Trial Day (Senior, Java, SpringBoot, ElasticSearch): Emergency Services Dashboard with Real-Time Search Capabilities

Objective: Develop a Java application using Spring Boot that acts as an emergency services dashboard. The application should interact with ElasticSearch to log and query real-time emergency incidents.

Specifications / Rules:

1) General:

- **Spring Boot Setup**: Use Spring Boot for setting up the application.
- **ElasticSearch Integration**: Integrate ElasticSearch for real-time search capabilities.
- **Data Persistence**: Use Hibernate for data persistence. Sync this data to ElasticSearch for optimized searching.
- **Unit Testing**: Implement unit tests for at least two core functionalities: incident logging and incident search.

2) Incidents:

- **Create and Log Incidents**:
  - Implement a RESTful API endpoint to log emergency incidents.
  - Incidents should have attributes like `incidentType` (e.g., fire, medical), `location` (latitude and longitude), `timestamp`, and `severityLevel` (e.g., low, medium, high).
- **Search Incidents**:
  - Create another API endpoint to perform searches based on parameters such as `incidentType`, `location`, and `timestamp`.
  - Allow combination searches (e.g., all 'fire' incidents in a specific 'location').

3) ElasticSearch:

- **Data Indexing**:
  - Use ElasticSearch to index the emergency incidents. The indexed data should be optimized for efficient querying.
- **Search Optimization**:
  - Leverage ElasticSearch's capabilities to ensure that search queries are fast and yield accurate results.

4) Optional Task - Real-Time Dashboard:

- **WebSocket Integration**:
  - Integrate a WebSocket using Spring to provide real-time updates of incidents.
- **UI (Optional)**:
  - Implement a simple UI using Thymeleaf to visualize these real-time updates. This is for extra points and is not mandatory.

5) Optional Task - Dockerization:

- **Docker Setup**:
  - Dockerize the application, including ElasticSearch, to make it portable and easy to deploy.

Notes for Developer:

- Please state clearly in the README file which features have been implemented and which have not.
- Include a simple guide on how to set up and run the project.

Use Docker to run Elasticsearch in development mode - check via http://localhost:9200/

```
docker pull docker.elastic.co/elasticsearch/elasticsearch:8.10.2
docker run --name elasticsearch -d -p 9200:9200 -p 9300:9300 -e "discovery.type=single-node" docker.elastic.co/elasticsearch/elasticsearch:8.10.2
```

Create the index and check it

```
curl -X PUT "localhost:9200/incidents" -H "Content-Type: application/json" -d'
{
  "settings" : {
    "index" : {
      "number_of_shards" : 1,
      "number_of_replicas" : 1
    }
  },
  "mappings": {
    "properties": {
      "incidentType": { "type": "keyword" },
      "location": { "type": "geo_point" },
      "timestamp": { "type": "date" },
      "severityLevel": { "type": "keyword" }
    }
  }
}'


curl -X GET "localhost:9200/_cat/indices?v"
```