

Các bước setup trên centos 7

Cài jdk

sudo yum install java-1.8.0-openjdk.x86\_64

kiểm tra phiên bản java:

[centos@centos7 ~]$ **java -version**

openjdk version "1.8.0\_372"

OpenJDK Runtime Environment (build 1.8.0\_372-b07)

OpenJDK 64-Bit Server VM (build 25.372-b07, mixed mode)

Install wget

Sudo yum install wget

Kết nối wget tới elasticsearch

wget <https://artifacts.elastic.co/downloads/elasticsearch/elasticsearch-7.9.2-x86_64.rpm>

install nó:

sudo rpm -ivh elasticsearch-7.9.2-x86\_64.rpm

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-elasticsearch-on-centos-7>

Sau khi download và chạy elasticsearch thì với máy ảo vmware thì bên máy chính cần bật tường lửa

**Các câu lệnh truy vấn ES:**

ES chạy trên http method nên có đủ các thuộc tính của http:

Lưu trữ dữ liệu dạng json cơ sở dữ liệu nosql.

Các khái niệm cơ bản:

* Index:
  + Index trong Elasticsearch tương đương với cơ sở dữ liệu trong hệ thống quan hệ SQL.
  + Nó đại diện cho một tập hợp các tài liệu có liên quan.
  + Mỗi index có một tên duy nhất và chứa các tài liệu được lưu trữ dưới dạng JSON hoặc BSON (Binary JSON).
* Mapping:
  + Mappings trong Elasticsearch xác định cấu trúc dữ liệu của các tài liệu trong index.
  + Nó giống như schema trong hệ thống quan hệ SQL và xác định các trường (fields) và kiểu dữ liệu của chúng.
  + Mappings cung cấp thông tin về cách Elasticsearch nên xử lý và lưu trữ dữ liệu trong index.
* Query:
  + Query là cách để truy vấn dữ liệu từ Elasticsearch.
  + Elasticsearch cung cấp một loạt các truy vấn để tìm kiếm và lấy dữ liệu từ index.
  + Các truy vấn có thể được xác định bằng cách sử dụng JSON hoặc thông qua API RESTful của Elasticsearch.
  + Elasticsearch hỗ trợ các truy vấn phong phú bao gồm truy vấn văn bản đầy đủ (full-text), truy vấn phủ định (negative query), truy vấn mờ (fuzzy query), truy vấn đa tiêu chí (bool query), truy vấn về khoảng (range query) và nhiều loại truy vấn khác.
* Aggegation:
  + Aggregation cho phép bạn thực hiện các phép tính tổng hợp và phân tích trên dữ liệu trong Elasticsearch.
  + Nó tương tự như GROUP BY và các hàm tổng hợp trong hệ thống quan hệ SQL.
  + Aggregation được sử dụng để tính toán thống kê, tổng hợp, đếm, trung bình, tìm giá trị lớn nhất/nhỏ nhất và nhiều loại phân tích khác trên các trường trong tài liệu.

Các câu lệnh phổ biến:

Xem các index có trên ES:

**GET \_cat/indices?v**

**Hoặc: curl -XGET 'http://localhost:9200/\_cat/indices?v'**

**API tạo Index trong Elasticsearch**

**PUT /{index\_name}**

**Xem các trường trong index**

**GET /your\_index\_name/\_mappings**

**API xóa Index trong Elasticsearch**

**DELETE /index\_name**

API Insert, thêm dữ liệu vào Elasticsearch

POST /index\_name/type\_name/id

{

"field" : "value"

}

PUT /index\_name/type\_name/id

{

"field" : "value"

}

API get document Elasticsearch.

