**Search, Storage, Web interface**

Phan Hữu Phước - 1412420

Table of Contents

[I. Tìm kiếm dữ liệu: 3](#_Toc496129667)

[1. Tìm kiếm 3](#_Toc496129668)

[2. Query string 3](#_Toc496129669)

[3. Lưu kết quả search 5](#_Toc496129670)

[4. Mở file search đã save 5](#_Toc496129671)

[5. Làm mới kết quả search 5](#_Toc496129672)

[II. Storage dự liệu 6](#_Toc496129673)

[III. Web Interface 6](#_Toc496129674)

# Tìm kiếm dữ liệu:

## Tìm kiếm

* + - Chúng ta có thế tìm kiếm dự liệu bằng cách nhập trực tiếp từ thanh tìm kiếm của Kibana, dữ kiệu cí thế là text plain, Lucene query syntax, hoặc là JSON-based Elasticsearch Query DSL.
    - Khi ta submit search request, kết quả sẽ được update tương ứng của phần kết quả search (documents table). Tổng số dòng tìm kiếm được sẽ hiện ở thanh tìm kiếm, chúng ta cũng có thể sắp xếp kết quả tìm kiếm dựa trên các trường.
    - Thực hiện tìm kiếm
      * Để tìm kiếm bằng text plain, đơn giản ta chỉ việc nhập text vào thanh công cụ tìm kiếm.
      * Đề tìm kiếm cho một trường nào đó, chỉ rõ trường đó ở trước dữ liệu muốn tìm. Ví dụ: status:404 để tìm kiếm tất cả các dữ liệu có status là 404 ở trường stautus
      * Để tìm kiếm trong một mien52 giá trị, chỉ rõ [START\_VALUE TO END\_VALUE]. Ví dụ để tìm status có giá trị 4xx ta sẽ dùng [400 TO 499]
      * Đề tìm kiếm kết hợp các điều kiện phức tạp, ta dùng các toán tử AND, OR, NOT.

## Query string

1. Giải thích các trường của một câu query

GET /\_search

{

"query": {

"query\_string" : {

"default\_field" : "content",

"query" : "this AND that OR thus"

}

}

}

|  |  |
| --- | --- |
| Parameter | Mô tả |
| query | Câu lệnh truy vấn |
| default\_field | Các trường mặc định nếu không có trường được chỉ định.  Mặc dịnh \_all |
| default\_operator | Toán tự mặc định sử dụng OR/AND |
| analyzer | Analyzer name để phân tích câu query |
| allow\_leading\_wildcard | Khi ta set trường này, \* hoặc ? đươc phép dùng như kí tự đầu tiên |
| enable\_position\_increments | Set true để tăng ví trí trong các câu truy vấn |
| fuzzy\_max\_expansions | Kiếm soát các cụm tự truy vấn mở rộng |
| fuzzy\_prefix\_length | Tiền tố cho fuzziness query |
| fuzziness | Set the fuzziness for fuzzy queries |
| phrase\_slop | Thiết lặp slop cho tiềm kiếm, nếu không, cụm từ chính xác sẽ được tìm kiếm |
| boost | Set giá trị boots cho query |
| auto\_generate\_phrase\_queries | Mặc định false |
| analyze\_wildcard | Phân tích wildcard |
| max\_determinized\_states | Giới hạng số lượng query regex state được tạo |
| minimum\_should\_match | Giới hạn số lượng match |
| lenient | Thiết lặp true nếu muốn bỏ qua syntax error |
| time\_zone | Set time zone |
| quote\_field\_suffix | Hậu tố cho query string |
| split\_on\_whitespace | Tách khoảng cách trước khi query |
| all\_fields | Query trên tất cả các trường |

1. Query string syntax

* Field name
  + status field chứa **active**

🡪 status:active

* + title chứa quick hoặc brown

🡪 title:(quick OR brown) hoặc title:(quick brown)

