:**

студент 4-го курсу, групи КП-91,  
спеціальності 121 – Інженерія  
програмного забезпечення  
*Власюк Сергій Петрович*

**Перевірив:**

ас. каф. ПЗКС  
*Юсин Яків Олексійович*

Київ 2023

## Постановка задачі за варіантом

Доповнити інформаційно-пошукову систему, реалізовану в рамках другої лабораторної роботи, можливістю повнотекстового пошуку. Робота з розробленим програмним забезпеченням повинне задовільнятися наступними вимогами:

- Для реалізації програмного забезпечення необхідно використати як основу розроблену інформаційно-пошукову систему з другої лабораторної роботи.
- Структуру документа, визначену у ході другої лабораторної роботи, необхідно доповнити декількома полями типу `text`, по яким буде виконуватись повнотекстовий пошук.
- Додані поля повинні відповідати предметній галузі, визначеній за варіантом студента у другій лабораторній роботі.
- Одне із доданих полів повинне використовувати стандартний аналізатор `Elasticsearch`.
- Одне із доданих полів повинне використовувати вбудований аналізатор `Elasticsearch`, відповідний до мови тексту, що зберігається в цьому полі. Наприклад, якщо там зберігається текст англійською мовою, то це поле має використовувати вбудований аналізатор для англійської мови.
- Одне із доданих полів повинне використовувати користувацький аналізатор, створений студентом. Фільтри символів, токенизатор та фільтри токенів для створення цього аналізатора обираються довільно, за бажанням студента, але потрібно обов'язково використати хоча б один фільтр символів та хоча б один фільтр токенів.
- Якщо для визначеної предметної галузі неможливо визначити три різних поля для повнотекстового пошуку, допускається визначення двох різних полів + дубля одного із них.
- Користувач повинен мати можливість виконувати повнотекстовий пошук документів за кожним із доданих полів.
- Обов'язковим для реалізації є повнотекстовий запит `match`. За бажанням студента, можна реалізувати інші типи запитів.

### 1)Реалізація програми

```
const elasticsearch = require('elasticsearch');
```

```
const readline = require('readline');
```

```
const client = new elasticsearch.Client({  
  host: 'localhost:9200',  
});
```

```
const indexName = 'games1';
```

```
const rl = readline.createInterface({  
  input: process.stdin,  
  output: process.stdout,  
});
```

```
const indexMapping = {  
  properties: {  
    name: { type: 'keyword' },  
    main_character: { type: 'keyword' },  
    date: { type: 'date' },  
    developers: { type: 'keyword' },  
    description: { type: 'text', analyzer: 'standard' },  
    plot: { type: 'text', analyzer: 'english' },  
    response: { type: 'text', analyzer: 'custom_analyzer' },  
  },  
};
```

```
function createDocument(callback) {  
  rl.question('Name: ', (name) => {  
    rl.question('Main_character: ', (main_character) => {  
      rl.question('Publication Date (YYYY-MM-DD): ', (date) => {  
        rl.question('Developers: ', (developers) => {  
          rl.question('Description: ', (description) => {  
            rl.question('Plot: ', (plot) => {  
              rl.question('Response: ', (response) => {  
                client  
                  .index({  
                    index: indexName,
```

```

        body: {
          name,
          main_character,
          date,
          developers: developers
            .split(',')
            .map((keyword) => keyword.trim()),
          description,
          plot,
          response,
        },
      })
    .then((response) => {
      console.log('Document created:', response);
      callback();
    })
    .catch((error) => {
      console.error('Error creating document:', error);
      callback();
    });
  });
});
});
});
});
});
});
});
});
});
});

function searchDocuments(callback) {
  rl.question('Search by (name, main_character, date, developers): ', (field) => {
    rl.question('Search value: ', (value) => {
      const filter =
        field === 'date' ? 'range' : 'term';
      const query =
        filter === 'range'

```

```

    ? { range: { [field]: { gte: value } } }
    : { term: { [field]: value } };
    client
    .search({
      index: indexName,
      body: {
        query: {
          bool: {
            filter: [query],
          },
        },
      },
    })
    .then((response) => {
      console.log(
        'Search results:',
        JSON.stringify(response.hits.hits, null, 4)
      );
      callback();
    })
    .catch((error) => {
      console.error('Error searching documents:', error);
      callback();
    });
  });
}

