

```
+++ title = "Dashboard HTTP API" description = "Grafana Dashboard HTTP
API" keywords = ["grafana", "http", "documentation", "api", "dashboard"]
aliases = ["/docs/grafana/latest/http_api/dashboard/"] +++
```

## Dashboard API

### Identifier (id) vs unique identifier (uid)

The identifier (id) of a dashboard is an auto-incrementing numeric value and is only unique per Grafana install.

The unique identifier (uid) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. It's automatically generated if not provided when creating a dashboard. The uid allows having consistent URLs for accessing dashboards and when syncing dashboards between multiple Grafana installs, see [dashboard provisioning]({{< relref "../administration/provisioning.md#dashboards" >}}) for more information. This means that changing the title of a dashboard will not break any bookmarked links to that dashboard.

The uid can have a maximum length of 40 characters.

### Create / Update dashboard

POST /api/dashboards/db

Creates a new dashboard or updates an existing dashboard. When updating existing dashboards, if you do not define the `folderId` or the `folderUid` property, then the dashboard(s) are moved to the General folder. (You need to define only one property, not both).

#### Example Request for new dashboard:

```
POST /api/dashboards/db HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

```
{
  "dashboard": {
    "id": null,
    "uid": null,
    "title": "Production Overview",
    "tags": [ "templated" ],
    "timezone": "browser",
    "schemaVersion": 16,
    "version": 0,
    "refresh": "25s"
```

```

    },
    "folderId": 0,
    "folderUid": "l3KqBxCMz",
    "message": "Made changes to xyz",
    "overwrite": false
  }
}

```

JSON Body schema:

- **dashboard** – The complete dashboard model, id = null to create a new dashboard.
- **dashboard.id** – id = null to create a new dashboard.
- **dashboard.uid** – Optional unique identifier when creating a dashboard. uid = null will generate a new uid.
- **folderId** – The id of the folder to save the dashboard in.
- **folderUid** – The UID of the folder to save the dashboard in. Overrides the **folderId**.
- **overwrite** – Set to true if you want to overwrite existing dashboard with newer version, same dashboard title in folder or same dashboard uid.
- **message** – Set a commit message for the version history.
- **refresh** – Set the dashboard refresh interval. If this is lower than [the minimum refresh interval]({{< relref "../administration/configuration.md#min\_refresh\_interval">}}), then Grafana will ignore it and will enforce the minimum refresh interval.

For adding or updating an alert rule for a dashboard panel the user should declare a **dashboard.panels.alert** block.

#### Example Request for updating dashboard alert rule:

```

HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 78

```

```

{
  "dashboard": {
    "id": 104,
    "panels": [
      {
        "alert": {
          "alertRuleTags": {},
          "conditions": [
            {
              "evaluator": {
                "params": [
                  25
                ],
                "type": "gt"
              }
            }
          ]
        }
      }
    ]
  }
}

```

```

        },
        "operator": {
            "type": "and"
        },
        "query": {
            "params": [
                "A",
                "5m",
                "now"
            ]
        },
        "reducer": {
            "params": [],
            "type": "avg"
        },
        "type": "query"
    }
],
"executionErrorState": "alerting",
"for": "5m",
"frequency": "1m",
"handler": 1,
"name": "Panel Title alert",
"noDataState": "no_data",
"notifications": []
},
"aliasColors": {},
"bars": false,
"dashLength": 10,
"dashes": false,
"datasource": null,
"fieldConfig": {
    "defaults": {
        "custom": {}
    },
    "overrides": []
},
"fill": 1,
"fillGradient": 0,
"gridPos": {
    "h": 9,
    "w": 12,
    "x": 0,
    "y": 0
},
"hiddenSeries": false,

```

```

    "id": 2,
    "legend": {
      "avg": false,
      "current": false,
      "max": false,
      "min": false,
      "show": true,
      "total": false,
      "values": false
    },
    "lines": true,
    "linewidth": 1,
    "nullPointMode": "null",
    "options": {
      "dataLinks": []
    },
    "percentage": false,
    "pointradius": 2,
    "points": false,
    "renderer": "flot",
    "seriesOverrides": [],
    "spaceLength": 10,
    "stack": false,
    "steppedLine": false,
    "targets": [
      {
        "refId": "A",
        "scenarioId": "random_walk"
      }
    ],
    "thresholds": [
      {
        "colorMode": "critical",
        "fill": true,
        "line": true,
        "op": "gt",
        "value": 50
      }
    ],
    "timeFrom": null,
    "timeRegions": [],
    "timeShift": null,
    "title": "Panel Title",
    "tooltip": {
      "shared": true,
      "sort": 0,

```