GET /index\_name/type\_name/id

GET /index\_name/type\_name/id/\_source lấy dữ liệu data source

Muốn lấy các trường riêng dùng \_source?field1,field2…

API đọc dữ liệu – Tìm kiếm dữ liệu, document Elasticsearch

Đọc tất cả dữ liệu: GET /\_search

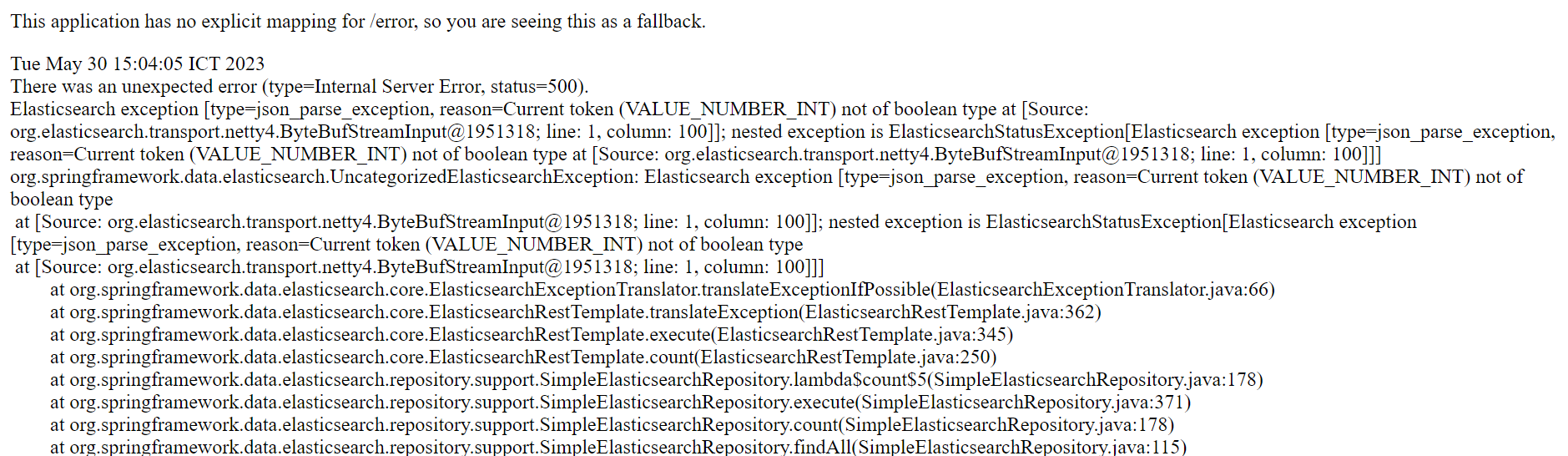
Tìm keyword: GET /\_search?q=word hoặc search theo field GET /\_search?q=name:word

API Xóa dữ liệu trong Elasticsearch

DELETE /index\_name/type\_name/id

Lỗi

Source: org.elasticsearch.transport.netty4.ByteBufStreamInput@1294639; line: 1, column: 100]]; nested exception is ElasticsearchStatusException[Elasticsearch exception [type=json\_parse\_exception, reason=Current token (VALUE\_NUMBER\_INT) not of boolean type



