

Simple LogQL simulator



- [Simple LogQL simulator](#)
 - [Additional sample queries](#)
 - [Logfmt](#)
 - [JSON](#)
 - [Unstructured text](#)
-

Simple LogQL simulator

The LogQL simulator is an online tool that you can use to experiment with writing simple LogQL queries and seeing the results, without needing to run an instance of Loki.

A set of example log lines are included for each of the primary log parsers supported by Loki:

- [Logfmt](#)
- [JSON](#)
- Unstructured text, which can be parsed with the Loki pattern or regex parsers

The [log stream selector](#) `{job="analyze"}` is shown as an example, and it remains fixed for all possible example queries in the simulator. A log stream is a set of logs which share the same labels. In LogQL, you use a log stream selector to determine which log streams to include in a query's results.

{{< admonition type="note" >}} This is a very limited simulator, primarily for evaluating filters and parsers. If you want to practice writing more complex queries, such as metric queries, you can use the [Explore](#) feature in Grafana. {{< /admonition >}}

To use the LogQL simulator:

1. Select a log line format using the radio buttons.
2. You can use the provided example log lines, or copy and paste your own log lines into the example log lines box.

3. Use the provided example LogQL query, or enter your own query. The [log stream selector](#) remains fixed for all possible example queries. There are additional sample queries at the end of this topic.
4. Click the **Run query** button to run the entered query against the example log lines.

The results output simulates how Loki would return results for your query. You can also click each line in the results pane to expand the details, which give an explanation for why the log line is or is not included in the query result set.

Log line format: logfmt JSON Unstructured text

Link copied to clipboard.

```
{job="analyze"}
```

Query:

```
{job="analyze"}
```

Additional sample queries

These are some additional sample queries that you can use in the LogQL simulator.

Logfmt

```
| logfmt | level = "debug"
```

Parses logfmt-formatted logs and returns only log lines where the "level" field is equal to "debug".

```
| logfmt | msg="server listening on addresses"
```

Parses logfmt-formatted logs and returns only log lines with the message “server listening on address.”

JSON

```
| json | level="INFO" | file="SpringApplication.java" | line_format `{{.class}}`
```

Parses JSON-formatted logs, filtering for lines where the 'level' field is "INFO" and the 'file' field is "SpringApplication.java", then formats the line to return only the 'class' field.

```
|~ `(T|t)omcat`
```

Performs a regular expression filter for the string 'tomcat' or 'Tomcat', without using a parser.

Unstructured text

```
| pattern "<_> - <_> <_> \"<method> <url> <protocol>\" <status> <_> <_> \"<_>\" <_>" |  
method="GET"
```

Parses unstructured logs with the pattern parser, filtering for lines where the HTTP method is "GET".

```
| pattern "<_> - <user> <_> \"<method> <url> <protocol>\" <status> <_> <_> \"<_>\" <_>" |  
user=~"kling.*"
```

Parses unstructured logs with the pattern parser, extracting the 'user' field, and filtering for lines where the user field starts with "kling".

[] [] #2	
[] [] []	2026-10-31 13:47:37 UTC
[] [] []	2026-10-31 13:47:38 UTC