TestData query editor

- Scenario reference
- Shared query options
- Data generation scenarios
  - Random Walk
  - Random Walk Table
  - Random Walk (with error)
  - Predictable Pulse
  - Predictable CSV Wave
  - Simulation
  - USA generated data
- Manual input scenarios
  - CSV Content
  - Steps
  - CSV File
  - CSV Metric Values
  - Raw Frames
  - Load Apache Arrow Data
  - Table Static
- Visualization test scenarios
  - Logs
  - Node Graph
  - Flame Graph
  - Trace
  - Annotations
  - Exponential heatmap bucket data
  - Linear heatmap bucket data
  - Exemplars
- Streaming scenarios

    - Streaming Client
    - Grafana Live
- Data retrieval scenarios
    - Grafana API
- Error and edge-case testing scenarios
    - Conditional Error
    - Error with source
    - Flaky Query
    - Errors and notices
    - No Data Points
    - Datapoints Outside Range
    - Slow Query
- Metadata scenarios
    - Query Metadata

Instead of a traditional query language, the TestData data source uses **scenarios** to generate simulated data. Each scenario produces a different type of data suited for testing specific visualizations, behaviors, or edge cases. TestData includes 33 scenarios covering time series, logs, traces, graphs, streaming, and error simulation.

Use scenarios to:

- **Prototype dashboards** without connecting to a real data source.
- **Reproduce bugs** with controlled, deterministic data that other developers can replicate.
- **Test panel behavior** with edge cases like empty results, timestamps outside the visible range, or mixed data and errors.
- **Validate alerting pipelines** using predictable patterns that fire and resolve on a known schedule.
- **Simulate streaming** to verify how panels handle real-time data updates.

To build a query, select a scenario from the **Scenario** drop-down. The query editor updates to show fields specific to that scenario. Click **Run queries** or use the keyboard shortcut to execute.

{{< admonition type="note" >}} Some scenarios run entirely in the browser: Streaming Client, Grafana Live, Grafana API, Steps, Annotations, Node Graph, Flame Graph, Trace, and Raw Frames. These scenarios don't send queries to the backend, which means they can't be used with Grafana Alerting or in any context that requires server-side evaluation. {{< /admonition >}}

Scenario reference

The scenarios are organized into the following categories. Use the table below to find the right scenario for your use case.

Scenario	Category	Purpose
Random Walk	Data generation	Random walk time series (default scenario).
Random Walk Table	Data generation	Random walk in table format with `Enum` state.
Random Walk (with error)	Data generation	Random walk that also returns an error.
Predictable Pulse	Data generation	Repeating on/off wave based on absolute time.
Predictable CSV Wave	Data generation	Custom repeating waveforms from CSV values.
Simulation	Data generation	Continuous simulation engine (flight, sine, tank).
USA generated data	Data generation	Multi-dimensional data with US state dimensions.
CSV Content	Manual input	Paste or type CSV data directly.
Steps	Manual input	Step-function data from CSV input.
CSV File	Manual input	Select from built-in CSV data files.
CSV Metric Values	Manual input	Time series from comma-separated values.
Raw Frames	Manual input	Define data frames in JSON format.
Load Apache Arrow Data	Manual input	Render base64-encoded Arrow payloads.
Table Static	Manual input	Static table with predefined columns.
Logs	Visualization testing	Simulated log data with random levels.
Node Graph	Visualization testing	Data for node graph visualizations.
Flame Graph	Visualization testing	Data for flame graph visualizations.
Trace	Visualization testing	Simulated distributed trace data.
Annotations	Visualization testing	Annotation data points.

Scenario	Category	Purpose
Exponential heatmap bucket data	Visualization testing	Heatmap data with exponential buckets.
Linear heatmap bucket data	Visualization testing	Heatmap data with linear buckets.
Exemplars	Visualization testing	Exemplar markers to overlay on a sibling series.
Streaming Client	Streaming	Browser-side streaming (signal, logs, traces).
Grafana Live	Streaming	Server-side streaming via live channels.
Grafana API	Data retrieval	Fetch data from internal Grafana endpoints.
Conditional Error	Error testing	Configurable error or CSV data.
Error with source	Error testing	Error with plugin or downstream classification.
Flaky Query	Error testing	Intermittent errors with configurable rate.
Errors and notices	Error testing	Series with info, warning, and error notices.
No Data Points	Error testing	Empty result with no data.
Datapoints Outside Range	Error testing	Data point outside the visible time range.
Slow Query	Error testing	Configurable delay before returning data.
Query Metadata	Metadata	Returns query context metadata.

Shared query options

Every scenario displays the **Scenario** drop-down. Most scenarios also show an **Alias** field and, when the scenario defines a default, a **String Input** field. Some scenarios expose additional shared controls. Scenario-specific options appear below these fields.

Field	Description
Scenario	Drop-down to select the data generation type. Scenarios are listed alphabetically.

Field	Description
Alias	Optional display name for the series. Hidden for the Simulation and Annotations scenarios. Supports the special values `__server_names` and `__house_locations` to use built-in name sets.
String Input	Text field shown when the selected scenario provides a default string value. Used differently by each scenario.
Labels	Available for Random Walk and Predictable Pulse. Set labels using key=value syntax, for example `key="value", key2="value2"`. Use `$seriesIndex` in a value to insert the series index.
Drop percent	Available for CSV Content and CSV File. Drops a random percentage (0-100) of data points from the result.

Data generation scenarios

These scenarios produce time-series data with configurable parameters.

Random Walk

Generates random walk time-series data. This is the default scenario.

Field	Description
Series count	Number of series to generate. Default: `1`.
Start value	Initial value for the walk. Default: auto (random).
Min	Minimum value the walk can reach. Default: none.
Max	Maximum value the walk can reach. Default: none.
Spread	Controls how far each step can deviate. Default: `1`.
Noise	Adds noise to each data point. Default: `0`.
Drop (%)	Percentage of points to randomly exclude. Default: `0`.

Random Walk Table

Generates random walk data in table format with columns for Time, Value, Min, Max, Info, and State. The State column uses an `enum` field with values Unknown, Up, and Down. Optionally includes null values when **withNil** is enabled.

Random Walk (with error)

Generates random walk time-series data and also returns an error in the response. Use this to test how panels handle data responses that contain both data and errors.

Predictable Pulse

Generates a predictable pulse wave based on absolute time from the epoch, making it reproducible across runs.

Field	Description
Step	Seconds between data points. Default: `60`.
On Count	Number of data points at the start of each cycle that use the on value. Default: `3`.
Off Count	Number of data points in each cycle that use the off value. Default: `3`.
On Value	The value for "on" data points. Can be a number, `null`, or `nan`. Default: `2`.
Off Value	The value for "off" data points. Can be a number, `null`, or `nan`. Default: `1`.

The wave cycles at `Step * (On Count + Off Count)` seconds. Timestamps align evenly on the step interval.

With the defaults (Step=60, On Count=3, Off Count=3), the wave completes a full cycle every 6 minutes. The on value (`2`) holds for the first 3 data points, then the off value (`1`) holds for the next 3.

Predictable CSV Wave

Generates one or more predictable waves from CSV-defined values. Each wave cycles through its comma-separated values at a fixed time step.

Field	Description
Values	Comma-separated numeric values for the wave. Supports `null` and `nan`. Default: `0,0,2,2,1,1`.
Step	Seconds between data points. Default: `60`.
Name	Optional name for the series.
Labels	Optional labels in key=value format.

Click the **+** button on the last wave row to add another wave. Click the **-** button on any other row to remove it.

For example, to create two overlapping waves for comparison:

- Wave 1: Values `0,1,2,3,4,5`, Step `60`, Name `rising`
- Wave 2: Values `5,4,3,2,1,0`, Step `60`, Name `falling`

Simulation

Runs a simulation engine that generates data continuously. Simulation supports streaming data through Grafana Live.

Field	Description
Simulation	Simulation type: `Flight` (circling flight path), `Sine` (sine wave), or `Tank` (fill and drain a water tank).
Stream	Toggle to stream data through Grafana Live instead of returning a static result.
Interval	Tick frequency in Hz. Default: `10`.
Last	Toggle to return only the most recent value instead of the full time range.
UID	Optional unique identifier. Allows multiple instances of the same simulation type to run concurrently.

USA generated data

Generates data with US state dimensions. Useful for testing geo-map visualizations and multi-dimensional data.

