+++ title = "Advanced variable format options" keywords = ["grafana", "templating", "documentation", "guide", "template", "variable"] weight = 600 +++

Advanced variable format options

The formatting of the variable interpolation depends on the data source, but there are some situations where you might want to change the default formatting.

For example, the default for the MySql data source is to join multiple values as comma-separated with quotes: `'server01','server02'`. In some cases, you might want to have a comma-separated string without quotes: `server01,server02`. You can make that happen with advanced variable formatting options listed below.

General syntax

Syntax: `${var_name:option}`

Test the formatting options on the Grafana Play site.

If any invalid formatting option is specified, then `glob` is the default/fallback option.

An alternative syntax (that might be deprecated in the future) is `[[var_name:option]]`.

CSV

Formats variables with multiple values as a comma-separated string.

```
servers = ['test1', 'test2']
String to interpolate: '${servers:csv}'
Interpolation result: 'test1,test2'
```

Distributed - OpenTSDB

Formats variables with multiple values in custom format for OpenTSDB.

```
servers = ['test1', 'test2']
String to interpolate: '${servers:distributed}'
Interpolation result: 'test1,servers=test2'
```

Doublequote

Formats single- and multi-valued variables into a comma-separated string, escapes `"` in each value by `\"` and quotes each value with `"`.

```
servers = ['test1', 'test2']
String to interpolate: '${servers:doublequote}'
Interpolation result: '"test1","test2"'
```

Glob - Graphite

Formats variables with multiple values into a glob (for Graphite queries).

```
servers = ['test1', 'test2']
String to interpolate: '${servers:glob}'
Interpolation result: '{test1,test2}'
```

JSON

Formats variables with multiple values as a comma-separated string.

```
servers = ['test1', 'test2']
String to interpolate: '${servers:json}'
Interpolation result: '["test1", "test2"]'
```

Lucene - Elasticsearch

Formats variables with multiple values in Lucene format for Elasticsearch.

```
servers = ['test1', 'test2']
String to interpolate: '${servers:lucene}'
Interpolation result: '("test1" OR "test2")'
```

Percentencode

Formats single and multi valued variables for use in URL parameters.

```
servers = ['foo()bar BAZ', 'test2']
String to interpolate: '${servers:percentencode}'
Interpolation result: 'foo%28%29bar%20BAZ%2Ctest2'
```

Pipe

Formats variables with multiple values into a pipe-separated string.

```
servers = ['test1.', 'test2']
String to interpolate: '${servers:pipe}'
Interpolation result: 'test1.|test2'
```

Raw

Turns off data source-specific formatting, such as single quotes in an SQL query.

```
servers = ['test.1', 'test2']
String to interpolate: '${var_name:raw}'
Interpolation result: 'test.1,test2'
```

Regex

Formats variables with multiple values into a regex string.

```
servers = ['test1.', 'test2']
String to interpolate: '${servers:regex}'
Interpolation result: '(test1\.|test2)'
```

Singlequote

Formats single- and multi-valued variables into a comma-separated string, escapes `'` in each value by `\'` and quotes each value with `'`.

```
servers = ['test1', 'test2']
String to interpolate: '${servers:singlequote}'
Interpolation result: "'test1','test2'"
```

Sqlstring

Formats single- and multi-valued variables into a comma-separated string, escapes `'` in each value by `''` and quotes each value with `'`.

```
servers = ["test'1", "test2"]
String to interpolate: '${servers:sqlstring}'
Interpolation result: "'test''1','test2'"
```

Text

Formats single- and multi-valued variables into their text representation. For a single variable it will just return the text representation. For multi-valued variables it will return the text representation combined with `+`.

```
servers = ["test1", "test2"]
String to interpolate: '${servers:text}'
Interpolation result: "test1 + test2"
```

Query parameters

Formats single- and multi-valued variables into their query parameter representation. Example: `var-foo=value1&var-foo=value2`

```
servers = ["test1", "test2"]
String to interpolate: '${servers:queryparam}'
Interpolation result: "var-servers=test1&var-servers=test2"
```