Overview

Although Logsene provides the ability to create your own queries through its Elasticsearch API, it also provides a rich, yet simple query syntax very much like the query syntax used by Google.

Terms

A query is broken up into terms and operators. There are two types of terms: Single Terms and Phrases.

A Single Term is a single word such as "test" or "hello".

A Phrase is a group of words surrounded by double quotes such as "hello dolly".

Multiple terms can be combined together with Boolean operators to form a more complex query (see below).

Fields

When performing a search you can either specify a field, or use the default field. The field names depend on how you've structured your logs.

You can search any field by typing the field name followed by a colon ":" and then the term you are looking for.

As an example, let's assume your logs contain two fields, *message* and *hostname.* If you want to find an event mentioning "connection closed" from a host "foo.example.com", you can enter:

```
message:"connection closed" AND hostname:foo.example.com
```

Since *message* and *hostname* fields will be included in the default field that is searched, which Logsene creates for you behind the scenes, the field indicator is not required.

Note: The field is only valid for the term that it directly precedes, so the query

```
message:connection closed
```

Will only find "connection" in the *message* field. It will find "closed" in the default field.

Term Modifiers

Logsene supports modifying query terms to provide a wide range of searching options.

Wildcard Searches

Logsene supports single and multiple character wildcard searches within single terms (not within phrase queries).

To perform a single character wildcard search use the "?" symbol.

To perform a multiple character wildcard search use the "*" symbol.

The single character wildcard search looks for terms that match that with the single character replaced. For example, to search for "text" or "test" you can use the search:

```
te?t
```

Multiple character wildcard searches looks for 0 or more characters. For example, to search for test, tests or tester, you can use the search:

```
test*
```

You can also use the wildcard searches in the middle of a term.

```
te*t
```

Note: You cannot use a * or ? symbol as the first character of a search.

Regular Expression Searches

Logsene supports regular expression searches matching a pattern between forward slashes "/". For example, to find events containing "moat" or "boat":

```
/[mb]oat/
```

Fuzzy Searches

Logsene supports fuzzy searches based on Damerau-Levenshtein Distance. To do a fuzzy search use the tilde, "~", symbol at the end of a Single word Term. For example to search for a term similar in spelling to "roam" use the fuzzy search:

```
roam~
```

This search will find terms like foam and roams.

An additional (optional) parameter can specify the maximum number of edits allowed. The value is between 0 and 2, For example:

```
roam~1
```

The default that is used if the parameter is not given is 2 edit distances.

Proximity Searches

Logsene supports finding words are a within a specific distance away. To do a proximity search use the tilde, "~", symbol at the end of a Phrase. For example to search for a "database" and "error" within 10 words of each other in an event use the search:

```
"database error"~10
```

Range Searches

Range Queries allow one to match events whose field(s) values are between the lower and upper bound specified by the Range Query. Range Queries can be inclusive or exclusive of the upper and lower bounds. Sorting is done lexicographically.

```
timestamp:[20020101 TO 20030101]
```

This will find events whose timestamp fields have values between 20020101 and 20030101, inclusive. Note that Range Queries are not reserved for date fields. You could also use range queries with non-date fields:

```
customerName:{Adrian TO Brian}
```

This will find all entries whose customerName is between Adrian and Brian, but not including Adrian and Brian.

Inclusive range queries are denoted by square brackets. Exclusive range queries are denoted by curly brackets.

Boolean Operators

Boolean operators allow terms to be combined through logic operators. Logsene supports AND, "+", OR, NOT and "-" as Boolean operators(Note: Boolean operators must be ALL CAPS).

OR

The OR operator is the default conjunction operator. This means that if there is no Boolean operator between two terms, the OR operator is used. The OR operator links two terms and finds a matching event if either of the terms exist in the event. This is equivalent to a union using sets. The symbol || can be used in place of the word OR.

To search for events that contain either "error or"exception" use the query:

```
error OR exception
```

AND

The AND operator matches events where both terms exist anywhere in the text of a single event. This is equivalent to an intersection using sets. The symbol && can be used in place of the word AND.

To search for events that contain “database connection” and “timeout” use the query:

```
"database connection" AND timeout
```

+

The “+” or required operator requires that the term after the “+” symbol exist somewhere in a the field of a single event.

To search for events that must contain “database” and “connect” use the query:

```
+database +connect
```

NOT

The NOT operator excludes events that contain the term after NOT. This is equivalent to a difference using sets. The symbol ! can be used in place of the word NOT.

To search for events that contain “database connection” but not “failed to” use the query:

```
"database connection" NOT "failed to"
```

Note: The NOT operator cannot be used with just one term. For example, the following search will return no results:

```
NOT "failed to"
```

-

The “-” or prohibit operator excludes events that contain the term after the “-” symbol.

To search for events that contain “database connection” but not “failed to” use the query:

```
+"database connection" -"failed to"
```

Grouping

Logsene supports using parentheses to group clauses to form sub queries. This can be very useful if you want to control the boolean logic for a query.

To search for either “database” or “solr” and “error” use the query:

```
(database OR solr) AND error
```

This eliminates any confusion and makes sure that text “error” must exist and either term “database” or “solr” exist.

Field Grouping

Logsene supports using parentheses to group multiple clauses to a single field.

To search for a title that contains both the word “closed” and the phrase “database connection” use the query:

```
message:(+closed +"database connection")
```

Escaping Special Characters

Logsene supports escaping special characters that are part of the query syntax. The current list special characters are

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

To escape these character use the \ before the character. For example to search for (1+1):2 use the query:

```
\(1\+1\)\:2
```