* + Chứa chính xác trường author có giá trị **austin**

🡪 author:”austin”

* + các trường book.title, book.content chứa giá trị quick hoặc brown

🡪 book.\\*:(quick brown)

* + title chứa giá trị not null

🡪 \_exists\_:title

* Wildcards
  + Sử dụng \* để thay thế một hoặc nhiều
  + Sử dụng ? để thay thế một kí tự
* Regular expressions
  + Chúng ta có thế nhúng regex bằng cách đặt chúng giữa đầu `/`

🡪 Ví dụ: name:/joh?n(ath[oa]n)/

* Ranges
  + Cú pháp [min TO max]
  + Ví dụ
    - Tất cả ngày trong năm 2012:

🡪 date:[2012-01-01 TO 2012-12-31]

* + - Số từ 1 đến 5

🡪 count:[1 TO 5]

* + - Tags giữa alpha và omega

🡪 tag:{alpha TO omega}

* + - Số lớn hơn 10

🡪 count:[10 TO \*]

* + - Ngày giữa năm 2012

🡪 date:{\* TO 2012-01-01}

* + - Số lớn 1 nhỏ hơn 5

🡪 count:[1 TO 5}

* Reserved characters
  + Nếu muốn dùng, ta phải escape bằng dấu `\`
  + Các reserved words: + - = && || > < ! ( ) { } [ ] ^ " ~ \* ? : \ /
* Grouping

Sử dụng toán tử

* + (quick OR brown) AND fox
  + status:(active OR pending) title:(full text search)^2

## Lưu kết quả search

* + - Chúng ta có thế lưu lại kết quả search coh query string và kết quả hiên tại
      * Chọn save ở thanh toolbar
      * Đặt tên cho file và save lại

## Mở file search đã save

* + - Chọn open ở Kibana toobar
    - Chọn file search muốn mở

## Làm mới kết quả search

* + - Chọn Time Picker ở thanh công cụ của Kibana
    - Chọn Auto Refresh
    - Chọn một khoảng thời gian ta muốn search tự động resey

🡪 Khi thực hiện set tự động refresh, Kibana sẽ tự động làm mới kết quả tìm kiếm theo khoảng thời gian mà bạn đã chọn

# Storage dự liệu

* Logstash chứ dữ liệu dựa vào elastic search
* Tóm tắt về cách lưu trữ dữ liệu elastic search. Dữ liệu của elastic search lưu dạng inverted index.
* Inverted index: Thay vì index theo từng row dữ liệu như cơ sở dữ liệu quan hệ thông thường, chúng ta sẽ biến chúng thành index theo dạng term/
* Ví dụ ta có 3 document như sao

D1 = “Toi den truong”

D2 = “Toi den cua hang”

D3 = “Toi o nha”

🡪 Ta sẽ có index của 3 document là:

“Toi” = {D1, D2, D3}

“den” = {D1, D2}

“truong” = {D1}

“cua” = {D2}

“hang” = {D2}

“o” ={D3}

“nha”={D3}

🡪 Dự vào kết quả, ta thấy rằng rất dễ dàng để tìm kiếm dữ liệu trên các index này. Giả sử ta muốn tìm từ “toi den”, ta sẽ query

{D1, D2} ∩ {D1, D2, D3} = {D1, D2}

Kết quả thu được là {D1, D2}

# Web Interface

Chúng ta dùng Kibana cho web interface

* Kibana là một open source cho UI plugin cho Elasticsearch.
* Kibana cung cấp một giao diện người dùng thiện cho người, bao gồm histograms, line graphs, pie charts, sunbursts,.. và tập hợp đầy đủ khả năng của ElasticSearch
* Biểu diễn GEO data lên bản đồ
* Time series cho việc phân tích dữ liệu tương lai
* Truy vấn, lọc dữ liệu
* Phân tích một liên hệ của dự liệu qua biểu đồ