Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Folder API

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information.

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

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

The unique identifier (uid) of a folder can be used for uniquely identify folders between multiple Grafana installs. It's automatically generated if not provided when creating a folder. The uid allows having consistent URLs for accessing folders and when syncing folders between multiple Grafana installs. This means that changing the title of a folder will not break any bookmarked links to that folder.

The uid can have a maximum length of 40 characters.

## Get all folders

`GET /api/folders`

Returns all folders that the authenticated user has permission to view. You can control the maximum number of folders returned through the `limit` query parameter, the default is 1000. You can also pass the `page` query parameter for fetching folders from a page other than the first one.

If nested folders are enabled, the operation expects an additional optional query parameter `parentUid` with the parent folder UID, and returns the immediate subfolders that the authenticated user has permission to view. If the parameter is not supplied, then the operation returns immediate subfolders under the root that the authenticated user has permission to view.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `folders:read` | `folders:*` |

**Example Request**:

```http
GET /api/folders?limit=10 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id":1,
    "uid": "nErXDvCkzz",
    "title": "Department ABC"
  },
  {
    "id":2,
    "uid": "k3S1cklGk",
    "title": "Department RND"
```

```
  }
]
```

## Get folder by uid

`GET /api/folders/:uid`

Will return the folder given the folder uid.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `folders:read` | `folders:*` |

**Example Request:**

http Copy

```
GET /api/folders/nErXDvCkzzh HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "uid": "nErXDvCkzz",
  "title": "Department ABC",
  "url": "/dashboards/f/nErXDvCkzz/department-abc",
  "hasAcl": false,
  "canSave": true,
  "canEdit": true,
```

```
  "canAdmin": true,
  "createdBy": "admin",
  "created": "2018-01-31T17:43:12+01:00",
  "updatedBy": "admin",
  "updated": "2018-01-31T17:43:12+01:00",
  "version": 1
}
```


If nested folders are enabled, and the folder is nested (lives under another folder), then the response additionally contains:

- **parentUid** - The parent folder UID.
- **parents** - An array with the whole tree hierarchy, starting from the root going down up to the parent folder.

Status Codes:

- **200** – Found
- **401** – Unauthorized
- **403** – Access Denied
- **404** – Folder not found

## Create folder

`POST /api/folders`

Creates a new folder.

**Required permissions**

See note in the introduction for an explanation.

`folders:create` allows creating folders in the root level. To create a subfolder, `folders:write` scoped to the parent folder is required in addition to `folders:create`.

| Action | Scope |
| --- | --- |
| `folders:create` | n/a |
| `folders:write` | `folders:*` |

**Example Request**:

```
http
```

Copy

```
POST /api/folders HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk


{
  "uid": "nErXDvCkzz",
  "title": "Department ABC",
  "parentUid" "fgnj5e52gel76g"
}
```

JSON Body schema:

- **uid** – Optional unique identifier.
- **title** – The title of the folder.
- **parentUid** - Optional field, the unique identifier of the parent folder under which the folder should be created. Requires nested folders to be enabled.

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json


{
  "id":1,
  "uid": "nErXDvCkzz",
  "title": "Department ABC",
  "url": "/dashboards/f/nErXDvCkzz/department-abc",
  "hasAcl": false,
  "canSave": true,
  "canEdit": true,
  "canAdmin": true,
  "createdBy": "admin",
  "created": "2018-01-31T17:43:12+01:00",
  "updatedBy": "admin",
  "updated": "2018-01-31T17:43:12+01:00",
  "version": 1
}
```

If nested folders are enabled, and the folder is nested (lives under another folder) then the response additionally contains:

- **parentUid** - the parent folder UID.
- **parents** - an array with the whole tree hierarchy starting from the root going down up to the parent folder.

Status Codes:

- **200** – Created
- **400** – Errors (invalid json, missing or invalid fields, etc)
- **401** – Unauthorized
- **403** – Access Denied
- **409** - Folder already exists

## Update folder

`PUT /api/folders/:uid`

Updates an existing folder identified by uid.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `folders:write` | `folders:*` |

**Example Request:**

http Copy

```
PUT /api/folders/nErXDvCkzz HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "title":"Department DEF",
```

```
  "version": 1
}
```

JSON Body schema:

- **title** – The title of the folder.
- **version** – Provide the current version to be able to update the folder. Not needed if `overwrite=true`.
- **overwrite** – Set to true if you want to overwrite existing folder with newer version.

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "uid": "nErXDvCkzz",
  "title": "Department DEF",
  "url": "/dashboards/f/nErXDvCkzz/department-def",
  "hasAcl": false,
  "canSave": true,
  "canEdit": true,
  "canAdmin": true,
  "createdBy": "admin",
  "created": "2018-01-31T17:43:12+01:00",
  "updatedBy": "admin",
  "updated": "2018-01-31T17:43:12+01:00",
  "version": 1
}
```

If nested folders are enabled, and the folder is nested (lives under another folder) then the response additionally contains:

- **parentUid** - the parent folder UID.
- **parents** - an array with the whole tree hierarchy starting from the root going down up to the parent folder.

Status Codes:

- **200** – Updated

---

- **400** – Errors (invalid json, missing or invalid fields, etc)
- **401** – Unauthorized
- **403** – Access Denied
- **404** – Folder not found
- **412** – Precondition failed

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

- The folder has been changed by someone else, `status=version-mismatch`

The response body will have the following properties:

http Copy

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

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

## Delete folder

`DELETE /api/folders/:uid`

Deletes an existing folder identified by UID along with all dashboards (and their alerts) stored in the folder. This operation cannot be reverted.

If Grafana Alerting is enabled, you can set an optional query parameter `forceDeleteRules=false` so that requests will fail with 400 (Bad Request) error if the folder contains any Grafana alerts. However, if this parameter is set to `true` then it will delete any Grafana alerts under this folder.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `folders:delete` | `folders:*` |

**Example Request:**

```http
DELETE /api/folders/nErXDvCkzz HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{
  "message":"Folder deleted",
  "id": 2
}
```

Status Codes:

- **200** – Deleted
- **401** – Unauthorized
- **400** – Bad Request
- **403** – Access Denied
- **404** – Folder not found

## Get folder by id

`GET /api/folders/id/:id`

Will return the folder identified by id.

This is deprecated. Use get folder by UID instead.

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `folders:read` | `folders:*` |

### Example Request:

```http
GET /api/folders/id/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "uid": "nErXDvCkzz",
  "title": "Department ABC",
  "url": "/dashboards/f/nErXDvCkzz/department-abc",
  "hasAcl": false,
  "canSave": true,
  "canEdit": true,
  "canAdmin": true,
  "createdBy": "admin",
  "created": "2018-01-31T17:43:12+01:00",
  "updatedBy": "admin",
  "updated": "2018-01-31T17:43:12+01:00",
  "version": 1
}
```

Status Codes:

- **200** – Found
- **401** – Unauthorized
- **403** – Access Denied
- **404** – Folder not found

## Move folder

`POST /api/folders/:uid/move`

Moves the folder.

This is relevant only if nested folders are enabled.

**Required permissions**

See note in the introduction for an explanation.

If moving the folder under another folder:

| Action | Scope |
| --- | --- |
| `folders:write` | `folders:uid:<destination folder UID>` |

If moving the folder under root:

| Action | Scope |
| --- | --- |
| `folders:create` | `folders:*` |

JSON body schema:

- **parentUid** – Optional unique identifier of the new parent folder. If this is empty, then the folder is moved under the root.

**Example Request**:

```http
POST /api/folders/a5393ec3-5568-4e88-8809-b866968ae8a6/move HTTP/1.1
Accept: application/json
Content-Type: application/json
```

```
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "parentUid": "d80b18c0-266a-4aa4-ad5d-5537a00cb8e8",
}
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
    "id": 4,
    "uid": "a5393ec3-5568-4e88-8809-b866968ae8a6",
    "title": "just-testing",
    "url": "/dashboards/f/a5393ec3-5568-4e88-8809-b866968ae8a6/just-testing",
    "hasAcl": false,
    "canSave": true,
    "canEdit": true,
    "canAdmin": true,
    "canDelete": true,
    "createdBy": "Anonymous",
    "created": "2023-04-27T21:55:01.593741+03:00",
    "updatedBy": "Anonymous",
    "updated": "2023-04-27T21:55:15.747444+03:00",
    "parentUid": "d80b18c0-266a-4aa4-ad5d-5537a00cb8e8",
    "parents": [
```

Expand code

Status Codes:

- **200** – Moved
- **400** – Errors (invalid JSON, missing or invalid fields, and so on)
- **401** – Unauthorized
- **403** – Access denied
- **404** – Not found

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Manage folders, data sources, and dashboards using Grafana Operator

Docs

Grafana SLO Folder RBAC

Docs

HTTP API

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

**Products**

Grafana Cloud

Grafana Enterprise Stack

**Open Source**

Grafana

Grafana Alloy

**Learn**

Documentation

Grafana Labs blog

Grafana visualizations
Plugins
Grafana Cloud Logs
Grafana Cloud Metrics
Grafana Cloud Traces
Grafana Cloud Profiles
AI/ML tools for observability
Application Observability
Frontend Observability
Grafana Cloud Asserts
Grafana IRM
Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Grafana Beyla
Grafana Faro
Grafana k6
Grafana Loki
Grafana Mimir
Grafana OnCall
Grafana Pyroscope
Grafana Tanka
Grafana Tempo
Graphite
OpenTelemetry
Prometheus
Dashboard templates
Downloads
GitHub

Webinars and videos
Tutorials
Workshops
Events
Community
Community forums
Community Slack
Grafana Champions
Community organizers
Observability Survey
OSS vs Cloud
Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Folder Permissions API

This API can be used to update/get the permissions for a folder.

Permissions with `folderId=-1` are the default permissions for users with the Viewer and Editor roles. Permissions can be set for a user, a team or a role (Viewer or Editor). Permissions cannot be set for Admins - they always have access to everything.

The permission levels for the permission field:

- 1 = View
- 2 = Edit
- 4 = Admin

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information.