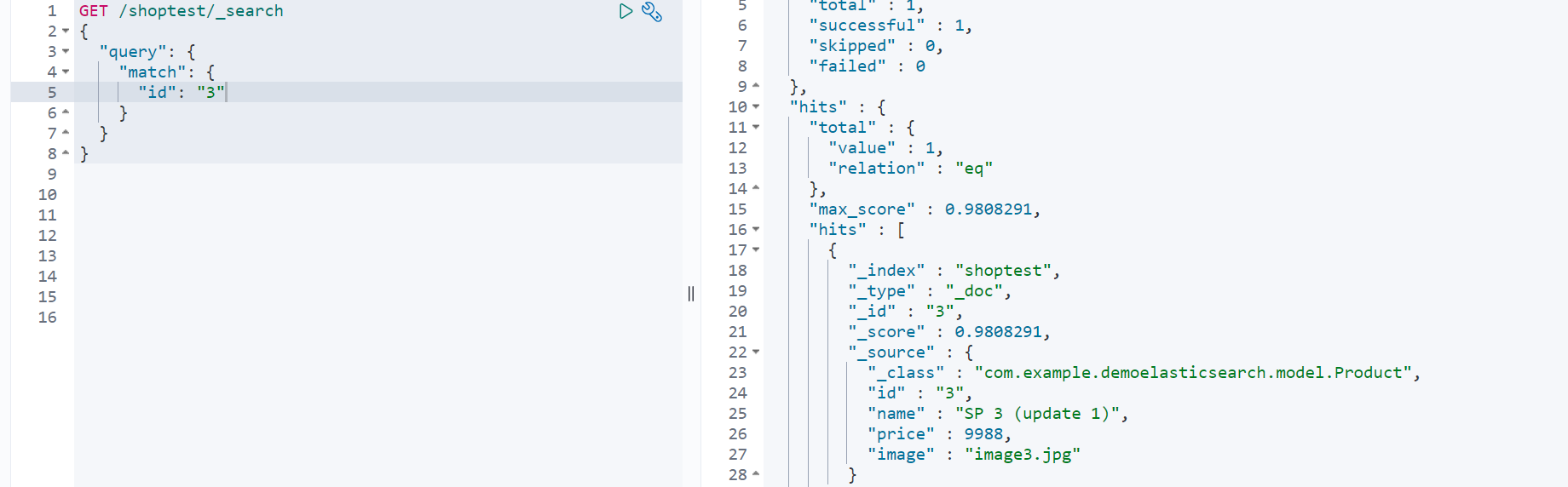
Các aggregation:

