search-templates.md

github.com/SumoLogic/sumologic-documentation/blob/7b53abb04b0e84a5189b3ef4320a95c27c2a5a2f/docs/search/get-started-with-search/build-search/search-templates.md

---

id: search-templates

title: Search Templates

description: Search templates narrow down your queries into a few parameters that other users can edit to find the data they need.

---

import useBaseUrl from '@docusaurus/useBaseUrl';

import Iframe from 'react-iframe';

Search templates can help you simplify searches for your users by giving them a few easy input choices. You can have search templates replace any text in a query, including fields, keywords, and arguments to operators. You can also determine what type of information is valid such as text, strings, and keywords.

Behind the scenes, selecting the parts of your query to use in the template is also pretty easy. You can select which parts of your search should be available and click **Create Parameter**.

Search templates work with [lookup (classic)](/docs/search/search-query-language/search-operators/lookup-classic). They are not supported with our newer [lookup tables](/docs/search/lookup-tables).

:::sumo Micro Lesson

Watch this micro lesson to learn how to use search template parameters.

<Iframe url="https://fast.wistia.net/embed/iframe/ja6kruhhp3?web_component=true&seo=true&videoFoam=false"

width="854px"

height="480px"

title="Micro Lesson: Using Search Template Parameters Video"

id="wistiaVideo"

className="video-container"

display="initial"

position="relative"

allow="autoplay; fullscreen"

allowfullscreen

/>

:::

## Create a general Search Template

From any query you create, or an existing one you manage, you can create a search template and specify parameters.

1. Open your query.

1. Highlight the field, argument, or operator you want to replace and click **Create a parameter** or **alt+v** if you want to use the keyboard shortcut. <br/><img src={useBaseUrl('img/search/get-started-search/build-search/search-templates/template-variable-selection.png')} alt="Template variable selection" style={{border: '1px solid gray'}} width="800" />

:::note

You can create a maximum of 10 parameters inside a search.

:::

1. From the **Manage Parameter Settings** dialog, provide the name, available values separated by commas, data type, and a brief description. There are four data types to choose, based on how you want to define a valid parameter:

Data Type	Description
Number	Numbers only.
String	Considers as a single phrase and will wrap in double-quotes. For example “system errors”.
Any	All characters. Best for substituting texts in paths.
Keyword	Any Sumo Logic keyword. There are some performance benefits to using Sumo Logic keywords so this is a great option to choose if you can.

1. Optionally, you can set autocomplete values for your parameter by selecting **Set Values for Parameter**. Select a format:

1. For text entries, enter each value on a separate line. Do not use commas to separate values as they will be marked invalid. If the string needs a comma, use quotes in the text entry, such as “abc,xyz”.

1. For Label-Value pairs, copy paste the label-value pairs as comma-delimited lines. If you're using a Lookup make sure that you are using a valid [lookup (classic)](/docs/search/search-query-language/search-operators/lookup-classic) file because the system will reject any lookup file path that it cannot validate.<br/><img src={useBaseUrl('img/search/get-started-search/build-search/search-templates/label-value-pairs.png')} alt="Label value pairs" style={{border: '1px solid gray'}} width="500" />

1. Select the appropriate values for the field, such as user ID.

1. Select values for the corresponding label, such as name. Both **Text** entries and **Label-value** pairs allow amaximum of 10,000 entries. A lookup file can have a maximum of 40,000 entries.

1. For a lookup file, you must enter a valid [lookup (classic)](/docs/search/search-query-language/search-operators/lookup-classic) file that you have [saved](/docs/search/search-query-language/search-operators/save-classic).

1. Under **Select a format**, select **Lookup**.

1. Enter in a valid lookup file or select a shared lookup file from the dropdown.

1. Select values for the corresponding label, such as name. <br/><img src={useBaseUrl('img/search/get-started-search/build-search/search-templates/autocomplete-values.png')} alt="Autocomplete values" style={{border: '1px solid gray'}} width="600" />

1. Click **Save.**

1. Share your search with any new users by clicking **Share** underneath your query window.
1. Grant **Edit** access to the users and roles that should use this search template.

### Create a Search Template for customer IDs

If you want to simplify a user search with a template, you can use label-value pairs to associate human-readable labels such as customer names with machine-understandable values, customer IDs. This association allows the user to search by a known category (customer name) instead of a more abstract and harder to memorize value such as a customer ID.

