Pom and main class

<project xmlns=*"http://maven.apache.org/POM/4.0.0"* xmlns:xsi=*"http://www.w3.org/2001/XMLSchema-instance"* xsi:schemaLocation=*"http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd"*>

<modelVersion>4.0.0</modelVersion>

<groupId>elasticsearch</groupId>

<artifactId>elasticsearch</artifactId>

<version>0.0.1-SNAPSHOT</version>

<dependencies>

<dependency>

<groupId>org.elasticsearch.client</groupId>

<artifactId>elasticsearch-rest-high-level-client</artifactId>

<version>7.16.3</version>

</dependency>

<dependency>

<groupId>org.elasticsearch.client</groupId>

<artifactId>elasticsearch-rest-client</artifactId>

<version>7.16.3</version>

</dependency>

<dependency>

<groupId>org.elasticsearch</groupId>

<artifactId>elasticsearch</artifactId>

<version>7.16.3</version><!--$NO-MVN-MAN-VER$-->

</dependency>

<dependency>

<groupId>com.fasterxml.jackson.core</groupId>

<artifactId>jackson-databind</artifactId>

<version>2.11.1</version>

</dependency>

</dependencies>

</project>

import java.io.IOException;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import java.util.Scanner;

import org.apache.http.HttpHost;

import org.elasticsearch.action.delete.DeleteRequest;

import org.elasticsearch.action.delete.DeleteResponse;

import org.elasticsearch.action.index.IndexRequest;

import org.elasticsearch.action.index.IndexResponse;

import org.elasticsearch.action.search.SearchRequest;

import org.elasticsearch.action.search.SearchResponse;

import org.elasticsearch.action.update.UpdateRequest;

import org.elasticsearch.action.update.UpdateResponse;

import org.elasticsearch.client.RequestOptions;

import org.elasticsearch.client.RestClient;

import org.elasticsearch.client.RestHighLevelClient;

import org.elasticsearch.index.query.QueryBuilders;

import org.elasticsearch.search.SearchHit;

import org.elasticsearch.search.aggregations.AggregationBuilders;

import org.elasticsearch.search.aggregations.Aggregations;

import org.elasticsearch.search.aggregations.bucket.terms.Terms;

import org.elasticsearch.search.builder.SearchSourceBuilder;

public class JavaElasticSearch {

public static void main(String[] args) throws IOException{

Scanner in=new Scanner(System.in);

EmployeeDelete d=new EmployeeDelete();

EmployeeeInsert i=new EmployeeeInsert();

EmployeeFetch f=new EmployeeFetch();

EmployeeUpdate u=new EmployeeUpdate();

System.out.println("Welcome");

System.out.println("1-insert new employee --- 2-Get employee detail --- 3 update employee detail--- 4-delete employee detail");

int choice=in.nextInt();

switch(choice) {

case 1:

i.insertEmp();

break;

case 2:

f.fetchEmp();

break;

case 3:

u.updateEmp();

break;

case 4:

d.deleteEmp();

break;

default:

System.out.println("Enter correct choice:");

}

}

}

Update,delete,insert ,feth:

import java.io.IOException;

import org.apache.http.HttpHost;

import org.elasticsearch.action.delete.DeleteRequest;

import org.elasticsearch.action.delete.DeleteResponse;

import org.elasticsearch.client.RequestOptions;

import org.elasticsearch.client.RestClient;

import org.elasticsearch.client.RestHighLevelClient;

public class EmployeeDelete {

void deleteEmp() throws IOException {

RestHighLevelClient client = new RestHighLevelClient(RestClient.builder(new HttpHost("localhost", 9200, "http")));

DeleteRequest deleteRequest = new DeleteRequest("elasticsearch","12");

DeleteResponse deleteResponse = client.delete(deleteRequest, RequestOptions.DEFAULT);

System.out.println("employee id: "+deleteResponse.getId());

System.out.println("Deletion sucessfull");

}

}

Insert:

import java.io.IOException;

import java.util.HashMap;

import org.apache.http.HttpHost;

import org.elasticsearch.action.index.IndexRequest;

import org.elasticsearch.action.index.IndexResponse;

import org.elasticsearch.client.RequestOptions;

import org.elasticsearch.client.RestClient;

import org.elasticsearch.client.RestHighLevelClient;

public class EmployeeeInsert {

void insertEmp() throws IOException {

RestHighLevelClient client = new RestHighLevelClient(RestClient.builder(new HttpHost("localhost", 9200, "http")));