Field	Description
Mode	Data format: `values-as-rows`, `values-as-fields`, `values-as-labeled-fields`, `timeseries`, or `timeseries-wide`.
Period	Time period for data generation. Default: `30m`.
Fields	Fields to include: `foo`, `bar`, `baz`. Default: all.
States	US state codes to include (for example, `CA`, `NY`, `TX`). Default: all 50 states plus DC.

Manual input scenarios

These scenarios let you provide your own data directly instead of generating it.

CSV Content

Provides a text editor where you paste or type CSV data directly. The first row is treated as headers. Use the **Drop percent** field to randomly exclude a percentage of data points.

For example, enter the following to create a time series with two value columns:

```
Time,Temperature,Humidity
2024-01-01 00:00:00,22.5,45
2024-01-01 01:00:00,21.8,48
2024-01-01 02:00:00,20.1,52
```

Steps

Provides a CSV text area (the same editor as CSV Content) for defining step-function data. The default content is `a`, `b`, `c`. This scenario is handled entirely in the browser.

CSV File

Selects from a set of built-in CSV data files. Use the **Drop percent** field to randomly exclude a percentage of data points.

Available files:

- `flight_info_by_state.csv`
- `population_by_state.csv`
- `gdp_per_capita.csv`
- `js_libraries.csv`
- `ohlc_dogecoin.csv`
- `weight_height.csv`
- `browser_marketshare.csv`
- `automobiles.csv`

CSV Metric Values

Generates time-series data from comma-separated values entered in the **String Input** field. Values are evenly distributed across the selected time range. Default: `1,20,90,30,5,0`.

For example, with the default values and a 6-hour time range, the six values (`1`, `20`, `90`, `30`, `5`, `0`) are spread across the range at equal intervals.

Raw Frames

Provides a JSON editor for defining data frames directly. The editor accepts the Grafana data frame JSON format and includes paste helpers for panel JSON and raw query results.

Load Apache Arrow Data

Loads data from a base64-encoded Apache Arrow payload entered in the **String Input** text area. Use this to test Arrow format rendering.

Table Static

Generates a static table with predefined columns: Time, Message, Description, and Value.

Visualization test scenarios

These scenarios generate data for testing specific visualization types.

Logs

Generates simulated log data with random log levels, container IDs, and hostnames.

Field	Description
Lines	Number of log lines to generate. Default: `10`. Maximum: `10000`.
Level	Toggle to include a separate level column in the data instead of embedding it in messages.

Node Graph

Generates data for the Node Graph visualization.

Field	Description

Data type	Type of graph data: `random`, `response_small`, `response_medium`, `random edges`, or `feature_showcase`.
Count	Number of nodes to generate. Available for `random` and `random edges` types.
Seed	Seed value for reproducible random generation. Available for `random` and `random edges` types.

Flame Graph

Generates data for the Flame Graph visualization.

Field	Description
Diff profile	Toggle to generate a differential flame graph comparing two profiles.

Trace

Generates simulated distributed trace data.

Field	Description
Span count	Number of spans in the generated trace.

Annotations

Generates synthetic annotation data points. These annotations are created entirely in the browser and are useful for testing how panels display annotation markers and overlays. They don't query an external source or persist data.

For more information about using annotations with TestData, refer to TestData annotations. For more information about annotations in Grafana, refer to Annotate visualizations.

Field	Description
Count	Number of annotations to generate. Default: `10`.

Exponential heatmap bucket data

Generates heatmap data with exponentially distributed bucket boundaries (1, 2, 4, 8, 16, ...). Use this to test heatmap panels with exponential distributions.

Linear heatmap bucket data

Generates heatmap data with linearly distributed bucket boundaries (0, 10, 20, 30, ...). Use this to test heatmap panels with linear distributions.

Exemplars

Generates only exemplars, the diamond markers a panel draws on top of a series, and no series of its own. Add a second query in the same panel, such as Random Walk, to give the exemplars a series to annotate. Values and timestamps are seeded from the dashboard time range, so the same range always returns the same exemplars.

