Використання Apache Solr для реалізації повнотекстового пошуку у комерційних системах

Виконав
Студент КН-1 МП
Кривонос А.А.

Керівник:
Яремко С.А.

Актуальність і мета курсової роботи

З розвитком технологій виникає необхідність у навігації по величезних обсягах даних. Звичайний інформаційний пошук засобами СКБД не завжди задовольняє всі потреби, зокрема у швидкості та гнучкості. Саме тому існує ряд спеціалізованих рішень, таких як Apache Solr, ElasticSearch, Lunrjs, Apache Nutch тощо.

Метою курсової роботи є дослідження теми повнотекстового пошуку і платформи Apache Solr, а також її застосування для реалізації пошуку у комерційних системах, зокрема у музичних стримінгових сервісах.

Коротко про розділи курсової роботи

- Розділ 1. Аналіз предметної області
  - Визначення повнотекстового пошуку
  - Бібліотека Apache Lucene
  - Загальні відомості про Apache Solr
  - Опис альтернативної платформи ElasticSearch та її порівняння з Apache Solr
- Розділ 2. Огляд інструментів розробки
  - Детальний розбір понять Apache Solr
  - Опис інших технологій (мови Python та використаних бібліотек)
- Розділ 3. Реалізація системи
  - Опис вхідних даних та схеми
  - Функціональні особливості системи
  - Архітектура
  - Тестування та опис можливостей для покращення

Apache Solr

Apache Solr - це open-source пошукова платформа. Її розробка розпочалась у 2004 році. Заснована на бібліотеці Apache Lucene.

Серед основних можливостей Solr є повнотекстовий та фасетний пошук, індексування у реальному часі, підтримка кластеризації та масштабування, та зручна інтеграція з базами даних.

Одним із найбільших конкурентів є ElasticSearch, який також заснований на Apache Lucene.

Інструменти розробки

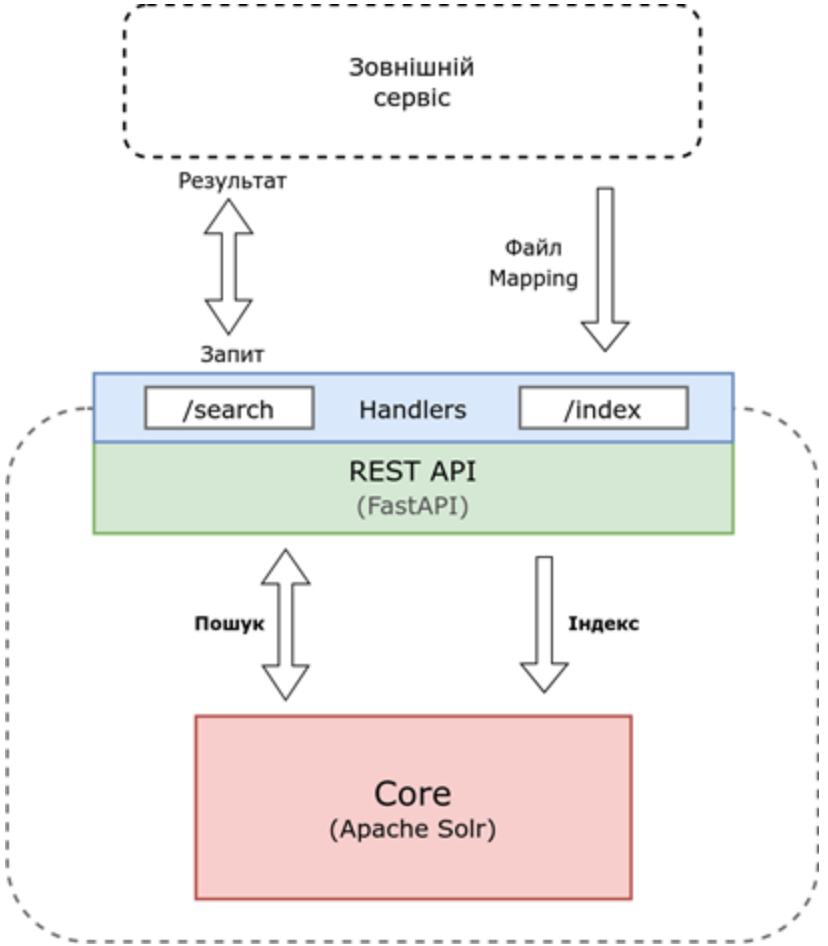
- Apache Solr
- Python
  - FastAPI
  - pysolr
  - requests