HashMap<String, Object> map = new HashMap<String, Object>();

map.put("empname","dinesh");

map.put("age",40);

map.put("dept","service");

map.put("desig"," manager");

map.put("salary","10000");

IndexRequest indexRequest = new IndexRequest("elasticsearch");

indexRequest.id("15");

indexRequest.source(map);

IndexResponse indexResponse = client.index(indexRequest, RequestOptions.DEFAULT);

System.out.println("employee id: "+indexResponse.getId());

System.out.println("employee name: "+indexResponse.getResult().name());

}

}

Update:

import java.io.IOException;

import java.util.HashMap;

import java.util.Map;

import org.apache.http.HttpHost;

import org.elasticsearch.action.update.UpdateRequest;

import org.elasticsearch.action.update.UpdateResponse;

import org.elasticsearch.client.RequestOptions;

import org.elasticsearch.client.RestClient;

import org.elasticsearch.client.RestHighLevelClient;

public class EmployeeUpdate {

void updateEmp() throws IOException {

RestHighLevelClient client = new RestHighLevelClient(RestClient.builder(new HttpHost("localhost", 9200, "http")));

Map<String, Object> updateMap = new HashMap<String, Object>();

updateMap.put("empname","abhisek m");

updateMap.put("age",21);

updateMap.put("dept","testing");

updateMap.put("desig","ceo");

updateMap.put("salary","75000");

UpdateRequest request = new UpdateRequest("elasticsearch", "11").doc(updateMap);

UpdateResponse updateResponse= client.update(request, RequestOptions.DEFAULT);

System.out.println("updated employee id: "+updateResponse.getId());

System.out.println("updated sucessfully");

}

}

Fetch:

import java.io.IOException;

import java.util.ArrayList;

import java.util.Arrays;

import java.util.HashMap;

import java.util.List;

import java.util.Map;

import org.apache.http.HttpHost;

import org.elasticsearch.action.search.SearchRequest;

import org.elasticsearch.action.search.SearchResponse;

import org.elasticsearch.client.RequestOptions;

import org.elasticsearch.client.RestClient;

import org.elasticsearch.client.RestHighLevelClient;

import org.elasticsearch.index.query.QueryBuilders;

import org.elasticsearch.search.SearchHit;

import org.elasticsearch.search.aggregations.AggregationBuilders;

import org.elasticsearch.search.aggregations.Aggregations;

import org.elasticsearch.search.aggregations.bucket.terms.Terms;

import org.elasticsearch.search.builder.SearchSourceBuilder;

public class EmployeeFetch {

void fetchEmp() throws IOException {

RestHighLevelClient client = new RestHighLevelClient(RestClient.builder(new HttpHost("localhost", 9200, "http")));

SearchRequest searchRequest = new SearchRequest();

searchRequest.indices("employeee");

SearchSourceBuilder searchSourceBuilder = new SearchSourceBuilder();

searchSourceBuilder.query(QueryBuilders.matchAllQuery());

searchSourceBuilder.aggregation(AggregationBuilders.terms("DISTINCT\_VALUES").field("age.keyword"));

searchRequest.source(searchSourceBuilder);

Map<String, Object> mapp=null;

try {

SearchResponse searchResponse = null;

searchResponse =client.search(searchRequest, RequestOptions.DEFAULT);

if (searchResponse.getHits().getTotalHits().value > 0) {

SearchHit[] searchHit = searchResponse.getHits().getHits();

for (SearchHit hit : searchHit) {

mapp = (HashMap<String, Object>) hit.getSourceAsMap();

System.out.println("map:"+Arrays.toString(mapp.entrySet().toArray()));

}

}

Aggregations aggregations = searchResponse.getAggregations();

List<String> list=new ArrayList<String>();

Terms aggTerms= aggregations.get("DISTINCT\_VALUES");

List<? extends Terms.Bucket> buckets = aggTerms.getBuckets();

for (Terms.Bucket bucket : buckets) {

list.add(bucket.getKeyAsString());

}

System.out.println("DISTINCT list values:"+list.toString());

} catch (IOException e) {

e.printStackTrace();

}

}

}