**Pre-Requisites:**

|  |  |
| --- | --- |
| Java | jdk1.8.0\_101 |
| Maven | apache-maven-3.3.9 |
| Elastic Search | elasticsearch-6.1.2  **Configuration:**  host: localhost  Port: 9200  Transport: http |
| Eclipse | Neon |

How to setup using source code:

1. Checkout source code and import as maven project to eclipse
2. Build the project using by running “mvn clean package”

Run the application:

Option 1: Build the executable jar from the source code. Instructions are listed above.

Option 2: Checkout executable jar from git located in target folder.

Usage

|  |  |  |
| --- | --- | --- |
| **Command line argument** | **Example** | **Notes** |
| 1 | java -jar SearchScaper-RestFul-0.0.1-SNAPSHOT-spring-boot.jar **1** | 1) Delete exiting index with name “searchtasks”  2) Create new index with name “searchtasks”  3) Add one document with “searchtask” type  {  "taskName": "Olympics",  "keyWords": "\"red gerard\",\"mikaela shiffrin\"",  "active": true,  "createdAt": "2018-02-15"  } |
| 2 | java -jar SearchScaper-RestFul-0.0.1-SNAPSHOT-spring-boot.jar **2** | 1) Delete exiting index with name “searchresults”  2) Create new index with name “searchresult” |
| No option | java -jar SearchScaper-RestFul-0.0.1-SNAPSHOT-spring-boot.jar | 1) Retrieve active searchtask documents  2) For each document, parse keyword and scrape google searching for each keyword  3) Add searchresult document by getting contents of the url’s returned by google search for each key word |

Running from Eclipse:

1. Run elasticsearch.searchscraper.ApplicationMain is main class in either Run mode or debug mode passing in the command line option.

Running from command line:

1. Make sure that JDK is in classpath.
2. Execute command “java -jar SearchScaper-RestFul-0.0.1-SNAPSHOT-spring-boot.jar” passing in the correct command line arguments if you want to create index and setup sample data.

Areas of Improvement:

1. Unit test cases
2. Implement multi threading
3. Comprehensive Error handling
4. Random polling