Архітектура системи

Ядром системи виступає Apache Solr

Для взаємодії зовнішніх сервісів імплементовано REST API, який має два основних ендпоінти:

- /search - для пошуку
- /index - для оновлення даних



Основні функціональні особливості

- Фільтрація зайвих даних, які подаються на індексацію
- Вилучення дублікатів
- Один єдиний ендпоінт для пошукових запитів по полям:
  - Назва пісні
  - Виконавець
  - Текст пісні
- Нечіткий пошук (часткове нівелювання допущення помилок у запиті)
- Маппінг полів (можливість інтеграції з іншими сховищами даних, де поля мають інші назви)

Тестування

Тестування було проведено для різних юзкейсів:

- Коли користувач знає точну назву виконавця та пісні
- Коли користувач знає назву виконавця та декілька слів з пісні
- Коли користувач знає приблизні слова з пісні
- Коли користувач вводить запит з помилками

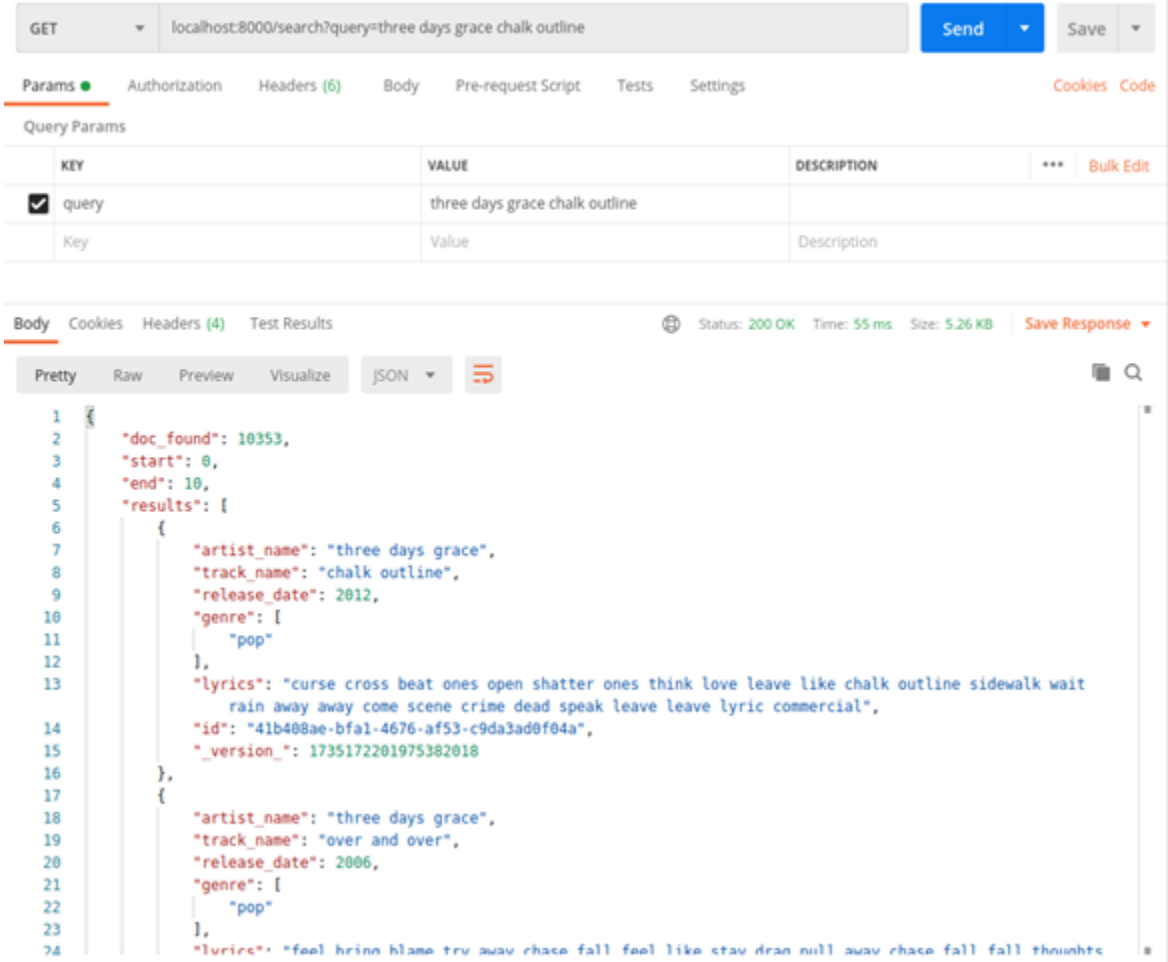
Інструмент для тестування: **Postman**

Тестування

Точна назва пісні і виконавця:

“Three days grace chalk outline”

(Three Days Grace – Chalk Outline)

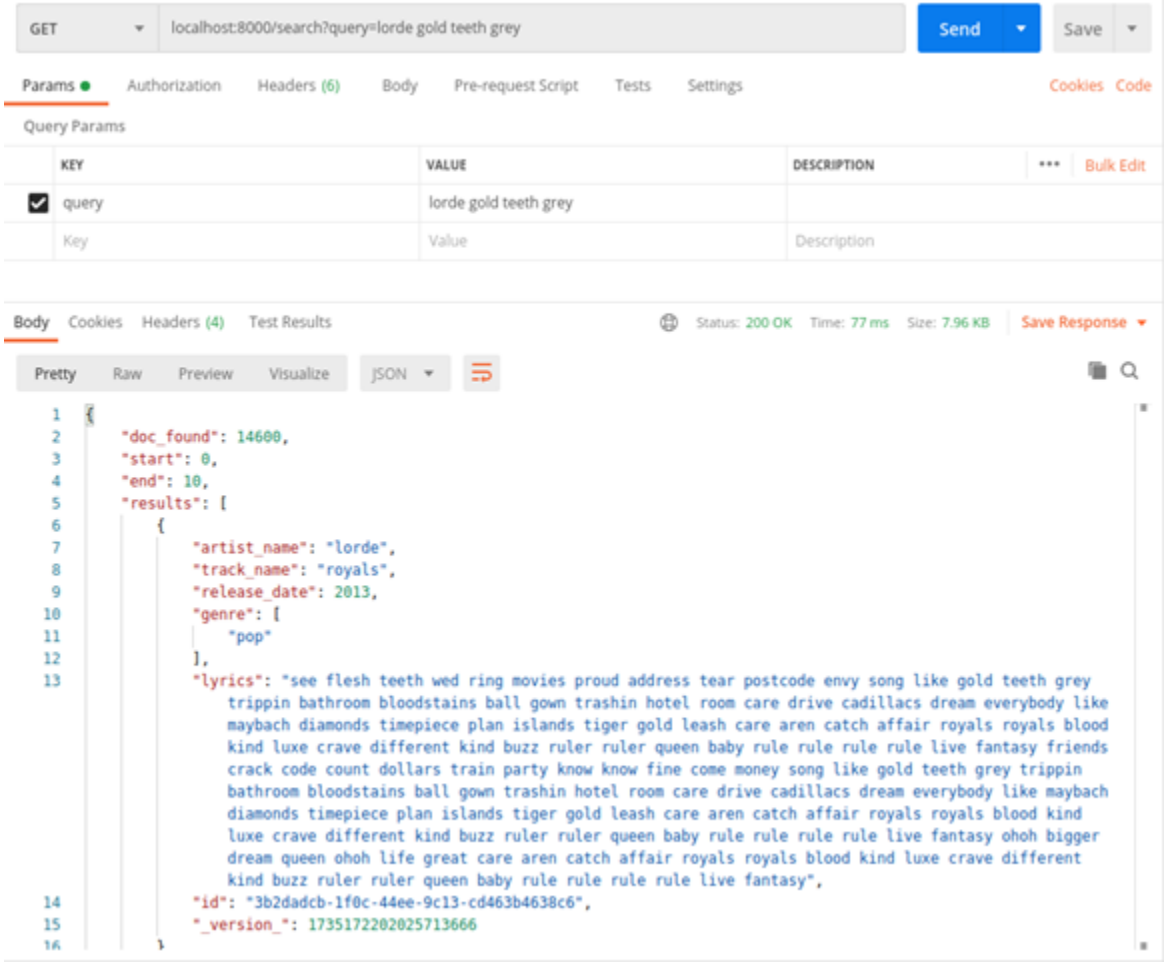


Тестування

Назва виконавця і кілька слів з пісні:

“lorde gold teeth grey trippin”

(Lorde – Royals)

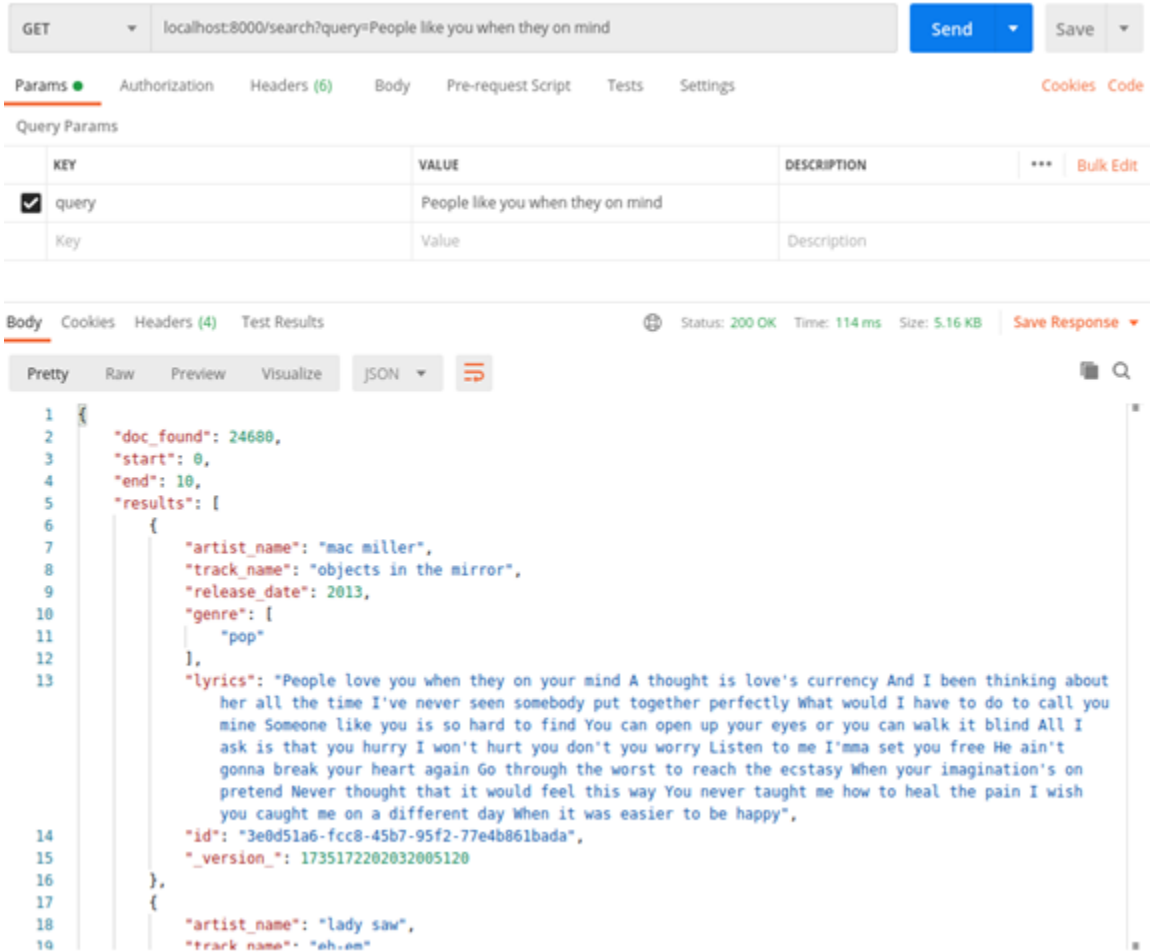


Тестування

Приблизні слова з пісні:

“People like you when they on the mind”

(Mac Miller – Objects in the Mirror)