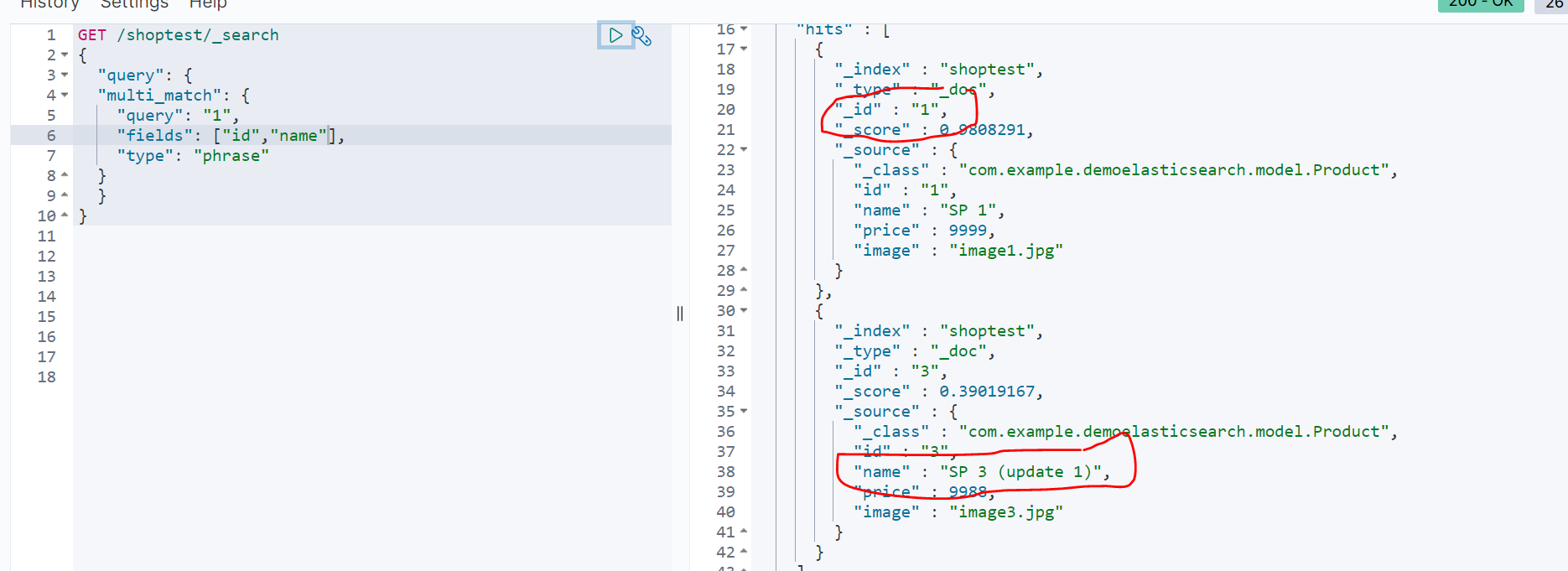
1. **Terms Aggregation**: Phân loại các giá trị theo một trường cụ thể và tính toán số lượng hoặc các thống kê khác cho mỗi giá trị. Ví dụ: đếm số lượng sản phẩm theo danh mục.
2. **Range Aggregation**: Phân loại các giá trị vào các khoảng giá trị được xác định trước và tính toán số lượng hoặc các thống kê khác cho mỗi khoảng. Ví dụ: đếm số lượng người dùng theo nhóm tuổi.
3. **Date Histogram Aggregation**: Phân loại các giá trị ngày thành các khoảng thời gian (ví dụ: theo ngày, tuần, tháng) và tính toán số lượng hoặc các thống kê khác cho mỗi khoảng. Ví dụ: đếm số lượng giao dịch theo tuần.
4. **Nested Aggregation**: Aggregation lồng nhau cho các trường có kiểu dữ liệu nested trong Elasticsearch. Điều này cho phép phân tích đa cấp trên các cấu trúc dữ liệu phức tạp. Ví dụ: tính toán tổng số sản phẩm theo danh mục và nhóm con của danh mục.
5. **Stats Aggregation**: Tính toán các thống kê cơ bản như min, max, sum, avg và count cho một trường cụ thể. Ví dụ: tính toán giá trị trung bình của một trường.
6. **Filter Aggregation**: Áp dụng một bộ lọc trên dữ liệu và tính toán số lượng hoặc các thống kê khác cho các tài liệu phù hợp với bộ lọc đó. Ví dụ: đếm số lượng người dùng có tuổi trên 18.
7. **Cardinality Aggregation**: Tính toán số lượng giá trị duy nhất của một trường cụ thể. Ví dụ: đếm số lượng địa điểm khác nhau mà người dùng đã đến.
8. **Top Hits Aggregation**: Trả về các tài liệu hàng đầu dựa trên một tiêu chí nhất định, chẳng hạn như điểm số tương quan. Ví dụ: lấy 5 sản phẩm có điểm đánh giá cao nhất.

Các câu query thường dùng:

1. Math query

Là một loại query phổ biến nó sẽ trả về kết quả nếu bất kỳ document nào chứa query truyền lên (tìm kiếm theo trường riêng biệt )



1. Match-phrase
   1. Search chính xác.
2. Multi-match
   1. Giống như math query nhưng nó search trên nhiều fiel
3. Term query

Term query thường dùng cho dạng số.



1. Range query

Dùng để chạy điều kiện trong 1 khoảng.



1. Bool query

Bool query dùng để kết hợp nhiều query khác nhau.

Các công việc đã hoàn thành

* Setup Elasticsearch, kibana trên mỗi trường windown
* Tìm hiểu được các khái niệm cơ bản như: index, mapper, query, aggregations
* Tổng hợp các index thao tác trên java với các chức năng: CRUD, search với RestHighLevelClient
* Cài đặt thử trên môi trường Vmware Centos. Phần code connect tới máy ảo còn lỗi connect.

**Các aggregation**

Kiểm tra xem số lượng sp trong khoảng: 

Sum balace state DC:

