Squirro Academy
Cheat Sheet - Query Syntax

Query syntax refers to the way users structure their search inputs. By using operators (AND, OR, NOT, +,-), brackets ( ), wildcards (*,?), and quotation marks (" "), you can narrow or broaden your search results. By starring, boosting, sorting, and manipulating time increments, you can adjust how your results are presented.

Boolean Operators

OPERATOR	⚠ REQUIREMENTS
AND OR NOT }	Operators must be entered in all capital letters.
+ }	Term after the **+** symbol must exist somewhere.
- }	Excludes documents that contain the term after the **-** symbol.

EXAMPLE BOOLEAN QUERIES

| squirro **AND** memonic	Search documents that contain squirro and memonic.
squirro **OR** memonic	Search documents that contain either squirro or memonic.
+memonic **-**squirro	Search documents that contain memonic but do not contain squirro.
squirro **NOT** memonic	Search documents that contain squirro but do not contain memonic.

Wildcard Search

- Two wildcard operators are permitted:
-   ***** matches 0 or more characters.
-   **?** matches any single character.
- Leading wildcards are allowed, but note that they can be very slow and potentially timeout within Elasticsearch. To prevent slow queries, avoid beginning the search query with ***** or **?**
- Wildcard search can also apply per field.

EXAMPLE WILDCARD QUERIES

squirr*****	Search documents that contain (e.g.) "squirro" and "squirrel".
*emonic	Search documents that contain (e.g.) "memonic" and "mnemonic". (As noted above, this has the potential to be an extremely slow search.)
te**?**t	Search documents that contain (e.g.) "test" and "text".
name:*****	Search documents that have (e.g.) the field "name".[1]
-name:*****	Search documents that do not have (e.g.) the field "name". [1]
name:squir*****	Search documents that contain the "name" field started by "squir", (e.g.) "name:squirro" and "name:squirrel". [1]

[1] Facet names containing spaces need to be put inside quotes within queries.
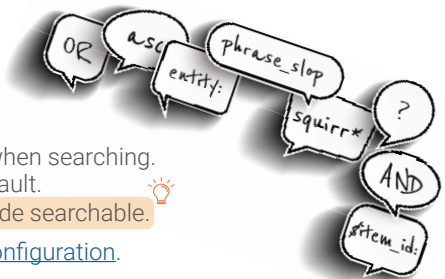
Defaults

- 'Title' and 'body' fields are taken into account when searching.
- Search terms entered are **OR** combined by default.
- Dynamically tagged text labels can also be made searchable.

🔖 Learn more about [Default Term Matching Configuration](#).

Phrase Search

- Use double quotes " " around the search terms to perform a phrase search.
- Phrase search returns results found within close distance (the default distance is **5** terms.)
- Manually specify the `phrase_slop` distance between terms using a tilde **~**

EXAMPLE PHRASE QUERIES

"oracle financial services"	Find documents where oracle, financial and services are found within the configured default `phrase_slop` distance. See query strategy configuration (`topic.search.query-strategy: :phrase.phrase_slop`)
"oracle financial services"**~**1	Find documents where oracle, financial and services match in exactly this order and within three terms.
"oracle financial leasing"**~**3	Find documents where oracle, financial and leasing must match but allow for up to 3 additional terms between them. The order of the terms is no longer strict, but swapping two words is equivalent to adding two words in terms of edit distance.

Grouping

Use round brackets/parentheses **( )** to group search terms.

EXAMPLE GROUPED QUERIES

(java **AND** solar) **OR** (python AND elasticsearch)	Search documents that contain both java and solar, or documents that contain both python and elasticsearch.
nektoon **AND** (squirro **OR** memonic)	Search documents that contain nektoon and either squirro or memonic.