## Get permissions for a folder

`GET /api/folders/:uid/permissions`

Gets all existing permissions for the folder with the given `uid`.

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `folders.permissions:read` | `folders:*` |

### Example request:

```http
GET /api/folders/nErXDvCkzz/permissions HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response

```http
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 551

[
  {
    "id": 1,
    "folderId": -1,
    "created": "2017-06-20T02:00:00+02:00",
    "updated": "2017-06-20T02:00:00+02:00",
    "userId": 0,
    "userLogin": "",
    "userEmail": "",
    "teamId": 0,
    "team": "",
    "role": "Viewer",
    "permission": 1,
    "permissionName": "View",
    "uid": "nErXDvCkzz",
```

Expand code

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Access denied

- **404** - Folder not found

## Update permissions for a folder

`POST /api/folders/:uid/permissions`

Updates permissions for a folder. This operation will remove existing permissions if they're not included in the request.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `folders.permissions:write` | `folders:*` |

**Example request:**

http Copy

```
POST /api/folders/nErXDvCkzz/permissions
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
{
  "items": [
    {
      "role": "Viewer",
      "permission": 1
    },
    {
      "role": "Editor",
      "permission": 2
    },
    {
      "teamId": 1,
      "permission": 1
    },
    {
```

Expand code

JSON body schema:

- **items** - The permission items to add/update. Items that are omitted from the list will be removed.

**Example response**:

```http
HTTP/1.1 200 OK
Content-Type: application/json; charset=UTF-8
Content-Length: 35

{"message":"Folder permissions updated","id":1,"title":"Department ABC"}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Access denied
- **404** - Dashboard not found

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Roles and permissions

Docs

Create Grafana Cloud API keys

Docs

Folder HTTP API

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

### Company

### Localized content

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



# Folder/Dashboard Search API

## Search folders and dashboards

`GET /api/search/`

Note: When using Role-based access control, search results will contain only dashboards and folders which you have access to.

Query parameters:

- **query** – Search Query
- **tag** – List of tags to search for
- **type** – Type to search for, `dash-folder` or `dash-db`
- **dashboardIds** – List of dashboard id's to search for
- **dashboardUID** - List of dashboard uid's to search for, It is deprecated since Grafana v9.1, please use dashboardUIDs instead
- **dashboardUIDs** – List of dashboard uid's to search for
- **folderUIDs** – List of folder UIDs to search in
- **starred** – Flag indicating if only starred Dashboards should be returned
- **limit** – Limit the number of returned results (max is 5000; default is 1000)
- **page** – Use this parameter to access hits beyond limit. Numbering starts at 1. limit param acts as page size. Only available in Grafana v6.2+.

**Example request for retrieving folders and dashboards at the root level:**

```http
GET /api/search?query=&starred=false HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

Example response for retrieving folders and dashboards at the root level:

```http
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id": 163,
    "uid": "000000163",
    "title": "Folder",
    "url": "/dashboards/f/000000163/folder",
    "type": "dash-folder",
    "tags": [],
    "isStarred": false,
    "uri":"db/folder" // deprecated in Grafana v5.0
  },
  {
    "id":1,
    "uid": "cIBgcSjkk",
    "title":"Production Overview",
    "url": "/d/cIBgcSjkk/production-overview",
```

Example request searching for dashboards:

```http
GET /api/search?query=Production%20Overview&starred=true&tag=prod HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

Example response searching for dashboards:

http

Copy

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id":1,
    "uid": "cIBgcSjkk",
    "title":"Production Overview",
    "url": "/d/cIBgcSjkk/production-overview",
    "type":"dash-db",
    "tags":[prod],
    "isStarred":true,
    "folderId": 2,
    "folderUid": "000000163",
    "folderTitle": "Folder",
    "folderUrl": "/dashboards/f/000000163/folder",
    "uri":"db/production-overview" // deprecated in Grafana v5.0
  }
]
```

Expand code

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Folder HTTP API

Docs

Search dashboards and folders

Docs

Search dashboards and folders

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

## Company

- Partnerships
- Our team
- Careers
- Success stories
- Newsroom
- Contact us
- Getting help
- Merch
- Open positions WE'RE HIRING
- The Story of Grafana

## Localized content

- Japanese pages
- German pages
- French pages
- Spanish pages
- Portuguese pages

---

- Grafana Cloud status
- Legal and Security
- Terms of Service
- Privacy Policy
- Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# User and Org Preferences API

Keys:

- **theme** - One of: `light`, `dark`, or an empty string for the default theme
- **homeDashboardId** - The numerical `:id` of a favorited dashboard, default: `0`
- **timezone** - One of: `utc`, `browser`, or an empty string for the default

Omitting a key will cause the current value to be replaced with the system default value.

## Get Current User Prefs

`GET /api/user/preferences`

**Example Request:**

```http
GET /api/user/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json
```

```
{
    "theme": "",
    "homeDashboardId": 217,
    "homeDashboardUID": "jcIIG-07z",
    "timezone": "utc",
    "weekStart": "",
    "navbar": {
        "savedItems": null
    },
    "queryHistory": {
        "homeTab": ""
    }
}
```


## Update Current User Prefs

`PUT /api/user/preferences`

### Example Request:

http Copy

```
PUT /api/user/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "theme": "",
  "homeDashboardUID":"home",
  "timezone":"utc"
}
```

### Example Response:

http Copy

```
HTTP/1.1 200
Content-Type: text/plain; charset=utf-8
```

```
{"message":"Preferences updated"}
```

## Patch Current User Prefs

Update one or more preferences without modifying the others.

`PATCH /api/user/preferences`

### Example Request:

```http
PATCH /api/user/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "theme": "dark"
}
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: text/plain; charset=utf-8

{"message":"Preferences updated"}
```

## Get Current Org Prefs

`GET /api/org/preferences`

### Example Request:

```http
GET /api/org/preferences HTTP/1.1
Accept: application/json
```

```
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
    "theme": "",
    "homeDashboardId": 0,
    "timezone": "",
    "weekStart": "",
    "navbar": {
        "savedItems": null
    },
    "queryHistory": {
        "homeTab": ""
    }
}
```

## Update Current Org Prefs

`PUT /api/org/preferences`

**Example Request:**

http Copy

```
PUT /api/org/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "theme": "",
  "homeDashboardUID":"home",
```

```
  "timezone":"utc"
}
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: text/plain; charset=utf-8

{"message":"Preferences updated"}
```

## Patch Current Org Prefs

Update one or more preferences without modifying the others.

`PATCH /api/org/preferences`

**Example Request:**

```http
PATCH /api/org/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "theme": "dark"
}
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: text/plain; charset=utf-8

{"message":"Preferences updated"}
```

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Preferences API

Docs

Organization preferences

Docs

Manage user preferences

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

**Products**

Grafana Cloud

Grafana Enterprise Stack

**Open Source**

Grafana

Grafana Alloy

**Learn**

Documentation

Grafana Labs blog

Grafana visualizations
Plugins
Grafana Cloud Logs
Grafana Cloud Metrics
Grafana Cloud Traces
Grafana Cloud Profiles
AI/ML tools for observability
Application Observability
Frontend Observability
Grafana Cloud Asserts
Grafana IRM
Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Grafana Beyla
Grafana Faro
Grafana k6
Grafana Loki
Grafana Mimir
Grafana OnCall
Grafana Pyroscope
Grafana Tanka
Grafana Tempo
Graphite
OpenTelemetry
Prometheus
Dashboard templates
Downloads
GitHub

Webinars and videos
Tutorials
Workshops
Events
Community
Community forums
Community Slack
Grafana Champions
Community organizers
Observability Survey
OSS vs Cloud
Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Snapshot API

## Create new snapshot

`POST /api/snapshots`

**Example Request:**

http Copy

```
POST /api/snapshots HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "dashboard": {
    "editable":false,
    "nav":[
    {
      "enable":false,
    "type":"timepicker"
    }
    ],
    "rows": [
      {

      }
```

Expand code

```
]
```

JSON Body schema:

- **dashboard** – Required. The complete dashboard model.
- **name** – Optional. snapshot name
- **expires** - Optional. When the snapshot should expire in seconds. 3600 is 1 hour, 86400 is 1 day. Default is never to expire.
- **external** - Optional. Save the snapshot on an external server rather than locally. Default is `false`.
- **key** - Optional. Define the unique key. Required if **external** is `true`.
- **deleteKey** - Optional. Unique key used to delete the snapshot. It is different from the **key** so that only the creator can delete the snapshot. Required if **external** is `true`.

> **NOTE**
> When creating a snapshot using the API, you have to provide the full dashboard payload including the snapshot data. This endpoint is designed for the Grafana UI.

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json
{
  "deleteKey":"XXXXXXX",
  "deleteUrl":"myurl/api/snapshots-delete/XXXXXXX",
  "key":"YYYYYYY",
  "url":"myurl/dashboard/snapshot/YYYYYYY",
  "id": 1
}
```

Keys:

- **deleteKey** – Key generated to delete the snapshot
- **key** – Key generated to share the dashboard

## Get list of Snapshots

`GET /api/dashboard/snapshots`

Query parameters:

- **query** – Search Query
- **limit** – Limit the number of returned results

**Example Request:**

```http
GET /api/dashboard/snapshots HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id":8,
    "name":"Home",
    "key":"YYYYYYY",
    "orgId":1,
    "userId":1,
    "external":false,
    "externalUrl":"",
    "expires":"2200-13-32T25:23:23+02:00",
    "created":"2200-13-32T28:24:23+02:00",
    "updated":"2200-13-32T28:24:23+02:00"
  }
]
```

## Get Snapshot by Key

`GET /api/snapshots/:key`

**Example Request:**

```http
```

```
GET /api/snapshots/YYYYYYY HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
  "meta":{
    "isSnapshot":true,
    "type":"snapshot",
    "canSave":false,
    "canEdit":false,
    "canStar":false,
    "slug":"",
    "expires":"2200-13-32T25:23:23+02:00",
    "created":"2200-13-32T28:24:23+02:00"
  },
  "dashboard": {
    "editable":false,
    "nav": [
      {
        "enable":false,
```

Expand code

## Delete Snapshot by Key

`DELETE /api/snapshots/:key`

**Example Request:**

http Copy

```
DELETE /api/snapshots/YYYYYYY HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Snapshot deleted. It might take an hour before it's cleared from any CDN cac
```

## Delete Snapshot by deleteKey

This API call can be used without authentication by using the secret delete key for the snapshot.

