# Version History

|  |  |  |
| --- | --- | --- |
| Version | Date | Release Note |
| V1.0 | 10-Mar-2020 | Initial Draft |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

index

1 Version History 2

2 Exercise 1 4

2.1 Create new repository 4

2.2 Chekcout from central repository 8

2.3 Daily Operations 11

2.4 Merge conflict 11

2.5 Raise pull request 11

2.6 Merge Branch 11

# Exercise 1

Limit the access request by Zuul

<https://www.baeldung.com/spring-cloud-zuul-rate-limit>

## Zuul

Zuul is the front door for all requests from devices and web sites to the backend of the Netflix streaming application. As an edge service application, Zuul is built to enable dynamic routing, monitoring, resiliency and security. It also has the ability to route requests to multiple Amazon Auto Scaling Groups as appropriate..

### Create sample endpoint

Free text search works within all fields—including the \_source  field, which includes all the other fields. If no specific field is indicated in the search, the search will be done on all of the fields that are being analyzed.

In the search field at the top of the Discover page, run these searches and examine the result (set the time parameter on the top right of the dashboard to the past month to capture more data):

* category
* Category
* categ
* cat\*
* categ?ry
* “category”
* category\/health
* “category/health”
* Chrome
* chorm\*

#### **Tips and Gotchas**

1. Text searches are not case sensitive. This means that "category" and "CaTeGory" will return the same results. When you put the text within double quotes (""), you are looking for an exact match, which means that the exact string must match what is inside the double quotes. This is why [category\/health] and ["category/health"] will return different results
2. Kibana wildcard searches—you can use the wildcard symbols [\*] or [?] in searches. [\*] means any number of characters, and [?] means only one character

### Field-Level Searches

Another common search in Kibana is field-level queries, sued for searching for data inside specific fields. To use this type of search that, you need to use the following format:

<fieldname>:search

As before, run the following searches to see what you get (some will purposely return no results):

name:chrome

name:Chrome

name:Chr\*

response:200

bytes:65

bytes:[65 TO \*]

bytes:[65 TO 99]

bytes:{65 TO 99}

\_exists\_:name

#### Tips and Gotchas

* Field-level searches depend on the type of field. (Logz.io users—all fields are not analyzed by default, which means that searches are case-sensitive and cannot use wildcard searches. The reason we save all of the fields as "not analyzed" is to save space in the index because the data is also duplicated in an analyzed field called  \_source )
* You can search a range within a field. If you use [], this means that the results are inclusive. If you use {}, this means that the results are exclusive.
* Using the \_exists\_   prefix for a field will search the documents to see if the field exists.
* When using a range, you need to follow a very strict format and use capital letters TO to specify the range

### Logical Statements

* You can use logical statements in searches in these ways:
* USA AND Firefox
* USA OR Firefox
* (USA AND Firefox) OR Windows
* -USA
* !USA
* +USA
* NOT USA

#### **Tips and gotchas**

* You need to make sure that you use the proper format such as capital letters to define logical terms like AND or OR
* You can use parentheses to define complex, logical statements
* You can use -,! and NOT to define negative terms

### Kibana Special Characters

All special characters need to be properly escaped. The following is a list of all available special characters:

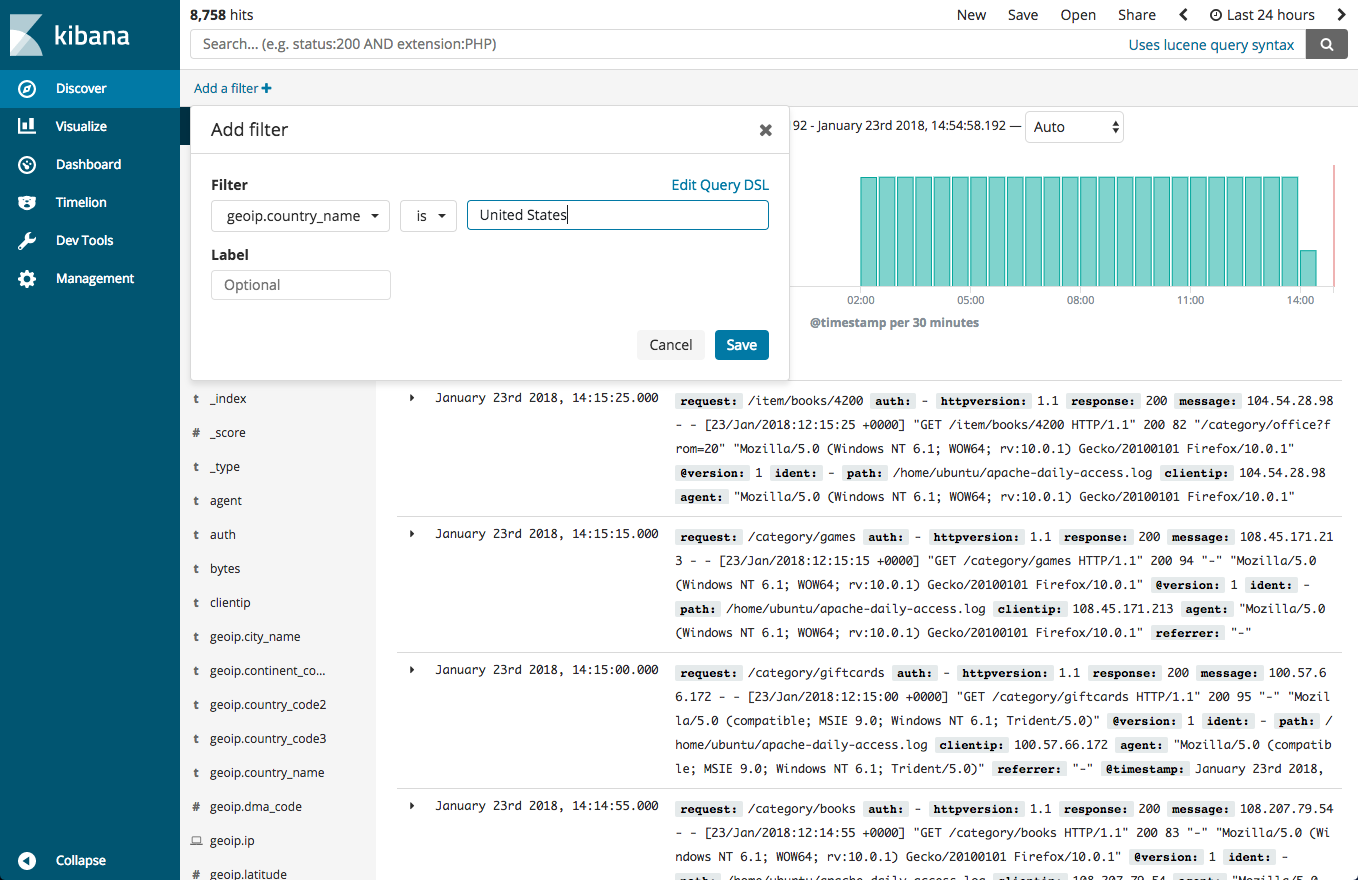
+ - && || ! ( ) { } [ ] ^ " ~ \* ? : \

### Proximity searches

Proximity searches are an advanced feature of Kibana that takes advantage of the Lucene query language.

[categovi~2] means a search for all the terms that are within two changes from [categovi]. This means that all category will be matched.

#### Tips and gotchas

Proximity searches use a lot of system resources and often trigger internal circuit breakers in Elasticsearch. If you try something such as [catefujt~10], it is likely not to return any results due to the amount of memory used to perform this specific search.

### Kibana Filtering Dialog

To assist users in searches, recent versions include a filtering dialog that allows easier forming of Kibana search syntax.

To use the dialog, simply click the Add a filter + button under the search box and begin experimenting with the conditionals.

Power users can also enter Elasticsearch queries using the Query DSL.

### Rest API Search

END of Document