УДК 004.4

ОПТИМІЗАЦІЯ ПРОЦЕСУ ПОШУКУ ТОВАРІВ НА МАРКЕТ ПЛЕЙСАХ НА ОСНОВІ JSOUP

Манжула В.В.[1)], Крепич С.Я.[2)], Шерстюк Ю.Ю.[3)]
Західноукраїнський національний університет
[1),3)] магістрант; [2)]к.т.н., доцент

І. Постановка проблеми

Ефективний пошук елементів на HTML сторінці за допомогою Java залежить від багатьох факторів. Основна проблема полягає в здатності ефективно виявляти та вибирати бажані елементи зі складного структурного дерева HTML. Це означає розуміння специфіки HTML-коду та здатність адаптувати алгоритм до різних форматів сторінок та їхньої різноманітності.

Однією з ключових задач є ефективне використання jsoup для навігації та вибору елементів. Для досягнення цієї мети потрібно ретельно вивчити документацію бібліотеки та розробити стратегію пошуку, що враховує різноманітність структур HTML. Додатково, виникає завдання оптимізації алгоритму для швидкого та ефективного пошуку на великих обсягах даних.

ІІ. Мета роботи

Метою даної роботи є розробка алгоритму пошуку елементів на HTML сторінці з використанням мови програмування Java та бібліотеки jsoup. Головними завданнями є створення ефективного та надійного алгоритму, здатного вибирати бажані елементи з будь-якої HTML-структури, а також оптимізація процесу пошуку для забезпечення швидкості та ефективності на великих обсягах даних. Ця робота спрямована на вдосконалення інструментів для обробки та аналізу веб-контенту, що є важливим у контексті розвитку сучасних інтернет-технологій та веб-аналітики.

ІІІ. Проектування алгоритму пошуку елементів

Jsoup– це бібліотека для обробки HTML та XML даних в Java. Вона надає зручні методи для отримання, зміни та видалення елементів на HTML сторінці. Ефективні методи для пошуку елементів на HTML сторінці за допомогою Jsoup забезпечує використання таких засобів:

- CSS-селекторів.Jsoup дозволяє використовувати CSS-селектори для вибору елементів. Надає можливість використовувати метод select(), передавши до нього CSS-селектор, щоб отримати список знайдених елементів;
- ID або класів. Можливість шукати елементи за їх ID або класами, використовуючи CSS-селектори;
- методів getElementsBy...().Jsoup також має набір методів getElementsBy...(), які дозволяють вам шукати елементи за їх тегом, класом, ID тощо;
- Пошук елементів за атрибутами;
- Поєднання кількох методів пошуку. Для складних запитів можна поєднувати кілька методів пошуку, наприклад, спочатку вибрати всі елементи з певним тегом, а потім вже серед них вибирати тільки ті, що мають певний клас.

Дані засоби Jsoup надають можливість здійснювати ефективний пошук елементів на HTML сторінці та виконувати різноманітні маніпуляції з ними.

Схематично, реалізацію алгоритму пошуку елементів можна зобразити як на рисунку 1.

Отримання HTML-коду сторінки. Цей етап включає в себе взаємодію з веб-сервером або файловою системою для отримання HTML-коду веб-сторінки. При взаємодії з веб-сервером ми виконуємо HTTP-запит до веб-сторінки і отримуємо відповідь, яка містить HTML-код сторінки. При роботі з локальними файлами ми читаємо збережений HTML-файл з файлової системи.

Створення об'єкту Document з використанням jsoup. Після отримання HTML-коду ми використовуємо бібліотеку jsoup для його парсингу та створення об'єкту типу Document. Цей об'єкт представляє собою дерево DOM сторінки і дозволяє легко навігуватися по її структурі та взаємодіяти з елементами.

Jsoup.connect("http://example.com").get();

*Використання селекторів для пошуку елементів.*Jsoup дозволяє використовувати CSS-подібні селектори для пошуку конкретних елементів на сторінці. Селектори можуть бути використані для вибору елементів за їх тегами, класами, ідентифікаторами, атрибутами тощо. Наприклад, select("div.content") вибирає всі елементи з класом "content".

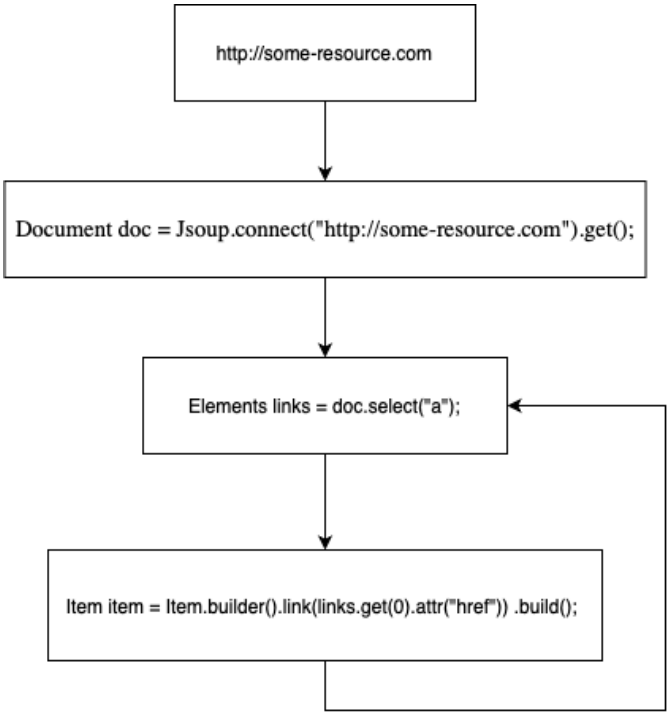

Рисунок 1 – Алгоритм пошуку елементів на сторінці

Обробка результатів пошуку. Після виконання пошуку ми отримуємо колекцію елементів, які задовольняють наш селектор. Ми можемо ітерувати цю колекцію та виконувати різні операції з кожним елементом, такі як отримання текстового вмісту, значень атрибутів, додавання класів, зміна стилів тощо. Згідно з цілями нашого алгоритму, додаємо текстовий вміст елементів або значення їхніх атрибутів до моделі Item.

```
Elementslinks = doc.select("a");
for (Elementlink : links) {
        Itemitem = Item.builder().
                .link(link.attr("href"))
                .build();        }
```

Висновок

У даній роботі розроблено алгоритм пошуку товарів на українських маркет плейсах засобами Jsoup, яке дозволяє здійснювати швидкий та глобальний пошук товарів. Ефективність запропонованого алгоритму обумовлена перевагами використання Jsoup. Зокрема,простий та зрозумілий API, що робить роботу з HTML даними легкою та зручною для розробників; ефективний пошук елементів,завдяки механізму використання CSS-селекторів та методів для отримання елементів за їх тегами, класами, ID та атрибутами; підтримка обробки складних HTML документів; підтримка функцій парсингу та обробки HTML і XML. Загалом, використання Jsoup є ефективним способом роботи з HTML даними в Java, що дозволяє швидко та зручно вибирати, змінювати та аналізувати вміст веб-сторінок.

Список використаних джерел

1. Lanet [Електронний ресурс]. — Режим доступу: https://lanet.click/ppc/reklama-v-prais-ahrehatorakh (2022).
2. JSoup[Електронний ресурс]. — Режим доступу: https://jsoup.org (2022).
3. uk.wikipedia.org – Вільна енциклопедія [Електронний ресурс]. – Режим доступу: http://uk.wikipedia.org для доступу до інформаційних ресурсів авторизація не потрібна — Назва з екрану. Вікіпедія. Вільна енциклопедія.