Entity Search

Query syntax to search for items having entities satisfied some criteria:

entity :{< any query to match a single entity document >**}**

EXAMPLE ENTITY QUERIES

entity:{type:company **AND name:"**Thomson Reuters**"}**	Search for Items containing at least one company-typed Entity "Thomson Reuters" and another one Entity "Squirro".
entity:{type:company **AND name:"**Thomson Reuters**" AND confidence >** 0.8**}**	Search for Items containing a specific Entity of type company with a confidence higher than 80%.

Starred and Read Items

- Starred items are items marked as favorite / bookmarked items.
- **Flags** must be enabled for your project(s) before you are able to query for starred and read items.
- To do so, enable the enable_flags_for_project_ids property in topic.ini.

Query syntax for (un)starred items and (un)read items:

1	is:starred	2	is:unstarred	3	is:read	4	is:unread

Sorting

You can use the following query syntax to sort the result:

sort:<field_name>[**:**<order>]

The square brackets mean the fields are optional. The brackets are not part of the syntax.

Where <field_name> is either **date** (default) or **relevance** or any item field name you want to sort by and <order> is either **asc** for ascending or **desc** for descending. The order suffix is optional, the default order is descending.

Additionally, you can add a second (or third etc.) sorting criteria by adding the following to the query syntax:

[**;**<2nd_sort_field>[**:**<2nd_order>]]

EXAMPLE SORTED QUERIES

sort:date	Sort by date (descending order by default).
sort:date:asc	Sort by date in ascending order.
sort:relevance:desc	Sort by relevance in descending order.
sort:<my_sortable_facet> **:desc;date:desc**	Sort by "my_sortable_facet" in descending order; additionally add a second sorting by descending date.

Facet Search

Use any document facet to restrict the search.

⚠ Facet searches are case sensitive.

EXAMPLE BOOSTED QUERIES

Country:France	Search documents that have a facet named Country with a value France.
Country:"United Kingdom"	Search documents that have a facet named Country with a value United Kingdom. [2]
"Mixed Sentiment":Yes	Search documents that have a facet named Mixed Sentiment with a value Yes. [2]

[2] Facet names containing spaces need to be put inside quotes within search queries.

Time Increment

Time increments shown in the main timeline and in the dashboard widgets can be controlled using **time_increment:**<value>

Possible values include:

1	time_increment:minute
2	time_increment:hour
3	time_increment:day
4	time_increment:week
5	time_increment:month
6	time_increment:quarter
7	time_increment:year

Values can also be further defined for increased flexibility. For example:

1	time_increment:12hours
2	time_increment:4days
3	time_increment:8weeks
4	time_increment:6months
5	time_increment:3year

⚠ There is a performance impact when using a time increment that results in many individual increments. This impact is both in the user interface, where each increment needs to be drawn, as well as on the Elasticsearch level, where they need to be calculated. Accordingly, use the setting carefully.

Boosting

- Individual elements of a query can be prioritized by boosting them.
- Sorting needs to be by relevance to notice the changed relevance scores.

EXAMPLE BOOSTED QUERIES

France^10 Europe	Search for France and Europe, but boost matches of "France".
France **OR Country:**-France^10	Search for France in full text, as well as the "Country" facet and boost items that have the value defined in the country facet.
France^0.1 Europe	Search for France and Europe, but de-prioritize matches of "France" (the default boost is 1.0).

Field Search

Only search within specific fields.

EXAMPLE FIELD QUERIES

$title: France	Search documents with the term France in the title.
$body: France	Search documents with the term France in the body.
$item_id: PgnAQM1FTSCP1uNOesoE7Q	Search for a specific document by id.
$item_created_at >= "2015-02-01T00:00:00**"**	Search documents created after Feb. 2, 2015.
$item_created_at <= "2015-02-01T00:00:00**"**	Search documents created before Feb. 2, 2015.
$item_created_at >= "now-7d/d**"**	Search documents created in the last 7 days (see Elasticsearch documentation)
$_size > 100000	Search documents with size > 100,000 bytes