For example, if we use the following sample query on how to locate users by IP addresses:

```sql

_sourceCategory=service "Successful login from UI"

| parse "[auth=User:*:*:*] [remote_ip=*]" as user,user_id,g,remote_ip

| where user_id matches joeX

| lookup city, region, country_name ,latitude, longitude from geo://location on ip=remote_ip

| where region matches CA

| count by latitude, longitude, user

```

For example, your team could use this query to locate “time-travelers”, suspicious users who log in from two different geographical areas in an impossibly short amount of time, like New York and California within a timerange of the last 15 minutes. These users do not want to modify the query or look up user ids to user names. With parameters we can make a search template that will give your users the dropdowns they need quickly.

First create parameters for `user_name` and `state` to eliminate the manual entries joeX and CA:

```sql
```

```
_sourceCategory=service "Successful login from UI"
| parse "[auth=User:*:*:*] [remote_ip=*]" as user,user_id,g,remote_ip
| where user_id matches {{user_name}}
| lookup city, region, country_name ,latitude, longitude from geo://location on ip=remote_ip
| where region matches {{state}}
| count by latitude, longitude, user
```

Next, specify the `user_name` parameter as a lookup that already has the association between our user names and our user IDs, in this case `/shared/angad/user_info_lookup`:

1. Enter in a valid [lookup (classic)](/docs/search/search-query-language/search-operators/lookup-classic) file that you have [saved](/docs/search/search-query-language/search-operators/save-classic) with the save operator. <br/><img src={useBaseUrl('img/search/get-started-search/build-search/search-templates/lookup.png')} alt="Lookup" style={{border: '1px solid gray'}} width="600" />

1. Select the appropriate values for the field, such as user ID.

1. Select values for the corresponding label, such as name.

1. Click **Save**.

1. Share your search with any new users as needed.

### Create a Search Template for timeslice

Let’s take a simple query:

```sql
_sourceCategory=apache_error
| timeslice 1m
| count by _timeslice
```

Yet even this simple query requires users to know about the query language, the use of pipes and fields in Sumo. You can specify the argument to timeslice as a parameter:

```sql
_sourceCategory=apache_error
| timeslice {{parameter}}m
| count by _timeslice
```

Make sure you specify the right data type for timeslice, Number. You do not want users to input a string.

The parameter is now available for your users to modify as they want, with any input value for the timeslice. Given that our search is a 15 minute time range, values over 5 are probably not useful.
Optional. To give your users even more flexibility letting them append seconds or minutes at the end of the time slice, make the timeslice parameter type **Any**. Users can then add values such as 15s or 5m, or even longer values of 1h, 1d, or 1w.

```sql
_sourceCategory=apache_error
| timeslice {{timeslice}}
| count by _timeslice
```

<img src={useBaseUrl('img/search/get-started-search/build-search/search-templates/searchtemplates.png')} alt="Search Templates" style={{border: '1px solid gray'}} width="500" />

Save. You now have a parameter for your search that allows users to just pick the timeslice from a list of values you feel is appropriate for the query.

## Rename a parameter

If you need to rename a parameter, you can do that from the Manage Parameters Settings dialog.

1. Click the details icon for the parameter setting and select **Manage Parameter Settings**.

1. From the **Manage Parameter Settings** dialog, edit the Parameter Name field. You must specify a valid name with no spaces or special characters, except for underscores.

1. Click **Save**.

## Delete a Search Template

If you want to delete a search template, delete the existing parameters. When you delete the last remaining parameter, you will no longer see the template view.

1. Select the details icon of your last parameter.

1. Select **Delete Parameter**.

When you delete a String parameter, the default value is substituted back into the query in double-quotes. For example “user_name”. If you did not specify a default value for this parameter, you must specify one now before re-running the query. You may also need to remove the quotes.

## Best Practices

The important thing to remember when using search templates is to carefully consider which parameters in your queries users need to change most.

Here are some of the best candidates for parameters in your search templates.

### Parameters

* **Keyword**. Let your users supply keywords in the query. In the example below, we have created a parameter to specify the keyword for a given query. The parameter data type should be string or keyword.

* **Value for Built in fields or FER fields.** Specify a value for built-in fields or FER fields. For example, let your users specify a `_sourceCategory` dynamically.

* **Arguments to an operator.** A parameter can be used to specify an argument to an operator. For example, you can use a parameter to specify a value to the timeslice operator. Make sure that the parameter data type corresponds with the operator argument supplied or the query will send an error to your users.

### Data types

There are some best practices for working with data types.

* If you have a parameter in the search expression, then you should choose Keywords because it makes the queries run faster.

* If you want to parameterize arguments to an operator then you should use the data types Any or Number, depending on what arguments the user needs to supply in the search.

* If you want to parameterize numeric arguments to an operator such as timeslice then you should always use the Number data type because there is a check to validate the right data type is being used. The **Any** data type does not support any validation so people can use it in unsupported values.

* If you want to parameterize a part of a path, for example, `/mysearch/{{parameter}}/k8s/api`, you must use the **Any** Input type.