```
package com.subu.els.demo;

import org.elasticsearch.common.transport.InetSocketTransportAddress;
import java.util.Date;
import java.util.HashMap;
import java.util.Map;
import org.elasticsearch.action.delete.DeleteResponse;
import org.elasticsearch.action.get.GetResponse;
import org.elasticsearch.script.Script;
import org.elasticsearch.client.transport.TransportClient;
import org.elasticsearch.common.settings.Settings;
import org.elasticsearch.transport.client.PreBuiltTransportClient;
import java.net.InetAddress;
import java.net.UnknownHostException;
import org.elasticsearch.action.update.UpdateRequest;



public class ElasticSearchClient {

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

		Settings settings =Settings.builder().put("cluster.name", "my-application").build();
		//Settings settings =Settings.builder().put("cluster.name", "elasticsearch").build();

		TransportClient  client = new PreBuiltTransportClient(settings)
		.addTransportAddress(new
InetSocketTransportAddress(InetAddress.getByName("127.0.0.1"), 9300));
```

```
            insertData(client);
            //System.out.println("------------------------------");

           getDocument(client);
            //System.out.println("------------------------------");

            updateData(client);
            //System.out.println("------------------------------");



            deleteData(client);
            //System.out.println("------------------------------");

            searchIndex(client);
            System.out.println("------------------------------");




}


public static void insertData(TransportClient client){

            try{
                        client.prepareIndex("subu", "article", "1")
                            .setSource(putDocument("ElasticSearch and Java API",
                                                "ElasticSearch provides the Java API, all operations like CRUD ",
                                                            new Date(),
```

```
                                                    new String[]{"elasticsearch"},
                                                "Subhasish Sahu")).execute().actionGet();
            }
            catch(Exception e)
            {
                System.out.println("Exception in Creating Data and Inserting data");
            }
        }

        public static void getDocument(TransportClient client){

            try{
                GetResponse getResponse = client.prepareGet("subu", "article",
"1").execute().actionGet();
                Map<String, Object> source = getResponse.getSource();
                System.out.println("------------------------------");
                System.out.println("Index: " + getResponse.getIndex());
                System.out.println("Type: " + getResponse.getType());
                System.out.println("Id: " + getResponse.getId());
                System.out.println("Version: " + getResponse.getVersion());
                System.out.println("Exists: " + getResponse.isExists());
                System.out.println("Source:"+source);
                System.out.println("------------------------------");
            }
            catch(Exception e)
            {
                System.out.println("Exception in Searching Data"+e);
            }
        }
```

```
	public static void updateData(TransportClient client){

		try{
			UpdateRequest updateRequest = new UpdateRequest("subu", "article", "1")
			.script(new Script("ctx._source.tag = \"subhasish welcome to ES\""));
			client.update(updateRequest).get();
		}
		catch(Exception e)
		{
			System.out.println("Exception in Searching Data");
		}
	}

	public static void deleteData(TransportClient client){

		try{
			DeleteResponse response = client.prepareDelete("subu", "article",
"1").execute().actionGet();
			System.out.println("Id:"+response.getId());
			System.out.println("Type:"+response.getType());
			System.out.println("Index:"+response.getIndex());
			System.out.println("Status: " + response.status());
		}
		catch(Exception e)
		{
			System.out.println("Exception in Searching Data");
		}
	}
```

```
	public static void searchIndex(TransportClient client){

		try{
			boolean exists = client.admin().indices()
					.prepareExists("subu")
					.execute().actionGet().isExists();
			System.out.println("Index exists:"+exists);

		}
		catch(Exception e)
		{
			System.out.println("Exception in Searching Data");
		}
	}

	public static Map<String, Object> putDocument(String title, String content, Date postDate,
			String[] tags, String author){

		Map<String, Object> returnDocument = new HashMap<String, Object>();
		returnDocument.put("title", title);
		returnDocument.put("content", content);
		returnDocument.put("date", postDate);
		returnDocument.put("tag", tags);
		returnDocument.put("author", author);

		return returnDocument;
	}
}
```