`GET /api/snapshots-delete/:deleteKey`

**Example Request**:

```http
GET /api/snapshots-delete/XXXXXXX HTTP/1.1
Accept: application/json
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Snapshot deleted. It might take an hour before it's cleared from any CDN cac
```

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Snapshot API

Docs

The Pipeline HTTP API

Docs

Issues snapshot in Kubernetes Monitoring

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions

Frontend Observability
Grafana Cloud Asserts
Grafana IRM
Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Grafana Tempo
Graphite
OpenTelemetry
Prometheus
Dashboard templates
Downloads
GitHub

Community organizers
Observability Survey
OSS vs Cloud
Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Library Element API

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

The identifier (ID) of a library element is an auto-incrementing numeric value that is unique per Grafana install.

The unique identifier (UID) of a library element uniquely identifies library elements between multiple Grafana installs. It's automatically generated unless you specify it during library element creation. The UID provides consistent URLs for accessing library elements and when syncing library elements between multiple Grafana installs.

The maximum length of a UID is 40 characters.

## Get all library elements

`GET /api/library-elements`

Returns a list of all library elements the authenticated user has permission to view. Use the `perPage` query parameter to control the maximum number of library elements returned; the default limit is 100. You can also use the `page` query parameter to fetch library elements from any page other than the first one.

Query parameters:

- `searchString`: Part of the name or description searched for.
- `kind`: Kind of element to search for. Use `1` for library panels or `2` for library variables.
- `sortDirection`: Sort order of elements. Use `alpha-asc` for ascending and `alpha-desc` for descending sort order.
- `typeFilter`: A comma separated list of types to filter the elements by.

- `excludeUid`: Element UID to exclude from search results.
- `folderFilter`: A comma separated list of folder IDs to filter the elements by.
- `perPage`: The number of results per page; default is 100.
- `page`: The page for a set of records, given that only `perPage` records are returned at a time. Numbering starts at `1`.

**Example Request**:

```http
GET /api/library-elements?perPage=10 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{
  "result": {
    "totalCount": 15,
    "page": 1,
    "perPage": 10
    "elements": [
      {
        "id": 25,
        "orgId": 1,
        "folderId": 0,
        "uid": "V--OrYHnz",
        "name": "API docs Example",
        "kind": 1,
        "type": "text",
        "description": "",
        "model": {...},
```

Status Codes:

- `200`: Found

- `401`: Unauthorized

## Get library element by uid

`GET /api/library-elements/:uid`

Returns a library element with the given UID.

**Example Request**:

```http
GET /api/library-elements/V--OrYHnz HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{
    "result": {
      "id": 25,
      "orgId": 1,
      "folderId": 0,
      "uid": "V--OrYHnz",
      "name": "API docs Example",
      "kind": 1,
      "type": "text",
      "description": "",
      "model": {...},
      "version": 1,
      "meta": {
          "folderName": "General",
          "folderUid": "",
          "connectedDashboards": 1,
```

Status Codes:

- `200`: Found
- `401`: Unauthorized
- `404`: Library element not found

## Get library element by name

`GET /api/library-elements/name/:name`

Returns a library element with the given name

**Example Request**:

```http
GET /api/library-elements/name/API docs Example HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{
    "result": [
        {
            "id": 25,
            "orgId": 1,
            "folderId": 0,
            "uid": "V--OrYHnz",
            "name": "API docs Example",
            "kind": 1,
            "type": "text",
            "description": "",
            "model": {...},
            "version": 1,
            "meta": {
                "folderName": "General",
```

Copy
Expand code

Status Codes:

- `200`: Found
- `401`: Unauthorized
- `404`: Library element not found

## Get library element connections

`GET /api/library-elements/:uid/connections`

Returns a list of connections for a library element based on the UID specified.

**Example Request**:

http Copy

```
GET /api/library-elements/V--OrYHnz/connections HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
    "result": [
        {
            "id": 148,
            "kind": 1,
            "elementId": 25,
            "connectionId": 527,
            "connectionUid": "dHEquNzGz",
            "created": "2021-09-27T10:00:07+02:00",
            "createdBy": {
                "id": 1,
                "name": "admin",
                "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
```

```
            }
        }
```

Expand code

Status Codes:

- `200`: Found
- `401`: Unauthorized
- `404`: Library element not found

## Create library element

`POST /api/library-elements`

Creates a new library element.

JSON Body schema:

- `folderId`: ID of the folder where the library element is stored. It is deprecated since Grafana v9
- `folderUid`: Optional, the UID of the folder where the library element is stored, empty string when it is at the root level.
- `name`: Optional, the name of the library element.
- `model`: The JSON model for the library element.
- `kind`: Kind of element to create, Use `1` for library panels or `2` for library variables.
- `uid`: Optional, the unique identifier.

**Example Request**:

http

Copy

```
POST /api/library-elements HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk


{
  "uid": "nErXDvCkzz",
  "folderUid": "",
  "name": "Example library panel",
  "model": {...},
  "kind": 1
}
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{
    "result": {
        "id": 28,
        "orgId": 1,
        "folderId": 0,
        "folderUid": "",
        "uid": "nErXDvCkzz",
        "name": "Example library panel",
        "kind": 1,
        "type": "",
        "description": "",
        "model": {...},
        "version": 1,
        "meta": {
            "folderName": "General",
            "folderUid": "",
```

Copy

Expand code

Status Codes:

- `200`: Created
- `400`: Errors (for example, name or UID already exists, invalid JSON, missing or invalid fields, and so on).
- `401`: Unauthorized
- `403`: Access denied

## Update library element

`PATCH /api/library-elements/:uid`

Updates an existing library element identified by uid.

JSON Body schema:

- `folderId`: ID of the folder where the library element is stored. It is deprecated since Grafana v9

- `folderUid`: UID of the folder where the library element is stored, empty string when it is at the root level.
- `name`: Name of the library element.
- `model`: The JSON model for the library element.
- `kind`: Kind of element to create. Use `1` for library panels or `2` for library variables.
- `version`: Version of the library element you are updating.
- `uid`: Optional, the unique identifier.

**Example Request**:

http Copy

```
PATCH /api/library-elements/nErXDvCkzz HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name": "Renamed library panel",
  "kind": 1,
  "version": 1
}
```

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
    "result": {
        "id": 28,
        "orgId": 1,
        "folderId": 0,
        "folderUid": "",
        "uid": "nErXDvCkzz",
        "name": "Renamed library panel",
        "kind": 1,
        "type": "",
```

```
        "description": "",
        "model": {
            "description": "",
            "type": ""
        },
```


Expand code

Status Codes:

- `200`: Updated
- `400`: Errors (for example, name or UID already exists, invalid JSON, missing or invalid fields, and so on).
- `401`: Unauthorized
- `403`: Access denied
- `404`: Library element not found
- `412`: Version mismatch

## Delete library element

`DELETE /api/library-elements/:uid`

Deletes an existing library element as specified by the UID. This operation cannot be reverted.

> **NOTE**
> You cannot delete a library element that is connected. This operation cannot be reverted.

**Example Request**:

http Copy

```
DELETE /api/library-elements/nErXDvCkzz HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
    "message": "Library element deleted",
    "id": 28
}
```

Status Codes:

- `200`: Deleted
- `401`: Unauthorized
- `400`: Bad request
- `403`: Access denied
- `404`: Library element not found

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Element.selection()

Docs

Client libraries and SDKs

Docs

Client libraries and SDKs

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

### Company

- Partnerships
- Our team

### Localized content

- Japanese pages
- German pages

Careers

Success stories

Newsroom

Contact us

Getting help

Merch

Open positions WE'RE HIRING

The Story of Grafana

French pages

Spanish pages

Portuguese pages

---

Grafana Cloud status

Legal and Security

Terms of Service

Privacy Policy

Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Enterprise License API

Licensing is only available in Grafana Enterprise. Read more about Grafana Enterprise.

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information.

## Check license availability

**Note:** Available in Grafana Enterprise v7.4+.

`GET /api/licensing/check`

Checks if a valid license is available.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| licensing:read | n/a |

### Examples

**Example request:**

```http
GET /api/licensing/check
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example response:**

```http
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 4

true
```

Status codes:

- **200** - OK

## Add license

> **NOTE**
> Available in Grafana Enterprise v7.4+.

`POST /api/licensing/token`

Applies a license to a Grafana instance.

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| licensing:write | n/a |

## Examples

### Example request:

```http
POST /licensing/token
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{"token":"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ0aGlzIjoiaXMiLCJub3QiOiJhIiwidmFsaWQiO
```

### Example response:

```http
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 357

{
  "jti":"2",
  "iss":"https://grafana.com",
  "sub":"https://play.grafana.org/"
  "lid":"1",
  "included_users":15,
  "lic_exp_warn_days":30,
  "tok_exp_warn_days":2,
  "update_days":1,
  "prod":["grafana-enterprise"],
  "company":"Grafana Labs"
}
```

The response is a JSON blob available for debugging purposes. The available fields may change at any time without any prior notice.

Status Codes:

- **200** - OK
- **400** - Bad request
- **500** - Internal server error (refer to server logs for more details)

## Manually force license refresh

> **NOTE**
> Available in Grafana Enterprise v7.4+.

