USING KIBANA TO EXECUTE QUERIES IN ELASTICSEARCH USING LUCENE AND KIBANA QUERY LANGUAGE

We have discussed at length how to query ElasticSearch with CURL. Now we show how to do that with Kibana.

You can follow this blog post to populate your ES server with some data.

Using JSON
JSON queries (aka JSON DSL) are what we use with curl. But you can use those with Kibana too. It will not work with aggregations, nested, and other queries.

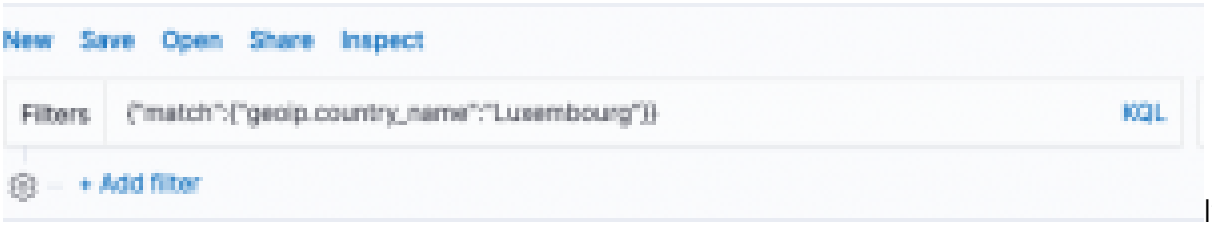
In using JSON, difference is that you only pass in the **query** object. So for this curl query:

```
{"query":{"match":{"geoip.country_name":"Luxembourg"}}}
```

You would paste in only this portion in Kibana.

```
{"match":{"geoip.country_name":"Luxembourg"}}
```

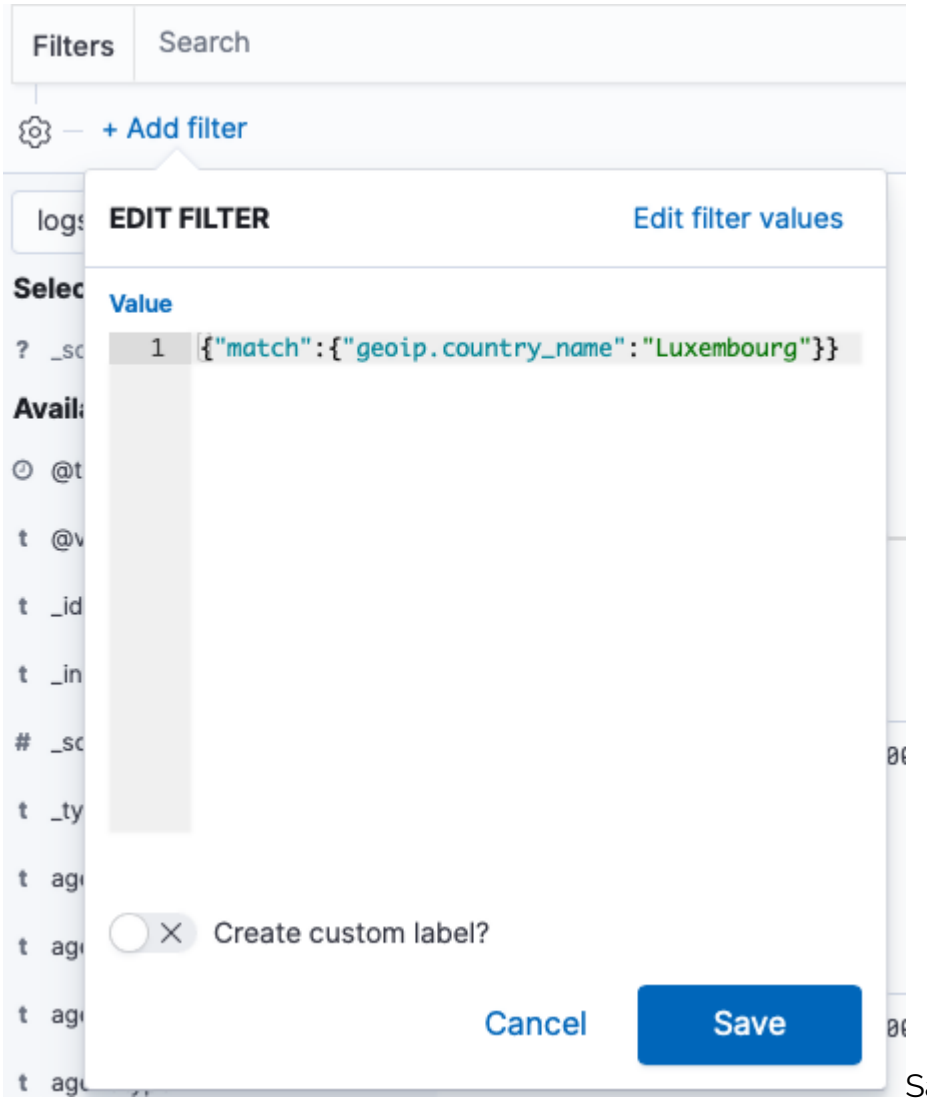
Entering Queries in Kibana
In the **Discovery** tab in Kibana, paste in the text above, first changing the query language to Lucene from KQL, making sure you select the logstash* index pattern. We discuss the Kibana Query Language (KBL) below.



