+++ title = "JSON model" keywords = ["grafana", "dashboard", "documentation", "json", "model"] aliases = ["/docs/grafana/latest/reference/dashboard/"] weight = 1200 +++

Dashboard JSON model

A dashboard in Grafana is represented by a JSON object, which stores metadata of its dashboard. Dashboard metadata includes dashboard properties, metadata from panels, template variables, panel queries, etc.

To view the JSON of a dashboard:

1. Navigate to a dashboard.
2. In the top navigation menu, click the **Dashboard settings** (gear) icon.
3. Click **JSON Model**.

JSON fields

When a user creates a new dashboard, a new dashboard JSON object is initialized with the following fields:

> ***Note:*** *In the following JSON, id is shown as null which is the default value assigned to it until a dashboard is saved. Once a dashboard is saved, an integer value is assigned to the* `id` *field.*

```
{
  "id": null,
  "uid": "cLV5GDCkz",
  "title": "New dashboard",
  "tags": [],
  "style": "dark",
  "timezone": "browser",
  "editable": true,
  "hideControls": false,
  "graphTooltip": 1,
  "panels": [],
  "time": {
    "from": "now-6h",
    "to": "now"
  },
  "timepicker": {
    "time_options": [],
    "refresh_intervals": []
  },
  "templating": {
    "list": []
  },
  "annotations": {
    "list": []
  },
  "refresh": "5s",
  "schemaVersion": 17,
  "version": 0,
  "links": []
}
```

Each field in the dashboard JSON is explained below with its usage:

Name	Usage
id	unique numeric identifier for the dashboard. (generated by the db)
uid	unique dashboard identifier that can be generated by anyone. string (8-40)
title	current title of dashboard
tags	tags associated with dashboard, an array of strings
style	theme of dashboard, i.e. `dark` or `light`
timezone	timezone of dashboard, i.e. `utc` or `browser`
editable	whether a dashboard is editable or not
graphTooltip	0 for no shared crosshair or tooltip (default), 1 for shared crosshair, 2 for shared crosshair AND shared tooltip
time	time range for dashboard, i.e. last 6 hours, last 7 days, etc
timepicker	timepicker metadata, see timepicker section for details
templating	templating metadata, see templating section for details
annotations	annotations metadata, see annotations section for details
refresh	auto-refresh interval
schemaVersion	version of the JSON schema (integer), incremented each time a Grafana update brings changes to said schema
version	version of the dashboard (integer), incremented each time the dashboard is updated
panels	panels array, see below for detail.

Panels

Panels are the building blocks of a dashboard. It consists of data source queries, type of graphs, aliases, etc. Panel JSON consists of an array of JSON objects, each representing a different panel. Most of the fields are common for all panels but some fields depend on the panel type. Following is an example of panel JSON of a text panel.

```
"panels": [
  {
    "type": "text",
    "title": "Panel Title",
    "gridPos": {
      "x": 0,
      "y": 0,
      "w": 12,
      "h": 9
    },
    "id": 4,
    "mode": "markdown",
```

```
    "content": "# title"
  }
```

Panel size and position

The gridPos property describes the panel size and position in grid coordinates.

- `w` 1-24 (the width of the dashboard is divided into 24 columns)
- `h` In grid height units, each represents 30 pixels.
- `x` The x position, in same unit as `w`.
- `y` The y position, in same unit as `h`.

The grid has a negative gravity that moves panels up if there is empty space above a panel.

timepicker

```
"timepicker": {
    "collapse": false,
    "enable": true,
    "notice": false,
    "now": true,
    "refresh_intervals": [
      "5s",
      "10s",
      "30s",
      "1m",
      "5m",
      "15m",
      "30m",
      "1h",
      "2h",
      "1d"
    ],
    "status": "Stable",
    "type": "timepicker"
  }
```

Usage of the fields is explained below:

Name	Usage
collapse	whether timepicker is collapsed or not
enable	whether timepicker is enabled or not
notice	TODO
now	TODO
refresh_intervals	TODO
status	TODO

type	TODO

templating

The `templating` field contains an array of template variables with their saved values along with some other metadata, for example:


```
"templating": {
   "enable": true,
   "list": [
     {
       "allFormat": "wildcard",
       "current": {
         "tags": [],
         "text": "prod",
         "value": "prod"
       },
       "datasource": null,
       "includeAll": true,
       "name": "env",
       "options": [
         {
           "selected": false,
           "text": "All",
           "value": "*"
         },
         {
           "selected": false,
           "text": "stage",
           "value": "stage"
         },
         {
           "selected": false,
           "text": "test",
           "value": "test"
         }
       ],
       "query": "tag_values(cpu.utilization.average,env)",
       "refresh": false,
       "type": "query"
     },
     {
       "allFormat": "wildcard",
       "current": {
         "text": "apache",
         "value": "apache"
       },
       "datasource": null,
       "includeAll": false,
       "multi": false,
       "multiFormat": "glob",
```

```
        "name": "app",
        "options": [
          {
            "selected": true,
            "text": "tomcat",
            "value": "tomcat"
          },
          {
            "selected": false,
            "text": "cassandra",
            "value": "cassandra"
          }
        ],
        "query": "tag_values(cpu.utilization.average,app)",
        "refresh": false,
        "regex": "",
        "type": "query"
      }
    ]
  }
```

Usage of the above mentioned fields in the templating section is explained below:

Name	Usage
enable	whether templating is enabled or not
list	an array of objects each representing one template variable
allFormat	format to use while fetching all values from data source, eg: `wildcard`, `glob`, `regex`, `pipe`, etc.
current	shows current selected variable text/value on the dashboard
data source	shows data source for the variables
includeAll	whether all value option is available or not
multi	whether multiple values can be selected or not from variable value list
multiFormat	format to use while fetching timeseries from data source
name	name of variable
options	array of variable text/value pairs available for selection on dashboard
query	data source query used to fetch values for a variable
refresh	TODO
regex	TODO
type	type of variable, i.e. `custom`, `query` or `interval`