`POST /api/licensing/token/renew`

Manually ask license issuer for a new token.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| licensing:write | n/a |

## Examples

### Example request:

```http
POST /api/licensing/token/renew
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{}
```

### Example response:

```http
```

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 357

{
  "jti":"2",
  "iss":"https://grafana.com",
  "sub":"https://play.grafana.org/"
  "lid":"1",
  "included_users":15,
  "lic_exp_warn_days":30,
  "tok_exp_warn_days":2,
  "update_days":1,
  "prod":["grafana-enterprise"],
  "company":"Grafana Labs"
}
```

The response is a JSON blob available for debugging purposes. The available fields may change at any time without any prior notice.

Status Codes:

- **200** - OK
- **401** - Unauthorized
- **403** - Access denied

## Remove license from database

> **NOTE**
> Available in Grafana Enterprise v7.4+.

`DELETE /api/licensing/token`

Removes the license stored in the Grafana database.

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| licensing:delete | n/a |

## Examples

### Example request:

```http
DELETE /api/licensing/token
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{"instance": "http://play.grafana.org/"}
```

JSON Body schema:

- **instance** – Root URL for the instance for which the license should be deleted. Required.

### Example response:

```http
HTTP/1.1 202 Accepted
Content-Type: application/json
Content-Length: 2

{}
```

Status codes:

- **202** - Accepted, license removed or did not exist.
- **401** - Unauthorized
- **403** - Access denied
- **422** - Unprocessable entity, incorrect instance name provided.

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Grafana Enterprise license

Docs

Activate a Grafana Enterprise license purchased through AWS Marketplace

Docs

Activate a Grafana Enterprise license from AWS Marketplace on ECS

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

**Products**

Grafana Cloud

Grafana Enterprise Stack

**Open Source**

Grafana

Grafana Alloy

**Learn**

Documentation

Grafana Labs blog

Grafana visualizations
Plugins
Grafana Cloud Logs
Grafana Cloud Metrics
Grafana Cloud Traces
Grafana Cloud Profiles
AI/ML tools for observability
Application Observability
Frontend Observability
Grafana Cloud Asserts
Grafana IRM
Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Grafana Beyla
Grafana Faro
Grafana k6
Grafana Loki
Grafana Mimir
Grafana OnCall
Grafana Pyroscope
Grafana Tanka
Grafana Tempo
Graphite
OpenTelemetry
Prometheus
Dashboard templates
Downloads
GitHub

Webinars and videos
Tutorials
Workshops
Events
Community
Community forums
Community Slack
Grafana Champions
Community organizers
Observability Survey
OSS vs Cloud
Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Organization API

The Organization HTTP API is divided in two resources, `/api/org` (current organization) and `/api/orgs` (admin organizations). One big difference between these are that the admin of all organizations API only works with basic authentication, see Admin Organizations API for more information.

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information.

## Current Organization API

### Get current Organization

`GET /api/org/`

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| orgs:read | N/A |

**Example Request**:

```http
GET /api/org/ HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "name":"Main Org."
}
```

## Get all users within the current organization

`GET /api/org/users`

Returns all org users within the current organization. Accessible to users with org admin role.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| org.users:read | users:* |

**Example Request**:

```http
GET /api/org/users HTTP/1.1
Accept: application/json
```

```
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "orgId": 1,
    "userId": 1,
    "email": "admin@localhost",
    "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56",
    "login": "admin",
    "role": "Admin",
    "lastSeenAt": "2019-08-09T11:02:49+02:00",
    "lastSeenAtAge": "< 1m"
  }
]
```

## Get all users within the current organization (lookup)

`GET /api/org/users/lookup`

Returns all org users within the current organization, but with less detailed information. Accessible to users with org admin role, admin in any folder or admin of any team. Mainly used by Grafana UI for providing list of users when adding team members and when editing folder/dashboard permissions.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| org.users:read | users:* |

**Example Request:**

```http
GET /api/org/users/lookup HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

[
  {
    "userId": 1,
    "login": "admin",
    "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
  }
]
```

## Updates the given user

`PATCH /api/org/users/:userId`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| org.users:write | users:* |

**Example Request:**

```http
PATCH /api/org/users/1 HTTP/1.1
Accept: application/json
```

```
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "role": "Viewer",
}
```

**Example Response:**

```
http
Copy

HTTP/1.1 200
Content-Type: application/json

{"message":"Organization user updated"}
```

## Delete user in current organization

`DELETE /api/org/users/:userId`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| org.users:remove | users:* |

**Example Request:**

```
http
Copy

DELETE /api/org/users/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```
http
Copy
```

```
HTTP/1.1 200
Content-Type: application/json

{"message":"User removed from organization"}
```

## Update current Organization

`PUT /api/org`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| orgs:write | N/A |

**Example Request:**

http Copy

```
PUT /api/org HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "name":"Main Org."
}
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Organization updated"}
```

## Add a new user to the current organization

`POST /api/org/users`

Adds a global user to the current organization.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| org.users:add | users:* |

**Example Request:**

```http
POST /api/org/users HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "role": "Admin",
  "loginOrEmail": "admin"
}
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"User added to organization","userId":1}
```

# Admin Organizations API

The Admin Organizations HTTP API does not currently work with an API Token. API Tokens are currently only linked to an organization and an organization role. They cannot be given the permission of server admin, only users can be given that permission. So in order to use these API calls you will have to use Basic Auth and the Grafana user must have the Grafana Admin permission (The default admin user is called `admin` and has permission to use this API).

## Get Organization by Id

`GET /api/orgs/:orgId`

Only works with Basic Authentication (username and password), see introduction.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| orgs:read | N/A |

**Example Request**:

```http
GET /api/orgs/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "name":"Main Org.",
  "address":{
    "address1":"",
    "address2":"",
```

```
    "city":"",
    "zipCode":"",
    "state":"",
    "country":""
  }
}
```


## Get Organization by Name

`GET /api/orgs/name/:orgName`

Only works with Basic Authentication (username and password), see introduction.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope | Note |
| --- | --- | --- |
| orgs:read | N/A | Needs to be assigned globally. |

**Example Request**:

http Copy

```
GET /api/orgs/name/Main%20Org%2E HTTP/1.1
Accept: application/json
Content-Type: application/json
```

**Example Response**:

http Copy


```
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "name":"Main Org.",
  "address":{
    "address1":"",
```

```
    "address2":"",
    "city":"",
    "zipCode":"",
    "state":"",
    "country":""
  }
}
```

## Create Organization

`POST /api/orgs`

Only works with Basic Authentication (username and password), see introduction.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope | Note |
| --- | --- | --- |
| orgs:create | N/A | Needs to be assigned globally. |

**Example Request:**

http Copy

```
POST /api/orgs HTTP/1.1
Accept: application/json
Content-Type: application/json

{
  "name":"New Org."
}
```

Note: The api will work in the following two ways

1. Need to set GF_USERS_ALLOW_ORG_CREATE=true
2. Set the config value users.allow_org_create to true in ini file

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{
  "orgId":"1",
  "message":"Organization created"
}
```

## Search all Organizations

`GET /api/orgs?perpage=10&page=1`

Only works with Basic Authentication (username and password), see introduction.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope | Note |
| --- | --- | --- |
| orgs:read | N/A | Needs to be assigned globally. |

**Example Request**:

```http
GET /api/orgs HTTP/1.1
Accept: application/json
Content-Type: application/json
```

Note: The api will only work when you pass the admin name and password to the request HTTP URL, like http://admin:admin@localhost:3000/api/orgs

Default value for the `perpage` parameter is `1000` and for the `page` parameter is `0`.

**Example Response**:

```http
```

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id":1,
    "name":"Main Org."
  }
]
```

## Update Organization

`PUT /api/orgs/:orgId`

Update Organization, fields *Address 1*, *Address 2*, *City* are not implemented yet. Only works with Basic Authentication (username and password), see introduction.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| orgs:write | N/A |

**Example Request**:

http Copy

```
PUT /api/orgs/1 HTTP/1.1
Accept: application/json
Content-Type: application/json

{
  "name":"Main Org 2."
}
```

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Organization updated"}
```

## Delete Organization

`DELETE /api/orgs/:orgId`

Only works with Basic Authentication (username and password), see introduction.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| orgs:delete | N/A |

**Example Request**:

```http
DELETE /api/orgs/1 HTTP/1.1
Accept: application/json
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Organization deleted"}
```

## Get Users in Organization

`GET /api/orgs/:orgId/users`

Only works with Basic Authentication (username and password), see introduction.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| org.users:read | users:* |

### Example Request:

```http
GET /api/orgs/1/users HTTP/1.1
Accept: application/json
Content-Type: application/json
```

Note: The api will only work when you pass the admin name and password to the request HTTP URL, like http://admin:admin@localhost:3000/api/orgs/1/users

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json
[
  {
    "orgId":1,
    "userId":1,
    "email":"admin@mygraf.com",
    "login":"admin",
    "role":"Admin"
  }
]
```

## Add User in Organization

`POST /api/orgs/:orgId/users`

Only works with Basic Authentication (username and password), see introduction.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| org.users:add | users:* |

**Example Request:**

```http
POST /api/orgs/1/users HTTP/1.1
Accept: application/json
Content-Type: application/json

{
  "loginOrEmail":"user",
  "role":"Viewer"
}
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"User added to organization", "userId": 1}
```

## Update Users in Organization

`PATCH /api/orgs/:orgId/users/:userId`

Only works with Basic Authentication (username and password), see introduction.

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| org.users:write | users:* |

**Example Request:**

```http
PATCH /api/orgs/1/users/2 HTTP/1.1
Accept: application/json
Content-Type: application/json

{
  "role":"Admin"
}
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Organization user updated"}
```

## Delete User in Organization

`DELETE /api/orgs/:orgId/users/:userId`

Only works with Basic Authentication (username and password), see introduction.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| org.users:remove | users:* |

**Example Request:**

```http
DELETE /api/orgs/1/users/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
```

Example Response:

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"User removed from organization"}
```

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Grafana OnCall organizations HTTP API