```

        "value_type": "individual"
    },
    "type": "graph",
    "xaxis": {
        "buckets": null,
        "mode": "time",
        "name": null,
        "show": true,
        "values": []
    },
    "yaxes": [
        {
            "format": "short",
            "label": null,
            "logBase": 1,
            "max": null,
            "min": null,
            "show": true
        },
        {
            "format": "short",
            "label": null,
            "logBase": 1,
            "max": null,
            "min": null,
            "show": true
        }
    ],
    "yaxis": {
        "align": false,
        "alignLevel": null
    }
}
],
"title": "Update alert rule via API",
"uid": "dHEquNzGz",
"version": 1
}
}

```

#### Example Response:

```

HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 78

```

```
{
```

```

    "id":      1,
    "uid":      "cIBgcSjkk",
    "url":      "/d/cIBgcSjkk/production-overview",
    "status":   "success",
    "version":  1,
    "slug":     "production-overview" //deprecated in Grafana v5.0
  }

```

Status Codes:

- **200** – Created
- **400** – Errors (invalid json, missing or invalid fields, etc)
- **401** – Unauthorized
- **403** – Access denied
- **412** – Precondition failed

The **412** status code is used for explaining that you cannot create the dashboard and why. There can be different reasons for this:

- The dashboard has been changed by someone else, **status=version-mismatch**
- A dashboard with the same name in the folder already exists, **status=name-exists**
- A dashboard with the same uid already exists, **status=name-exists**
- The dashboard belongs to plugin <plugin title>, **status=plugin-dashboard**

The response body will have the following properties:

```

HTTP/1.1 412 Precondition Failed
Content-Type: application/json; charset=UTF-8
Content-Length: 97

```

```

{
  "message": "The dashboard has been changed by someone else",
  "status": "version-mismatch"
}

```

In case of title already exists the **status** property will be **name-exists**.

## Get dashboard by uid

GET /api/dashboards/uid/:uid

Will return the dashboard given the dashboard unique identifier (uid). Information about the unique identifier of a folder containing the requested dashboard might be found in the metadata.

### Example Request:

```

GET /api/dashboards/uid/cIBgcSjkk HTTP/1.1
Accept: application/json

```

Content-Type: application/json  
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

**Example Response:**

HTTP/1.1 200  
Content-Type: application/json

```
{
  "dashboard": {
    "id": 1,
    "uid": "cIBgcSjkk",
    "title": "Production Overview",
    "tags": [ "templated" ],
    "timezone": "browser",
    "schemaVersion": 16,
    "version": 0
  },
  "meta": {
    "isStarred": false,
    "url": "/d/cIBgcSjkk/production-overview",
    "folderId": 2,
    "folderUid": "l3KqBxCMz",
    "slug": "production-overview" //deprecated in Grafana v5.0
  }
}
```

**Status Codes:**

- **200** – Found
- **401** – Unauthorized
- **403** – Access denied
- **404** – Not found

**Delete dashboard by uid**

DELETE /api/dashboards/uid/:uid

Will delete the dashboard given the specified unique identifier (uid).

**Example Request:**

DELETE /api/dashboards/uid/cIBgcSjkk HTTP/1.1  
Accept: application/json  
Content-Type: application/json  
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

**Example Response:**

HTTP/1.1 200  
Content-Type: application/json

```
{  
  "title": "Production Overview",  
  "message": "Dashboard Production Overview deleted",  
  "id": 2  
}
```

Status Codes:

- **200** – Deleted
- **401** – Unauthorized
- **403** – Access denied
- **404** – Not found

## Gets the home dashboard

GET /api/dashboards/home

Will return the home dashboard.

### Example Request:

GET /api/dashboards/home HTTP/1.1  
Accept: application/json  
Content-Type: application/json  
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

### Example Response:

HTTP/1.1 200  
Content-Type: application/json

```
{  
  "dashboard": {  
    "editable": false,  
    "hideControls": true,  
    "nav": [  
      {  
        "enable": false,  
        "type": "timepicker"  
      }  
    ],  
    "style": "dark",  
    "tags": [],  
    "templating": {  
      "list": [  
        ]  
      }  
    }  
  }
```



```

    },
    "time":{
    },
    "timezone":"browser",
    "title":"Home",
    "version":5
  },
  "meta": {
    "isHome":true,
    "canSave":false,
    "canEdit":false,
    "canStar":false,
    "url":"",
    "expires":"0001-01-01T00:00:00Z",
    "created":"0001-01-01T00:00:00Z"
  }
}

```

## Tags for Dashboard

GET /api/dashboards/tags

Get all tags of dashboards

### Example Request:

GET /api/dashboards/tags HTTP/1.1

Accept: application/json

Content-Type: application/json

Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

### Example Response:

HTTP/1.1 200

Content-Type: application/json

```

[
  {
    "term":"tag1",
    "count":1
  },
  {
    "term":"tag2",
    "count":4
  }
]

```

## Dashboard Search

See [Folder/Dashboard Search API]({{< relref "folder\_dashboard\_search.md">}}).