Splunk® Enterprise
Splunk Dashboard Studio 9.1.1

Escapes, quotation marks, and deltas in dynamic options syntax

Generated: 10/08/2023 12:43 pm

Escapes, quotation marks, and deltas in dynamic options syntax

When you use dynamic options syntax (DOS), there are formatting rules that you must follow. See the following sections to learn more about these rules:

- Escaping characters in DOS
- Quotations marks in DOS
- Delta function in DOS

Escaping characters in DOS

Dashboard definitions in Splunk Dashboard Studio are formatted using JSON. Because of JSON formatting, when you use certain characters in DOS, you need to escape them. When you escape a character, you indicate that the character you've escaped is part of the string's value, and not a JSON keyword.

In JSON, you escape a character by inserting a forward slash in front of the character you'd like to escape. See the following table for an example of escaping a character:

Line without escaped character	Line with escaped character
`"> table | seriesByName("Purchases")"`	`"> table | seriesByName(\"Purchases\")"`

The following table lists all the characters you must escape in JSON:

Character	Definition
"	Double quotes
\	Backslash
/	Forward slash
\b	Backspace
\f	Form feed
\n	Newline
\r	Carriage return
\t	Tab

Quotation marks in DOS

When you input a value in a DOS selector function or a DOS formatting function, you must wrap certain values in quotation marks. The following table shows the types of inputs you can use in a DOS selector or formatting function and whether you need to wrap quotation marks around the input. If you need to wrap quotation marks around the input, you must escape the quotation marks for JSON to properly parse your line of DOS. See Escaping characters in DOS for more information.

Type of input	Quotation marks necessary?	Example
Strings	Yes	`"> table | seriesByName(\"Purchases\")"`
Number	No	`"> sparklineValues | delta(-2)"`

Type of input	Quotation marks necessary?	Example
JSON object in the context section of dashboard definition	No	`"> trendValue | rangeValue(trendColorEditorConfig)"`

Instead of escaping a double quotation mark in DOS, you can also use single quotes to wrap your input for a selector function. See the following table for an example of using single quotes compared to double quotes in DOS:

Single quotes	Double quotes
`"> table | seriesByName('Purchases')"`	`"> table | seriesByName(\"Purchases\")"`

To learn more about selector and formatting functions in DOS, see Selector and formatting functions.

Delta function in DOS

The delta function in DOS is a selector function. A selector function identifies and selects data associated with the visualization. The `delta(index)` function calculates the difference between the point at the index you pass into the delta function and the last point in the dataset.

You can pass in both positive and negative values to the delta function. A negative value indicates an offset from the end of the dataset. For example, `delta(-2)` calculates the difference between the second-to-last number in the dataset and the last number in the dataset.

In the following example, the delta function calculates the trend value in the dashboard based on the command `delta(-2)`, which takes the difference between the second-to-last point in the dataset and the last point in the dataset.

When you pass in a positive value to the delta function, the value of the positive number refers to its numerical index in the dataset. When you use the delta function, indices start at the value 0. For example, the `delta(0)` command calculates the difference between the first value in the dataset and the last value in the dataset. The `delta(2)` command calculates the difference between the third value in the dataset and the last value in the dataset.