Docs

Manage organizations

Docs

Organizations

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

### Company

- Partnerships
- Our team
- Careers

### Localized content

- Japanese pages
- German pages
- French pages

Success stories

Newsroom

Contact us

Getting help

Merch

Open positions WE'RE HIRING

The Story of Grafana

Spanish pages

Portuguese pages

Grafana Cloud status

Legal and Security

Terms of Service

Privacy Policy

Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Frontend Settings API

## Get Settings

`GET /api/frontend/settings`

**Example Request:**

```http
GET /api/frontend/settings HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{
  "allowOrgCreate":true,
  "appSubUrl":"",
  "buildInfo":{
    "buildstamp":xxxxxx,
    "commit":"vyyyy",
```

```
    "version":"zzzzz"
  },
  "datasources":{
    "datasourcename":{
      "index":"grafana-dash",
      "meta":{
        "annotations":true,
        "module":"plugins/datasource/grafana/datasource",
        "name":"Grafana",
```


Expand code

# Login API

## Renew session based on remember cookie

`GET /api/login/ping`

### Example Request:

http Copy

```
GET /api/login/ping HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{"message": "Logged in"}
```

# Health API

## Returns health information about Grafana

`GET /api/health`

### Example Request

```http
GET /api/health
Accept: application/json
```

### Example Response:

```http
HTTP/1.1 200 OK

{
  "commit": "087143285",
  "database": "ok",
  "version": "5.1.3"
}
```

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Other HTTP API

Docs

Graphite HTTP API

Docs

Grafana Mimir HTTP API

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

### Company

### Localized content

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Playlist API

## Search Playlist

`GET /api/playlists`

Get all existing playlist for the current organization using pagination

**Example Request**:

```http
GET /api/playlists HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

Querystring Parameters:

These parameters are used as querystring parameters.

- **query** - Limit response to playlist having a name like this value.
- **limit** - Limit response to *X* number of playlist.

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json
[
```

```
  {
    "uid": "1",
    "name": "my playlist",
    "interval": "5m"
  }
]
```

## Get one playlist

`GET /api/playlists/:uid`

### Example Request:

http Copy

```
GET /api/playlists/1 HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response:

http Copy

```
HTTP/1.1 200
Content-Type: application/json
{
  "uid" : "1",
  "name": "my playlist",
  "interval": "5m",
  "items": [
    {
      "id": 1,
      "playlistUid": "1",
      "type": "dashboard_by_id",
      "value": "3",
      "order": 1,
      "title":"my third dashboard"
    },
    {
      "id": 2,
      "playlistUid": "1",
```

Expand code

```
"type": "dashboard_by_tag"
```


## Get Playlist items

`GET /api/playlists/:uid/items`

### Example Request:

http Copy

```
GET /api/playlists/1/items HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response:

http Copy


```
HTTP/1.1 200
Content-Type: application/json
[
  {
    "id": 1,
    "playlistUid": "1",
    "type": "dashboard_by_id",
    "value": "3",
    "order": 1,
    "title":"my third dashboard"
  },
  {
    "id": 2,
    "playlistUid": "1",
    "type": "dashboard_by_tag",
    "value": "myTag",
    "order": 2,
    "title":"my other dashboard"
  }
```


Expand code

## Create a playlist

`POST /api/playlists/`

### Example Request:

```http
PUT /api/playlists/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
  {
    "name": "my playlist",
    "interval": "5m",
    "items": [
      {
        "type": "dashboard_by_id",
        "value": "3",
        "order": 1,
        "title":"my third dashboard"
      },
      {
        "type": "dashboard_by_tag",
        "value": "myTag",
        "order": 2,
        "title":"my other dashboard"
```

Copy

Expand code

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json
  {
    "uid": "1",
    "name": "my playlist",
    "interval": "5m"
  }
```

Copy

## Update a playlist

`PUT /api/playlists/:uid`

**Example Request:**

```http
```

Copy

```
PUT /api/playlists/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
  {
    "name": "my playlist",
    "interval": "5m",
    "items": [
      {
        "playlistUid": "1",
        "type": "dashboard_by_id",
        "value": "3",
        "order": 1,
        "title":"my third dashboard"
      },
      {
        "playlistUid": "1",
        "type": "dashboard_by_tag",
        "value": "myTag",
```

Expand code

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json
{
  "uid" : "1",
  "name": "my playlist",
  "interval": "5m",
  "items": [
    {
      "id": 1,
      "playlistUid": "1",
      "type": "dashboard_by_id",
      "value": "3",
      "order": 1,
      "title":"my third dashboard"
    },
    {
      "id": 2,
```

Expand code

```
    "playlistUid": "1",
```

## Delete a playlist

`DELETE /api/playlists/:uid`

**Example Request:**

```http
DELETE /api/playlists/1 HTTP/1.1
Accept: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json
{}
```

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Playlist HTTP API

Docs

Manage playlists

Docs

Manage playlists

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report

Pricing

Log monitoring

## Company

Partnerships

Our team

Careers

Success stories

Newsroom

Contact us

Getting help

Merch

Open positions WE'RE HIRING

The Story of Grafana

## Localized content

Japanese pages

German pages

French pages

Spanish pages

Portuguese pages

---

Grafana Cloud status

Legal and Security

Terms of Service

Privacy Policy

Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Public Dashboard API

**NOTE**

If you're running Grafana Enterprise, you'll need to have specific permissions for some endpoints. Refer to Role-based access control permissions for more information.

## Create a public dashboard

`POST /api/dashboards/uid/:uid/public-dashboards/`

Creates a new public dashboard.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `dashboards.public:write` | `dashboards:uid:<dashboard UID>` |

### Example Request for new public dashboard:

```
http
```

Copy

```
POST /api/dashboards/uid/xCpsVuc4z/public-dashboards/ HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
    "uid": "cd56d9fd-f3d4-486d-afba-a21760e2acbe",
    "accessToken": "5c948bf96e6a4b13bd91975f9a2028b7",
    "timeSelectionEnabled": false,
    "isEnabled": true,
    "annotationsEnabled": false,
    "share": "public"
}
```

JSON Body schema:

- **uid** – Optional. Unique identifier when creating a public dashboard. If it's null, it will generate a new uid.
- **accessToken** – Optional. Unique access token. If it's null, it will generate a new access token.
- **timeSelectionEnabled** – Optional. Set to `true` to enable the time picker in the public dashboard. The default value is `false`.
- **isEnabled** – Optional. Set to `true` to enable the public dashboard. The default value is `false`.
- **annotationsEnabled** – Optional. Set to `true` to show annotations. The default value is `false`.
- **share** – Optional. Set the share mode. The default value is `public`.

**Example Response**:

http Copy

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

{
    "uid": "cd56d9fd-f3d4-486d-afba-a21760e2acbe",
    "dashboardUid": "xCpsVuc4z",
    "accessToken": "5c948bf96e6a4b13bd91975f9a2028b7",
```

```
    "createdBy": 1,
    "updatedBy": 1,
    "createdAt": "2023-09-05T15:48:21-03:00",
    "updatedAt": "2023-09-05T15:48:21-03:00",
    "timeSelectionEnabled": false,
    "isEnabled": false,
    "annotationsEnabled": false,
    "share": "public"
}
```

Status Codes:

- **200** – Created
- **400** – Errors (such as invalid json, missing or invalid fields, or dashboard is public)
- **401** – Unauthorized
- **403** – Access denied
- **404** – Dashboard not found

The error response body will have the following properties:

http Copy

```
HTTP/1.1 400 Bad request
Content-Type: application/json; charset=UTF-8
Content-Length: 107

{
    "statusCode": 400,
    "messageId": "publicdashboards.dashboardIsPublic",
    "message": "Dashboard is already public"
}
```

## Update a public dashboard

`PATCH /api/dashboards/uid/:uid/public-dashboards/:publicDashboardUid`

Will update the public dashboard given the specified unique identifier (uid).

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `dashboards.public:write` | `dashboards:uid:<dashboard UID>` |

**Example Request for updating a public dashboard:**

```http
PATCH /api/dashboards/uid/xCpsVuc4z/public-dashboards/cd56d9fd-f3d4-486d-afba-a21760e2acb
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
    "timeSelectionEnabled": false,
    "isEnabled": true,
    "annotationsEnabled": false,
    "share": "public"
}
```

JSON Body schema:

- **timeSelectionEnabled** – Optional. Set to `true` to enable the time picker in the public dashboard. The default value is `false`.
- **isEnabled** – Optional. Set to `true` to enable the public dashboard. The default value is `false`.
- **annotationsEnabled** – Optional. Set to `true` to show annotations. The default value is `false`.
- **share** – Optional. Set the share mode. The default value is `public`.

**Example Response:**

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

{
    "uid": "cd56d9fd-f3d4-486d-afba-a21760e2acbe",
```

```
    "dashboardUid": "xCpsVuc4z",
    "accessToken": "5c948bf96e6a4b13bd91975f9a2028b7",
    "createdBy": 1,
    "updatedBy": 1,
    "createdAt": "2023-09-05T15:48:21-03:00",
    "updatedAt": "2023-09-05T15:48:21-03:00",
    "timeSelectionEnabled": false,
    "isEnabled": false,
    "annotationsEnabled": false,
    "share": "public"
}
```


Status Codes:

- **200** – Updated
- **400** – Errors (such as invalid json, missing or invalid fields)
- **401** – Unauthorized
- **403** – Access denied
- **404** – Public dashboard not found

The error response body will have the following properties:

http Copy


```
HTTP/1.1 400 Bad request
Content-Type: application/json; charset=UTF-8
Content-Length: 107

{
    "statusCode": 400,
    "messageId": "publicdashboards.dashboardIsPublic",
    "message": "Dashboard is already public"
}
```


## Get public dashboard by dashboard uid

`GET /api/dashboards/uid/:uid/public-dashboards/`

Will return the public dashboard given the dashboard unique identifier (uid).