```

```

function deleteDocument(callback) {
  rl.question('Document ID to delete: ', (id) => {
    client
      .delete({
        index: indexName,
        id,
      })
      .then((response) => {

```

```

        console.log('Document deleted:', response);
        callback();
    })
    .catch((error) => {
        console.error('Error deleting document:', error);
        callback();
    });
});
}

```

```

function fuzzinessSearch(callback) {
    rl.question('Search by (name, main_character): ', (field) => {
        rl.question('Search value (fuzzy syntax): ', (value) => {
            client
                .search({
                    index: indexName,
                    body: {
                        query: {
                            fuzzy: {
                                [field]: {
                                    value: value,
                                    fuzziness: 2
                                }
                            },
                        },
                    },
                })
                .then((response) => {
                    console.log('Search results:', JSON.stringify(response.hits.hits, null, 4));
                    callback();
                })
                .catch((error) => {
                    console.error('Error searching documents:', error);
                    callback();
                });
        });
    });
}

```

```
});  
}
```

```
function getAllDocuments(callback) {  
  client  
    .search({  
      index: indexName,  
      body: {  
        query: {  
          match_all: {},  
        },  
      },  
    })  
    .then((response) => {  
      console.log(  
        'All documents:',  
        JSON.stringify(response.hits.hits, null, 4)  
      );  
      callback();  
    })  
    .catch((error) => {  
      console.error('Error getting all documents:', error);  
      callback();  
    });  
}
```

```
function searchFullText(callback) {  
  rl.question(  
    'Search by (description, plot, response): ',  
    (field) => {  
      rl.question('Search value: ', (value) => {  
        client  
          .search({  
            index: indexName,  
            body: {  
              query: {
```

```

match: {
  [field]: value,
},
},
},
}))
.then((response) => {
  console.log(
    'Search results:',
    JSON.stringify(response.hits.hits, null, 4)
  );
  callback();
})
.catch((error) => {
  console.error('Error searching documents:', error);
  callback();
});
});
}
);
}

```

```

function mainMenu() {
  rl.question('Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): ', (action) => {
    if (action === 'create' || action === '1') {
      createDocument(mainMenu);
    } else if (action === 'search' || action === '2') {
      searchDocuments(mainMenu);
    } else if (action === 'delete' || action === '3') {
      deleteDocument(mainMenu);
    } else if (action === 'fuzzy' || action === '4') {
      fuzzinessSearch(mainMenu);
    } else if (action === 'get all' || action === '5') {
      getAllDocuments(mainMenu);
    } else if (action === 'full text search' || action === '6') {
      searchFullText(mainMenu);
    }
  });
}

```



```

    } else {
      console.error('Invalid action');
      rl.close();
    }
  });
}

```

client.indices

```

.exists({ index: indexName })
.then((exists) => {
  if (!exists) {
    createIndex();
  } else {
    mainMenu();
  }
})
.catch((error) => {
  console.error('Error checking index existence:', error);
});

```

function createIndex() {

```

  client.indices
    .create({
      index: indexName,
      body: {
        settings: {
          analysis: {
            analyzer: {
              custom_analyzer: {
                type: 'custom',
                tokenizer: 'standard',
                char_filter: ['html_strip'],
                filter: ['lowercase', 'asciifolding'],
              },
            },
          },
        },
      },
    },
  ),
}

```

```
    },  
    },  
    mappings: indexMapping,  
  },  
  })  
  .then((response) => {  
    console.log('Index created:', response);  
    mainMenu();  
  })  
  .catch((error) => {  
    console.error('Error creating index:', error);  
  });  
}
```

