Variable syntax

- Advanced variable format options
  - General syntax
  - CSV
  - Distributed - OpenTSDB
  - doublequote
  - Glob - Graphite
  - Join
  - JSON
  - Lucene - Elasticsearch
  - percentencode
  - Pipe
  - Query parameters
  - Raw
  - Regex
  - singlequote
  - sqlstring
  - Text

Panel titles and metric queries can refer to variables using two different syntaxes:

- `$varname` This syntax is easy to read, but it doesn't allow you to use a variable in the middle of a word. **Example:** apps.frontend.$server.requests.count
- `${var_name}` Use this syntax when you want to interpolate a variable in the middle of an expression.
- `${var_name:<format>}` This format gives you more control over how Grafana interpolates values. Refer to Advanced variable format options for more detail on all the formatting types.
- `[[varname]]` Do not use. Deprecated syntax, that will be removed in a future release.

Before queries are sent to your data source, the query is *interpolated*, meaning the variable is replaced with its current value. During interpolation, the variable value might be *escaped* to conform to the syntax of the query language and where it's used. For example, variables in

InfluxDB or Prometheus regular expression queries are automatically escaped. Read the data source specific documentation topic for details on value escaping during interpolation.

For advanced syntax to override data source default formatting, refer to Advanced variable format options.

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}'
```

Join

Formats multi-valued variables with a custom delimiter. If no delimiter argument is supplied, they're combined with `,`.

```
servers = ["test1", "test2"]
String to interpolate: '${servers:join:&}'
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'
```

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"
```

Use the `customqueryparam` formatter to customize how the query parameters are formatted. It accepts two optional arguments to specify the parameter name, and a value prefix.

```
servers = ["test1", "test2"]
String to interpolate: '${servers:customqueryparam:v-servers:x-}'
```

```
Interpolation result: "v-servers=x-test1&v-servers=x-test2"
```

Raw

Doesn't apply any data source-specific formatting to the variable.

For example, in this case, there's a dashboard with a Prometheus data source and a multi-value variable. Grafana typically converts the variable values as follows to accommodate Prometheus:

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

Using the raw format, the values are returned without that formatting:

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

Regex

Formats variables with multiple values into a regular expression 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 just returns the text representation. For multi-valued variables it returns the text representation combined with `+`.

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

□□□ #2
□ □□□ □□□ 2026-10-31 16:32:54 UTC
□ □□□ □□□ 2026-10-31 16:32:56 UTC