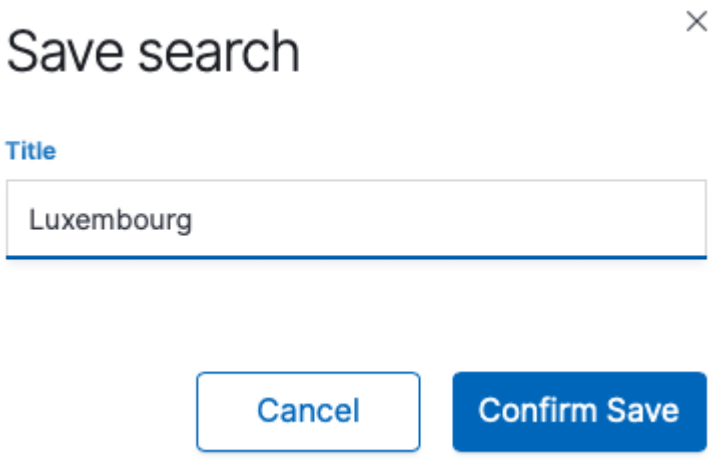
If you forget to change the query language from KQL to Lucene it will give you the error:

```
Discover: input.charAt is not a function. (In 'input.charAt(peg$currPos)',
'input.charAt' is undefined)
```

The easiest way to enter the JSON DSL query is to use the query editor since it creates the query object for you:



Save the query, giving it some name:



Kibana Query Language (KBL) versus Lucene

You can use KBL or Lucene in Kibana. They are basically the same except that KBL provides some simplification and supports scripting.

Here are some common queries and how you do them in each query language.

KBL	Lucene	Explanation
request:"/wordpress/"	request:"/wordpress/"	The colon (:) means equals to. Quotes mean a collection of words, i.e. a phrase.
request:/wordpress/	request:/wordpress/	Do don't need quotes for one word.
request:/wordpress/	request:/wordpress/	Do don't need quotes for one word.
request:/wordpress/ and response:404	request:/wordpress/ and response:200	for KBL you have to explicitly put the boolean operator. For Lucene the operator is not recognized as an operator but as a string of text unless you use write it in capital letters.
wordpress	wordpress	Matches based on any text (**wordpress** in this example) in the document and not a specific field.
200 or 404	200 404	adding the word **or** to Lucene would also include text containing the string "or." So leave it off or use capital OR.
200 and 404	200 AND 404	Use uppercase with Lucene for logical operators.
geoip.country_name: "Luxembourg"	{"match":{"geoip.country_name": "Luxembourg"}}	Lucene supports JSON DSL query language, as we illustrated above

response:>=200 and response:< =404	response:	range query

kilobytes > 1	not supported	Scripted field, where **kilobytes** is:
		if (doc.size()==0) { return 0;
		}
		return doc.value / 1024;