
Deprecated plugin. Use Infinity data source plugin instead

Project status

[!CAUTION] This plugin is now **DEPRECATED** and **no longer maintained** by the Grafana team. You can use Grafana Infinity data source plugin as an alternative for connecting to JSON/CSV/XML/GraphQL endpoints. Refer the migration guide here, if you still prefer connecting to existing grafana simple JSON backend server implementation. This deprecation means that this plugin won't receive any feature updates or bug fixes. After 6 months (End of June 2024), the plugin will reach EOL and the repository will be archived. If you are looking for building your own plugin, consider the examples here and documentation here.

Simple JSON Datasource - a generic backend datasource

You can find more documentation about datasource plugins in Grafana's Docs.

This also serves as a living example implementation of a datasource.

Your backend needs to implement 4 urls:

- `/` should return 200 ok. Used for "Test connection" on the datasource config page.
- `/search` used by the find metric options on the query tab in panels.
- `/query` should return metrics based on input.
- `/annotations` should return annotations.

Those two urls are optional:

- `/tag-keys` should return tag keys for ad hoc filters.
- `/tag-values` should return tag values for ad hoc filters.

Installation

To install this plugin using the `grafana-cli` tool:

```
1 sudo grafana-cli plugins install grafana-simple-json-datasource
2 sudo service grafana-server restart
```

See here for more information.

Example backend implementations

- <https://github.com/bergquist/fake-simple-json-datasource>

-
- <https://github.com/smcquay/jsonds>
 - <https://github.com/ContextLogic/eventmaster>
 - <https://gist.github.com/linar-jether/95ff412f9d19fdf5e51293eb0c09b850> (Python/pandas backend)

Query API

Example `timeserie` request

```
1 {
2   "panelId": 1,
3   "range": {
4     "from": "2016-10-31T06:33:44.866Z",
5     "to": "2016-10-31T12:33:44.866Z",
6     "raw": {
7       "from": "now-6h",
8       "to": "now"
9     }
10  },
11  "rangeRaw": {
12    "from": "now-6h",
13    "to": "now"
14  },
15  "interval": "30s",
16  "intervalMs": 30000,
17  "targets": [
18    { "target": "upper_50", "refId": "A", "type": "timeserie" },
19    { "target": "upper_75", "refId": "B", "type": "timeserie" }
20  ],
21  "adhocFilters": [{
22    "key": "City",
23    "operator": "=",
24    "value": "Berlin"
25  }],
26  "format": "json",
27  "maxDataPoints": 550
28 }
```

Example `timeserie` response

```
1 [
2   {
3     "target": "upper_75", // The field being queried for
4     "datapoints": [
5       [622, 1450754160000], // Metric value as a float , unixtimestamp
6                             // in milliseconds
7       [365, 1450754220000]
8     ]
9   }
10 ]
```

```
8   },
9   {
10    "target": "upper_90",
11    "datapoints": [
12      [861, 1450754160000],
13      [767, 1450754220000]
14    ]
15  }
16 ]
```

If the metric selected is `"type": "table"`, an example `table` response:

```
1 [
2   {
3     "columns": [
4       {"text": "Time", "type": "time"},
5       {"text": "Country", "type": "string"},
6       {"text": "Number", "type": "number"}
7     ],
8     "rows": [
9       [1234567, "SE", 123],
10      [1234567, "DE", 231],
11      [1234567, "US", 321]
12    ],
13    "type": "table"
14  }
15 ]
```

Annotation API

The annotation request from the Simple JSON Datasource is a POST request to the `/annotations` endpoint in your datasource. The JSON request body looks like this:

```
1 {
2   "range": {
3     "from": "2016-04-15T13:44:39.070Z",
4     "to": "2016-04-15T14:44:39.070Z"
5   },
6   "rangeRaw": {
7     "from": "now-1h",
8     "to": "now"
9   },
10  "annotation": {
11    "name": "deploy",
12    "datasource": "Simple JSON Datasource",
13    "iconColor": "rgba(255, 96, 96, 1)",
14    "enable": true,
15    "query": "#deploy"
16  }
```

```
17 }
```

Grafana expects a response containing an array of annotation objects in the following format:

```
1 [
2   {
3     annotation: annotation, // The original annotation sent from
        Grafana.
4     time: time, // Time since UNIX Epoch in milliseconds. (required)
5     title: title, // The title for the annotation tooltip. (required)
6     tags: tags, // Tags for the annotation. (optional)
7     text: text // Text for the annotation. (optional)
8   }
9 ]
```

Note: If the datasource is configured to connect directly to the backend, you also need to implement an OPTIONS endpoint at `/annotations` that responds with the correct CORS headers:

```
1 Access-Control-Allow-Headers:accept, content-type
2 Access-Control-Allow-Methods:POST
3 Access-Control-Allow-Origin:*
```

Search API

Example request

```
1 { target: 'upper_50' }
```

The search api can either return an array or map.

Example array response

```
1 ["upper_25","upper_50","upper_75","upper_90","upper_95"]
```

Example map response

```
1 [ { "text" : "upper_25", "value": 1 }, { "text" : "upper_75", "value": 2 }
    ]
```

Tag Keys API

Example request

```
1 { }
```

The tag keys api returns:

```
1 [
2   {"type":"string","text":"City"},
3   {"type":"string","text":"Country"}
4 ]
```

Tag Values API

Example request

```
1 {"key": "City"}
```

The tag values api returns:

```
1 [
2   {'text': 'Eins!'},
3   {'text': 'Zwei'},
4   {'text': 'Drei!'}
5 ]
```

Dev setup

This plugin requires node 6.10.0

```
1 npm install -g yarn
2 yarn install
3 npm run build
```

Changelog

1.4.1 - Fix for query editor to be compatible with Grafana 7+ (broke due to change in Grafana).

1.4.0 - Support for adhoc filters: - added tag-keys + tag-values api - added adHocFilters parameter to query body

1.3.5 - Fix for dropdowns in query editor to allow writing template variables (broke due to change in Grafana).

1.3.4 - Adds support for With Credentials (sends grafana cookies with request) when using Direct mode
- Fix for the typeahead component for metrics dropdown (/ [search](#) endpoint).

1.3.3 - Adds support for basic authentication

1.2.4 - Add support returning sets in the search endpoint

1.2.3 - Allow nested templates in find metric query. #23

1.2.2 - Dont execute hidden queries - Template support for metrics queries - Template support for annotation queries

If using Grafana 2.6

NOTE! for grafana 2.6 please use this version

Copy the data source you want to </public/app/plugins/datasource/>. Then restart grafana-server. The new data source should now be available in the data source type dropdown in the Add Data Source View.