**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 |
| Kafka | kafka\_2.11-1.0.0  Commands to start kafka and create topics used by this application.  **Start Zookeeper:**  *zookeeper-server-start.bat ../../config/zookeeper.properties*  **Start Kafka:**  *kafka-server-start.bat ../../config/server.properties*  **Create Topics:**  *kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic searchTaskQueue*  *kafka-topics.bat --create --zookeeper localhost:2181 --replication-factor 1 --partitions 1 --topic searchResultQueue* |
| 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-kafka-elastic-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"  }  4) Scheduler will run the processing of searchresults |
| 2 | java -jar SearchScaper-RestFul-kafka-elastic-0.0.1-SNAPSHOT-spring-boot.jar **2** | 1) Delete exiting index with name “searchresults”  2) Create new index with name “searchresult”  3) Scheduler will run the processing of searchresults |
| No option | java -jar SearchScaper-RestFul-kafka-elastic-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 spring.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-kafka-elastic-0.0.1-SNAPSHOT-spring-boot.jar” passing in the correct command line arguments if you want to create index and setup sample data for elastic search.

**How it works:**

1. Application queries the elastic search for the search tasks.
2. Looping through each search task, it pushes search task onto “searchTaskQueue”.
3. Search Task Consumer of the “searchTaskQueue” consumes search task and search the google for each key word from the search task (Example: referring to above document, it searches for keyword “red gerard” and “mikaela Shiffrin”)
4. For each link returned for the keyword, Search Task Consumer puts search result onto “searchResultQueue”.
5. Search Result Consumer consumes the request and gets the contents of the URL and stored into the elastic search.