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `dashboards:read` | `dashboards:uid:<dashboard UID>` |

**Example Request:**

```http
GET /api/dashboards/uid/xCpsVuc4z/public-dashboards/ HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{
    "uid": "e71950f3-e7dd-4d1e-aa8a-a857bc5e7d64",
    "dashboardUid": "xCpsVuc4z",
    "accessToken": "dab10f3a4fbb4342a602b03079c7ed64",
    "createdBy": 1,
    "updatedBy": 1,
    "createdAt": "2023-09-05T15:48:21-03:00",
    "updatedAt": "2023-09-05T15:48:21-03:00",
    "timeSelectionEnabled": false,
    "isEnabled": false,
    "annotationsEnabled": false,
    "share": "public"
}
```

Status Codes:

- **200** – Found
- **401** – Unauthorized

- **403** – Access denied
- **404** – Not found

## Delete public dashboard by dashboard uid and public dashboard uid

`DELETE /api/dashboards/uid/:uid/public-dashboards/:publicDashboardUid`

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

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `dashboards.public:write` | `dashboards:uid:<dashboard UID>` |

**Example Request**:

```http
DELETE /api/dashboards/uid/xCpsVuc4z/public-dashboards/cd56d9fd-f3d4-486d-afba-a21760e2a
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

Status Codes:

- **200** – Deleted
- **401** – Unauthorized
- **403** – Access denied

## Get a list of all public dashboards with pagination

`GET /api/dashboards/public-dashboards`

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `dashboards:read` | `dashboards:uid:<dashboard UID>` |

### Example Request:

```http
GET /api/dashboards/public-dashboards?perpage=2&page=3 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json

{
    "publicDashboards": [
        {
            "uid": "e9f29a3c-fcc3-4fc5-a690-ae39c97d24ba",
            "accessToken": "6c13ec1997ba48c5af8c9c5079049692",
            "title": "Datasource Shared Queries",
            "dashboardUid": "d2f21d0a-76c7-47ec-b5f3-9dda16e5a996",
            "isEnabled": true
        },
        {
            "uid": "a174f604-6fe7-47de-97b4-48b7e401b540",
            "accessToken": "d1fcff345c0f45e8a78c096c9696034a",
            "title": "Datasource with template variables",
            "dashboardUid": "51DiOw0Vz",
            "isEnabled": true
        }
```

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Graphite HTTP API

Docs

Grafana Mimir HTTP API

Docs

Use the HTTP API to manage alerting resources

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

**Products**

Grafana Cloud

Grafana Enterprise Stack

**Open Source**

Grafana

Grafana Alloy

**Learn**

Documentation

Grafana Labs blog

Grafana visualizations
Plugins
Grafana Cloud Logs
Grafana Cloud Metrics
Grafana Cloud Traces
Grafana Cloud Profiles
AI/ML tools for observability
Application Observability
Frontend Observability
Grafana Cloud Asserts
Grafana IRM
Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Grafana Beyla
Grafana Faro
Grafana k6
Grafana Loki
Grafana Mimir
Grafana OnCall
Grafana Pyroscope
Grafana Tanka
Grafana Tempo
Graphite
OpenTelemetry
Prometheus
Dashboard templates
Downloads
GitHub

Webinars and videos
Tutorials
Workshops
Events
Community
Community forums
Community Slack
Grafana Champions
Community organizers
Observability Survey
OSS vs Cloud
Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Query and resource caching API

**NOTE**

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information.

## Enable caching for a data source

`POST /api/datasources/:dataSourceUID/cache/enable`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| datasources.caching:write | datasources:* |

## Examples

### Example Request:

http Copy

```
POST /api/datasources/jZrmlLCGka/cache/enable HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
   "message": "Data source cache enabled",
   "dataSourceID": 1,
   "dataSourceUID": "jZrmlLCGka",
   "enabled": true,
   "ttlQueriesMs": 300000,
   "ttlResourcesMs": 300000,
   "useDefaultTTL": true,
   "defaultTTLMs": 300000,
   "created": "2023-04-21T11:49:22-04:00",
   "updated": "2023-04-24T16:30:29-04:00"
}
```

### Status codes

| Code | Description |
|---|---|
| 200 | Cache was successfully enabled for the data source |
| 500 | Unexpected error. Refer to the body and/or server logs for more details. |

## Disable caching for a data source

`POST /api/datasources/:dataSourceUID/cache/disable`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| datasources.caching:write | datasources:* |

## Examples

### Example Request:

http

Copy

```
POST /api/datasources/jZrmlLCGka/cache/disable HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response:

http

Copy

```
HTTP/1.1 200
Content-Type: application/json

{
   "message": "Data source cache disabled",
   "dataSourceID": 1,
   "dataSourceUID": "jZrmlLCGka",
   "enabled": false,
   "ttlQueriesMs": 300000,
   "ttlResourcesMs": 300000,
   "useDefaultTTL": true,
   "defaultTTLMs": 0,
   "created": "2023-04-21T11:49:22-04:00",
   "updated": "2023-04-24T16:30:31-04:00"
}
```

## Status codes

| Code | Description |
| --- | --- |
| 200 | Cache was successfully enabled for the data source |
| 500 | Unexpected error. Refer to the body and/or server logs for more details. |

## Clean cache for all data sources

`POST /api/datasources/:dataSourceUID/cache/clean`

Will clean cached data for *all* data sources with caching enabled. The `dataSourceUID` specified will only be used to return the configuration for that data source.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| datasources.caching:write | datasources:* |

## Examples

### Example Request:

```http
POST /api/datasources/jZrmlLCGka/cache/clean HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json
```

```
{
   "message": "Data source cache cleaned",
   "dataSourceID": 1,
   "dataSourceUID": "jZrmlLCGka",
   "enabled": false,
   "ttlQueriesMs": 300000,
   "ttlResourcesMs": 300000,
   "useDefaultTTL": true,
   "defaultTTLMs": 0,
   "created": "2023-04-21T11:49:22-04:00",
   "updated": "2023-04-24T16:30:31-04:00"
}
```


### Status codes

| Code | Description |
| --- | --- |
| 200 | Cache was successfully enabled for the data source |
| 500 | Unexpected error. Refer to the body and/or server logs for more details. |

## Update cache configuration for a data source

`POST /api/datasources/:dataSourceUID/cache`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| datasources.caching:write | datasources:* |

## Examples

### Example Request:

```
http
```

Copy

```
POST /api/datasources/jZrmlLCGka/cache HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
   "dataSourceID": 1,
   "dataSourceUID": "jZrmlLCGka",
   "enabled": true,
   "useDefaultTTL": false,
   "ttlQueriesMs": 60000,
   "ttlResourcesMs": 300000,
}
```

## JSON Body Schema

| Field name | Data type | Description |
|---|---|---|
| dataSourceID | number | The ID of the data source to configure. |
| dataSourceUID | string | The UID of the data source to configure. |
| enabled | boolean | Whether or not to enable caching for this data source. |
| useDefaultTTL | boolean | Whether the configured default TTL (Time-To-Live) should be used for both query and resource caching, instead of the user-specified values. |
| ttlQueriesMs | number | The TTL to use for query caching, in milliseconds. |
| ttlResourcesMs | number | The TTL to use for resource caching, in milliseconds. |

## Example Response:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
   "message": "Data source cache settings updated",
```

```
    "dataSourceID": 1,
    "dataSourceUID": "jZrmlLCGka",
    "enabled": true,
    "useDefaultTTL": false,
    "ttlQueriesMs": 60000,
    "ttlResourcesMs": 300000,
    "defaultTTLMs": 300000,
    "created": "2023-04-21T11:49:22-04:00",
    "updated": "2023-04-24T17:03:40-04:00"
}
```


### Status codes

| Code | Description |
| --- | --- |
| 200 | Cache was successfully enabled for the data source |
| 400 | Request errors (invalid json, missing or invalid fields, etc) |
| 500 | Unexpected error. Refer to the body and/or server logs for more details. |

## Get cache configuration for a data source

`GET /api/datasources/:dataSourceUID/cache`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| datasources.caching:read | datasources:* |

## Examples

### Example Request:

```
http
```

Copy

```
GET /api/datasources/jZrmlLCGka/cache HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
   "message": "Data source cache settings loaded",
   "dataSourceID": 1,
   "dataSourceUID": "jZrmlLCGka",
   "enabled": true,
   "useDefaultTTL": false,
   "ttlQueriesMs": 60000,
   "ttlResourcesMs": 300000,
   "defaultTTLMs": 300000,
   "created": "2023-04-21T11:49:22-04:00",
   "updated": "2023-04-24T17:03:40-04:00"
}
```

### Status codes

| Code | Description |
|---|---|
| 200 | Cache was successfully enabled for the data source |
| 500 | Unexpected error. Refer to the body and/or server logs for more details. |

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Redshift and Athena: Async query caching

Docs

Data source management

Docs

Monitor Memcached with Prometheus and Grafana Cloud

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

**Products**

Grafana Cloud
Grafana Enterprise Stack
Grafana visualizations
Plugins
Grafana Cloud Logs
Grafana Cloud Metrics

**Open Source**

Grafana
Grafana Alloy
Grafana Beyla
Grafana Faro
Grafana k6
Grafana Loki

**Learn**

Documentation
Grafana Labs blog
Webinars and videos
Tutorials
Workshops
Events

Grafana Cloud Traces
Grafana Cloud Profiles
AI/ML tools for observability
Application Observability
Frontend Observability
Grafana Cloud Asserts
Grafana IRM
Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Grafana Mimir
Grafana OnCall
Grafana Pyroscope
Grafana Tanka
Grafana Tempo
Graphite
OpenTelemetry
Prometheus
Dashboard templates
Downloads
GitHub

Community
Community forums
Community Slack
Grafana Champions
Community organizers
Observability Survey
OSS vs Cloud
Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Query history API

This API can be used to add queries to Query history. It requires that the user is logged in and that Query history feature is enabled in config file.

## Add query to Query history

`POST /api/query-history`

Adds query to query history.

**Example request:**

http Copy

```
POST /api/query-history HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
{
  "datasourceUid": "PE1C5CBDA0504A6A3",
  "queries": [
    {
        "refId": "A",
        "key": "Q-87fed8e3-62ba-4eb2-8d2a-4129979bb4de-0",
        "scenarioId": "csv_content",
        "datasource": {
            "type": "testdata",
```

```
            "uid": "PD8C576611E62080A"
        }
    }
]
}
```

JSON body schema:

- **datasourceUid** – Data source uid.
- **queries** – JSON of query or queries.

**Example response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json
{
  "result": {
    "uid": "Ahg678z",
    "datasourceUid": "PE1C5CBDA0504A6A3",
    "createdBy": 1,
    "createdAt": 1643630762,
    "starred": false,
    "comment": "",
    "queries": [
      {
        "refId": "A",
        "key": "Q-87fed8e3-62ba-4eb2-8d2a-4129979bb4de-0",
        "scenarioId": "csv_content",
        "datasource": {
            "type": "testdata",
            "uid": "PD8C576611E62080A"
        }
```

Expand code

Status codes:

- **200** – OK
- **400** - Errors (invalid JSON, missing or invalid fields)
- **401** – Unauthorized
- **500** – Internal error

## Query history search

`GET /api/query-history`

Returns a list of queries in the query history that matches the search criteria. Query history search supports pagination. Use the `limit` parameter to control the maximum number of queries returned; the default limit is 100. You can also use the `page` query parameter to fetch queries from any page other than the first one.