Field	Description
Count	Number of exemplars to generate.
Min	Minimum value for the exemplars. When empty, the range is derived from a reference random walk over the same time range, padded by 10%.
Max	Maximum value for the exemplars. When empty, the range is derived as described for **Min**.
Labels	Exemplar labels shown in the tooltip. Each label defines a **Name**, a **Length**, and an optional **Link**.

To keep the exemplars aligned with a sibling series, pin **Min** and **Max**, or give the sibling query an explicit start value and spread. The legend filter matches on exemplar labels, so hiding a series in the legend also hides its markers.

Streaming scenarios

These scenarios produce real-time streaming data. Data updates continuously in the browser without requiring manual query execution.

Streaming Client

Generates streaming data directly from the browser client.

Field	Description

Type	Stream type: `Signal`, `Logs`, `Fetch`, `Traces`, or `Watch`.

Additional fields depend on the selected type:

- **Signal:** Speed (ms), Spread, Noise, Bands.
- **Logs, Traces, Watch:** Speed (ms) only.
- **Fetch:** URL of a remote CSV endpoint to stream incrementally.

Grafana Live

Connects to a Grafana Live channel that streams random data from the server.

Field	Description
Channel	The live channel to subscribe to. Options:

Channel	Description
`random-2s-stream`	Random stream with points every 2s.
`random-flakey-stream`	Stream that returns data at random intervals.
`random-labeled-stream`	Value with moving labels.
`random-20Hz-stream`	Random stream with points at 20 Hz.

Data retrieval scenarios

These scenarios fetch data from internal Grafana endpoints.

Grafana API

Fetches data from internal Grafana API endpoints and returns the result as a data frame. This scenario runs in the browser.

Field	Description
Endpoint	The API endpoint to query. Options:

Endpoint	Description
Data Sources	Lists configured data sources.
Search	Returns dashboard search results.

Endpoint	Description
Annotations	Returns annotations.

Error and edge-case testing scenarios

These scenarios help test how Grafana handles errors, empty data, and slow responses.

Conditional Error

Produces an error or data depending on the **String Input** and **Error type** fields. When the **String Input** field is empty, the scenario triggers the selected error type. When the field contains CSV values (default: `1,20,90,30,5,0`), it returns time-series data like the CSV Metric Values scenario, regardless of the error type.

Field	Description
Error type	Type of error to simulate: `Server panic` (backend panic), `Frontend exception` (uncaught error in the browser), or `Frontend observable` (error emitted through the query observable).

To trigger an error, clear the **String Input** field and select the error type you want to test.

Error with source

Returns an error with a configurable source classification.

Field	Description
Error source	Source of the error: `Plugin` (plugin error) or `Downstream` (downstream service error).

Use this to test how Grafana differentiates between plugin errors and downstream errors in alerting and error handling.

Flaky Query

Returns errors intermittently based on a configurable error rate, with optional delays. Use this to test how panels, alert rules, and error handling behave when a data source fails part of the time.

Field	Description
Error rate	Percentage of requests that return an error. Default: `50`.
Status code	HTTP status code returned with the error. Default: `400`.
Error source	Source of the error: `Plugin` (plugin error) or `Downstream` (downstream service error).
Query delay	Base delay applied to each request in Go duration syntax, for example `1s` or `500ms`. Default: `5s`.
Delay variability	Randomizes the query delay by plus or minus this percentage. For example, `100` on a `1s` delay sleeps between 0 and 2 seconds.
Error message	Message returned with the error. Default: `Flaky query error`.

Errors and notices

Returns a random walk series with one query-result notice of each severity (info, warning, and error) attached to the frame metadata. Use this to preview how a panel displays query errors and notices.

No Data Points

Returns an empty result with no data points. Use this to test how panels display when there's no data.

Datapoints Outside Range

Returns a single data point with a timestamp one hour before the query time range. Use this to test how panels handle data outside the visible range.

Slow Query

Introduces a configurable delay before returning random walk data. Set the delay duration in the **String Input** field using Go duration syntax (for example, `5s`, `1m`, `500ms`). Default: `5s`.

Metadata scenarios

These scenarios return metadata rather than time-series data.

Query Metadata

Returns a table with query metadata including the current user's username. Use this to verify that query context information is available.

□□□ #2
□ □□□ □□□ 2026-10-31 16:04:38 UTC
□ □□□ □□□ 2026-10-31 16:04:45 UTC