## **Результати виконання роботи**

```
PS C:\Users\Haylevel_SV\Desktop\search\lab3> node .\1.js
Index created: { acknowledged: true, shards_acknowledged: true, index: 'games1' }
Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): 5
All documents: []
Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): 1
Name: Stalker
Main_character: Strelok
Publication Date (YYYY-MM-DD): 2008-01-12
Developers: Anton, Serhii
Description: First PerSoN shooter
Plot: War with mutants
Response: <p>Good game</p>
Document created: {
  _index: 'games1',
  _id: 'ZDT9WIcBsaxPrq1ONdES',
  _version: 1,
  result: 'created',
  _shards: { total: 2, successful: 1, failed: 0 },
  _seq_no: 0,
  _primary_term: 1
}
Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): 5
All documents: [
  {
    "_index": "games1",
    "_id": "ZDT9WIcBsaxPrq1ONdES",
    "_score": 1,
    "_source": {
      "name": "Stalker",
      "main_character": "Strelok",
      "date": "2008-01-12",
      "developers": [
        "Anton",
        "Serhii"
      ],
      "description": "First PerSoN shooter",
      "plot": "War with mutants",
      "response": "<p>Good game</p>"
    }
  }
]
Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): 6
Search by (description, plot, response): description
Search value: person
Search results: [
  {
    "_index": "games1",
    "_id": "ZDT9WIcBsaxPrq1ONdES",
    "_score": 0.2876821,
    "_source": {
      "name": "Stalker",
```

```

        "main_character": "Strellok",
        "date": "2008-01-12",
        "developers": [
            "Anton",
            "Serhii"
        ],
        "description": "First PerSoN shooter",
        "plot": "War with mutants",
        "response": "<p>Good game</p>"
    }
}

]
Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): 6
Search by (description, plot, response): description
Search value: per
Search results: []
Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): 6
Search by (description, plot, response): plot
Search value: war
Search results: [
    {
        "_index": "games1",
        "_id": "ZDT9WIcBsaxPrq1ONdES",
        "_score": 0.2876821,
        "_source": {
            "name": "Stalker",
            "main_character": "Strellok",
            "date": "2008-01-12",
            "developers": [
                "Anton",
                "Serhii"
            ],
            "description": "First PerSoN shooter",
            "plot": "War with mutants",
            "response": "<p>Good game</p>"
        }
    }
]
Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): 6
Search by (description, plot, response): response
Search value: Good game
Search results: [
    {
        "_index": "games1",
        "_id": "ZDT9WIcBsaxPrq1ONdES",
        "_score": 0.5753642,
        "_source": {
            "name": "Stalker",
            "main_character": "Strellok",
            "date": "2008-01-12",
            "developers": [

```

```

        "developers": [
            "Anton",
            "Serhii"
        ],
        "description": "First PerSoN shooter",
        "plot": "War with mutants",
        "response": "<p>Good game</p>"
    }
}
]
Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): 6
Search by (description, plot, response): response
Search value: Good
Search results: [
    {
        "_index": "games1",
        "_id": "ZDT9WIcBsaXPrq1ONdES",
        "_score": 0.2876821,
        "_source": {
            "name": "Stalker",
            "main_character": "Strelok",
            "date": "2008-01-12",
            "developers": [
                "Anton",
                "Serhii"
            ],
            "description": "First PerSoN shooter",
            "plot": "War with mutants",
            "response": "<p>Good game</p>"
        }
    }
]
Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): 6
Search by (description, plot, response): response
Search value: <p>
Search results: []
Choose action (1-create, 2-search, 3-delete, 4-fuzzy, 5-all, 6-full text): 6
Search by (description, plot, response): response
Search value: good
Search results: [
    {
        "_index": "games1",
        "_id": "ZDT9WIcBsaXPrq1ONdES",
        "_score": 0.2876821,
        "_source": {
            "name": "Stalker",
            "main_character": "Strelok",
            "date": "2008-01-12",
            "developers": [
                "Anton",
                "Serhii"
            ],
            "description": "First PerSoN shooter",
            "plot": "War with mutants",
            "response": "<p>Good game</p>"
        }
    }
]

```

## **Висновки**

В ході виконання даної лабораторної роботи, я ознайомився з повнотекстовим пошуком за допомогою Elasticsearch. Розробка програмного забезпечення виконувалась з використанням мови програмування JavaScript. Було доповнено інформаційно-пошукову систему, реалізовану в рамках другої лабораторної роботи, можливістю повнотекстового пошуку.