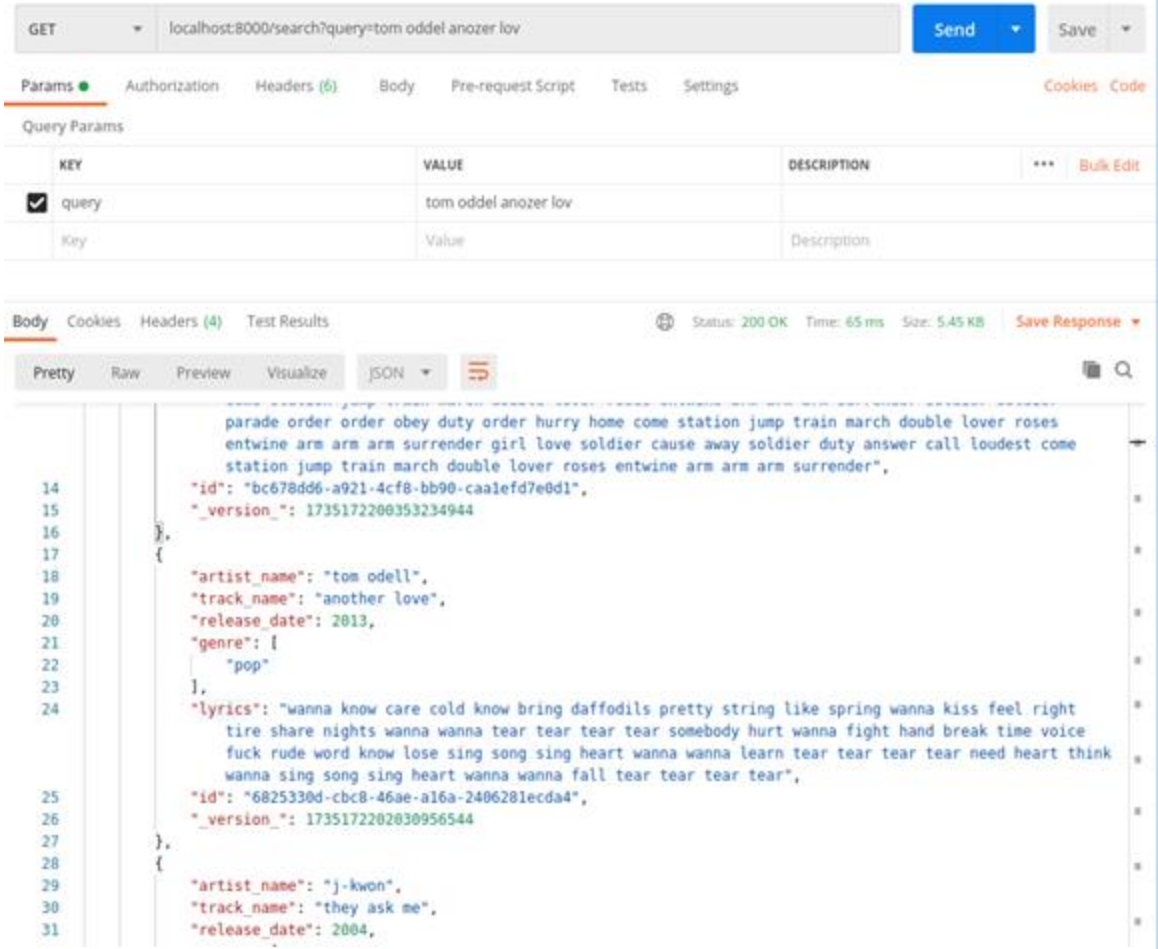


Тестування

Запит з помилками:

“tom oddel anozer lov“

(Tom Odell – Another Love)



Можливості для покращення системи

- Додати синхронізацію з зовнішніми БД
- Зробити інтеграцію з рекомендаційними системами (додавати динамічні ваги для улюблених жанрів та виконавців)
- Додавати фасетинг для підбивання статистики по окремим жанрам
- Реалізувати підтримку додаткових полів (наприклад, параметрів звуку)

Висновки

- Було досліджено тему повнотекстового пошуку та платформу Apache Solr
- Реалізовано систему, яка виконує пошук по музичній колекції
- Перевагами Apache Solr є швидкість пошуку, велика кількість функціональних можливостей, підтримка кластеризації та можливість динамічної конфігурації через REST API
- Недоліки Apache Solr: складність відлагодження при виникненні помилок, плутанина в документації через різні версії.
- Apache Solr займає свою нішу, але ElasticSearch дедалі більше випереджає її за багатьма показниками