Query parameters:

- **datasourceUid** - Filter the query history for the selected data source. To perform an "AND" filtering with multiple data sources, specify the data source parameter using the following format: `datasourceUid=uid1&datasourceUid=uid2`.
- **searchString** – Filter the query history based on the content.
- **sort** - Specify the sorting order. Sorting can be `time-asc` or `time-desc`. The default is `time-desc`.
- **onlyStarred** - Search for queries that are starred. Defaults to `false`.
- **page** - Search supports pagination. Specify which page number to return. Use the limit parameter to specify the number of queries per page.
- **limit** - Limits the number of returned query history items per page. The default is 100 queries per page.
- **from/to** - Specifies time range for the query history search. The time can be either epoch timestamps in milliseconds or relative using Grafana time units. For example, now-5m.

**Example request for query history search**:

```http
GET /api/query-history?datasourceUid="PE1C5CBDA0504A6A3"&datasourceUid="FG1C1CBDA0504A6E
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example response for query history search**:

```http
HTTP/1.1 200
Content-Type: application/json
```

```
{
  "result": {
    "totalCount": 150,
    "page": 1,
    "perPage": 100
    "queryHistory":[{
    "uid": "Ahg678z",
    "datasourceUid": "PE1C5CBDA0504A6A3",
    "createdBy": 1,
    "createdAt": 1643630762,
    "starred": false,
    "comment": "",
    "queries": [
      {
        "refId": "A",
        "key": "Q-87fed8e3-62ba-4eb2-8d2a-4129979bb4de-0",
```


Expand code

Status codes:

- **200** – OK
- **401** – Unauthorized
- **500** – Internal error

## Delete query from Query history by UID

`DELETE /api/query-history/:uid`

Deletes the query in query history that matches the specified uid. It requires that the user is logged in and that Query history feature is enabled in config file.

**Example Request:**

http Copy

```
DELETE /api/query-history/P8zM2I1nz HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
    "message": "Query deleted",
    "id": 28
}
```

Status codes:

- **200** – OK
- **401** – Unauthorized
- **500** – Internal error

## Update comment of query in Query history by UID

`PATCH /api/query-history/:uid`

Updates comment of a query with a specific uid that is stored in the query history.

Query parameters:

- **comment** – New comment that will be added to the specified query.

**Example Request**:

http Copy

```
PATCH /api/query-history/P8zM2I1nz HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
{
  "comment": "Debugging query",
}
```

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json
```

```
{
  "result": {
    "uid": "P8zM2I1nz",
    "datasourceUid": "PE1C5CBDA0504A6A3",
    "createdBy": 1,
    "createdAt": 1643630762,
    "starred": false,
    "comment": "Debugging query",
    "queries": [
      {
        "refId": "A",
        "key": "Q-87fed8e3-62ba-4eb2-8d2a-4129979bb4de-0",
        "scenarioId": "csv_content",
        "datasource": {
            "type": "testdata",
            "uid": "PD8C576611E62080A"
```


Expand code

Status codes:

- **200** – OK
- **400** - Errors (invalid JSON, missing or invalid fields)
- **401** – Unauthorized
- **500** – Internal error

## Star query in Query history

`POST /api/query-history/star/:uid`

Stars query in query history.

**Example request:**

http Copy

```
POST /api/query-history/star/P8zM2I1nz HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json
{
  "result": {
    "uid": "P8zM2I1nz",
    "datasourceUid": "PE1C5CBDA0504A6A3",
    "createdBy": 1,
    "createdAt": 1643630762,
    "starred": false,
    "comment": "Debugging query",
    "queries": [
      {
        "refId": "A",
        "key": "Q-87fed8e3-62ba-4eb2-8d2a-4129979bb4de-0",
        "scenarioId": "csv_content",
        "datasource": {
            "type": "testdata",
            "uid": "PD8C576611E62080A"
        }
```

Expand code

Status codes:

- **200** – OK
- **401** – Unauthorized
- **500** – Internal error

## Unstar query in Query history

`DELETE /api/query-history/star/:uid`

Removes stars from query in query history.

**Example request:**

http Copy

```
DELETE /api/query-history/star/P8zM2I1nz  HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example response:**

http

Copy

```
HTTP/1.1 200
Content-Type: application/json
{
  "result": {
    "uid": "P8zM2I1nz",
    "datasourceUid": "PE1C5CBDA0504A6A3",
    "createdBy": 1,
    "createdAt": 1643630762,
    "starred": false,
    "comment": "Debugging query",
    "queries": [
      {
        "refId": "A",
        "key": "Q-87fed8e3-62ba-4eb2-8d2a-4129979bb4de-0",
        "scenarioId": "csv_content",
        "datasource": {
            "type": "testdata",
            "uid": "PD8C576611E62080A"
        }
```

Expand code

Status codes:

- **200** – OK
- **401** – Unauthorized
- **500** – Internal error

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Preferences API

Docs

Graphite HTTP API

Docs

Grafana Mimir HTTP API

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud

Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Prometheus
Dashboard templates
Downloads
GitHub

Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# RBAC API

Role-based access control API is only available in Grafana Cloud or Grafana Enterprise. Read more about Grafana Enterprise.

The API can be used to create, update, delete, get, and list roles.

To check which basic or fixed roles have the required permissions, refer to RBAC role definitions.

## Get status

`GET /api/access-control/status`

Returns an indicator to check if role-based access control is enabled or not.

### Required permissions

| Action | Scope |
| --- | --- |
| status:accesscontrol | services:accesscontrol |

### Example request

http Copy

```
GET /api/access-control/status
Accept: application/json
Content-Type: application/json
```

### Example response

http Copy

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

{
  "enabled": true
}
```

### Status codes

| Code | Description |
| --- | --- |
| 200 | Returned a flag indicating if the role-based access control is enabled or no. |
| 403 | Access denied |
| 404 | Not found, an indication that role-based access control is not available at all. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Create and manage custom roles

### Get all roles

`GET /api/access-control/roles`

Gets all existing roles. The response contains all global and organization local roles, for the organization which user is signed in.

Query Parameters:

- `includeHidden`: Optional. Set to `true` to include roles that are `hidden`.

## Required permissions

| Action | Scope |
| --- | --- |
| roles:read | roles:* |

## Example request

http

Copy

```
GET /api/access-control/roles
Accept: application/json
Content-Type: application/json
```

## Example response

http

Copy

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

[
    {
        "version": 3,
        "uid": "XvHQJq57z",
        "name": "fixed:reports:reader",
        "displayName": "Report reader",
        "description": "Read all reports and shared report settings.",
        "group": "Reports",
        "updated": "2021-11-19T10:48:00+01:00",
        "created": "2021-11-19T10:48:00+01:00",
        "global": false
    },
    {
        "version": 5,
        "uid": "vi9mlLjGz",
        "name": "fixed:datasources.permissions:writer",
```

Expand code

### Status codes

| Code | Description |
| --- | --- |
| 200 | Global and organization local roles are returned. |
| 403 | Access denied |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Get a role

`GET /api/access-control/roles/:uid`

Get a role for the given UID.

### Required permissions

| Action | Scope |
| --- | --- |
| roles:read | roles:* |

### Example request

http Copy

```
GET /api/access-control/roles/PYnDO3nMk
Accept: application/json
Content-Type: application/json
```

### Example response

http Copy

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

```
{
    "version": 4,
    "uid": "6dNwJq57z",
    "name": "fixed:reports:writer",
    "displayName": "Report writer",
    "description": "Create, read, update, or delete all reports and shared report setting
    "group": "Reports",
    "permissions": [
        {
            "action": "reports:delete",
            "scope": "reports:*",
            "updated": "2021-11-19T10:48:00+01:00",
            "created": "2021-11-19T10:48:00+01:00"
        },
        {
```


Expand code

### Status codes

| Code | Description |
| --- | --- |
| 200 | Role is returned. |
| 403 | Access denied. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Create a new custom role

`POST /api/access-control/roles`

Creates a new custom role and maps given permissions to that role. Note that roles with the same prefix as Fixed roles can't be created.

### Required permissions

`permissions:type:delegate` scope ensures that users can only create custom roles with the same, or a subset of permissions which the user has. For example, if a user does not have required permissions for creating users, they won't be able to create a custom role which allows to do that. This is done to prevent escalation of privileges.

| Action | Scope |
| --- | --- |
| roles:write | permissions:type:delegate |

## Example request

http

Copy

```
POST /api/access-control/roles
Accept: application/json
Content-Type: application/json

{
    "version": 1,
    "uid": "jZrmlLCGka",
    "name": "custom:delete:roles",
    "displayName": "custom delete roles",
    "description": "My custom role which gives users permissions to delete roles",
    "group":"My Group",
    "displayName": "My Custom Role",
    "global": false,
    "permissions": [
        {
            "action": "roles:delete",
            "scope": "permissions:type:delegate"
        }
    ]
```

Expand code

## JSON body schema

| Field Name | Date Type | Required | Description |
| --- | --- | --- | --- |
| uid | string | No | UID of the role. If not present, the UID will be automatically created for you and returned in response. Refer to the Custom roles for more information. |
| global | boolean | No | A flag indicating if the role is global or not. If set to `false`, the default org ID of the authenticated user will be used from the request. |

| Field Name | Date Type | Required | Description |
| --- | --- | --- | --- |
| version | number | No | Version of the role. If not present, version 0 will be assigned to the role and returned in the response. Refer to the Custom roles for more information. |
| name | string | Yes | Name of the role. Refer to Custom roles for more information. |
| description | string | No | Description of the role. |
| displayName | string | No | Display name of the role, visible in the UI. |
| group | string | No | The group name the role belongs to. |
| hidden | boolean | No | Specify whether the role is hidden or not. If set to `true`, then the role does not show in the role picker. It will not be listed by API endpoints unless explicitly specified. |
| permissions | Permission | No | If not present, the role will be created without any permissions. |

### Permission

| Field Name | Data Type | Required | Description |
| --- | --- | --- | --- |
| action | string | Yes | Refer to Custom role actions and scopes for full list of available actions. |
| scope | string | No | If not present, no scope will be mapped to the permission. Refer to Custom role actions and scopes for full list of available scopes. |

## Example response

http Copy

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

{
    "version": 2,
    "uid": "jZrmlLCGka",
    "name": "custom:delete:create:roles",
```

```
    "displayName": "custom delete create roles",
    "description": "My custom role which gives users permissions to delete and create ro
    "group":"My Group",
    "displayName": "My Custom Role",
    "global": false,
    "permissions": [
        {
            "action": "roles:delete",
            "scope": "permissions:type:delegate",
            "updated": "2021-05-13T23:19:46+02:00",
            "created": "2021-05-13T23:19:46+02:00"
```

Expand code

## Create role validation errors

Permission validation only occurs when permission validation is enabled (`rbac.permission_validation_enabled = true`).

> It has been enabled by default since Grafana 10.2.

### Invalid action

The following example shows a request with an invalid action. The action `serviceaccounts.permissions:reader` is not a valid action. The valid action should be `serviceaccounts.permissions:read`.

http Copy

```
POST /api/access-control/roles HTTP/1.1
Content-Type: application/json
{
    "Name": "Read Service Account with id 6",
    "Permissions": [
            {
            "action": "serviceaccounts.permissions:reader",
            "scope": "serviceaccounts:uid:6"
        }
    ]
}
```

http Copy

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
{
    "extra": {
        "validationError": "the provided action was not found in the list of valid actio
    },
    "message": "Permission contains an invalid action",
    "messageId": "accesscontrol.permission-invalid-action",
    "statusCode": 400,
    "traceID": ""
}
```

### Invalid scope

The following example shows a request with an invalid scope. The scope `serviceaccounts:serviceaccount6` is not a valid scope for the action `serviceaccounts.permissions:read`. The valid scopes for this action are `*`, `serviceaccounts:*` and `serviceaccounts:id:*`.

http Copy

```
POST /api/access-control/roles HTTP/1.1
Content-Type: application/json
{
    "Name": "Read Service Account with id 6",
    "Permissions": [
            {
            "action": "serviceaccounts.permissions:read",
            "scope": "serviceaccounts:serviceaccount6"
        }
    ]
}
```

http Copy

```
HTTP/1.1 400 Bad Request
Content-Type: application/json
{
    "extra": {
        "validationError": "unknown scope: serviceaccounts:serviceaccount6 for action: s
    },
```

```
    "message": "Invalid scope",
    "messageId": "accesscontrol.permission-invalid-scope",
    "statusCode": 400,
    "traceID": ""
}
```


### Status codes

| Code | Description |
| --- | --- |
| 200 | Role is updated. |
| 400 | Bad request (invalid json, missing content-type, missing or invalid fields, etc.). |
| 403 | Access denied (one of the specified permissions is not assigned to the requester) |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Update a role

`PUT /api/access-control/roles/:uid`

Update the role with the given UID, and its permissions. The operation is idempotent and all permissions of the role will be replaced based on the request content. You need to increment the version of the role with each update, otherwise the request will fail.

You can update `custom` roles and `basic` roles permissions. However `fixed` roles cannot be updated.

### Required permissions

`permissions:type:delegate` scope ensures that users can only update custom roles with the same, or a subset of permissions which the user has. For example, if a user does not have required permissions for creating users, they won't be able to update a custom role which allows to do that. This is done to prevent escalation of privileges.

| Action | Scope |
| --- | --- |
| roles:write | permissions:type:delegate |

## Example request

http

Copy

```
PUT /api/access-control/roles/jZrmlLCGka
Accept: application/json
Content-Type: application/json

{
    "version": 3,
    "name": "custom:delete:write:roles",
    "displayName": "custom delete write roles",
    "description": "My custom role which gives users permissions to delete and write rol
    "group":"My Group",
    "displayName": "My Custom Role",
    "global": false,
    "permissions": [
        {
            "action": "roles:delete",
            "scope": "permissions:type:delegate"
        },
        {
            "action": "roles:write",
```

Expand code

## JSON body schema

| Field Name | Data Type | Required | Description |
|---|---|---|---|
| version | number | Yes | Version of the role. Must be incremented for update to work. |
| name | string | Yes | Name of the role. |
| description | string | No | Description of the role. |
| displayName | string | No | Display name of the role, visible in the UI. |
| group | string | No | The group name the role belongs to. |
| hidden | boolean | No | Specify whether the role is hidden or not. If set to `true`, then the role does not show in the role picker. It will not be listed by API endpoints unless explicitly specified. |

| Field Name | Data Type | Required | Description |
| --- | --- | --- | --- |
| permissions | List of Permissions | No | The full list of permissions for the role after the update. |

### Permission

| Field Name | Data Type | Required | Description |
| --- | --- | --- | --- |
| action | string | Yes | Refer to Custom role actions and scopes for full list of available actions. |
| scope | string | No | If not present, no scope will be mapped to the permission. Refer to Custom role actions and scopes for full list of available scopes. |

## Example response

http Copy

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

{
    "version":3,
    "uid":"jZrmlLCGka",
    "name":"custom:delete:write:roles",
    "displayName":"custom delete write roles",
    "description":"My custom role which gives users permissions to delete and write roles
    "group":"My Group",
    "displayName": "My Custom Role",
    "permissions":[
        {
            "action":"roles:delete",
            "scope":"permissions:type:delegate",
            "updated":"2021-08-06T18:27:40+02:00",
            "created":"2021-08-06T18:27:40+02:00"
        },
        {
```

Expand code

### Update role validation errors

Permission validation only occurs when permission validation is enabled (`rbac.permission_validation_enabled = true`).

> It has been enabled by default since Grafana 10.2.

For more information, refer to Create role validation errors.

### Status codes

| Code | Description |
| --- | --- |
| 200 | Role is updated. |
| 400 | Bad request (invalid json, missing content-type, missing or invalid fields, etc.). |
| 403 | Access denied (one of the specified permissions is not assigned to the requester) |
| 404 | Role was not found to update. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Delete a custom role

`DELETE /api/access-control/roles/:uid?force=false`

Delete a role with the given UID, and it's permissions. If the role is assigned, the deletion operation will fail, unless the `force` query param is set to `true`, and in that case all assignments will also be deleted.

### Required permissions

`permissions:type:delegate` scope ensures that users can only delete a custom role with the same, or a subset of permissions which the user has. For example, if a user does not have required permissions for creating users, they won't be able to delete a custom role which allows to do that.

| Action | Scope |
| --- | --- |
| roles:delete | permissions:type:delegate |

## Example request

```http
DELETE /api/access-control/roles/jZrmlLCGka?force=true&global=false
Accept: application/json
```

## Query parameters

| Param | Type | Required | Description |
| --- | --- | --- | --- |
| force | boolean | No | When set to `true`, the role will be deleted with all its assignments. |
| global | boolean | No | A flag indicating if the role is global or not. If set to false, the default org ID of the authenticated user will be used from the request. Refer to the About RBAC for more information. |

## Example response

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

{
    "message": "Role deleted"
}
```

## Status codes

| Code | Description |
| --- | --- |
| 200 | Role is deleted. |
| 400 | Bad request (invalid json, missing content-type, missing or invalid fields, etc.). |

| Code | Description |
| --- | --- |
| 403 | Access denied |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

# Create and remove user role assignments

## List roles assigned to a user

`GET /api/access-control/users/:userId/roles`

Lists the roles that have been directly assigned to a given user. The list does not include basic roles (Viewer, Editor, Admin or Grafana Admin), and it does not include roles that have been inherited from a team.

Query Parameters:

- `includeHidden`: Optional. Set to `true` to include roles that are `hidden`.

### Required permissions

| Action | Scope |
| --- | --- |
| users.roles:read | users:id:`<user ID>` |

### Example request

```http
GET /api/access-control/users/1/roles
Accept: application/json
```

### Example response

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

```
[
    {
        "version": 4,
        "uid": "6dNwJq57z",
        "name": "fixed:reports:writer",
        "displayName": "Report writer",
        "description": "Create, read, update, or delete all reports and shared report se
        "group": "Reports",
        "updated": "2021-11-19T10:48:00+01:00",
        "created": "2021-11-19T10:48:00+01:00",
        "global": false
    }
]
```


### Status codes

| Code | Description |
| --- | --- |
| 200 | Set of assigned roles is returned. |
| 403 | Access denied. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## List your permissions

`GET /api/access-control/user/permissions`

Lists the permissions granted to the signed in user.

### Required permissions

No permission is required.

### Query parameters

| Param | Type | Required | Description |
| --- | --- | --- | --- |
| reloadcache | boolean | No | A flag to reload the permission cache. |

### Example request

```http
GET /api/access-control/user/permissions
Accept: application/json
```

### Example response

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

{
  "dashboards:read": ["dashboards:uid:70KrY6IVz"],
  "dashboards:write": ["dashboards:uid:70KrY6IVz"],
  "datasources.id:read": ["datasources:*"],
  "datasources:read": ["datasources:*"],
  "datasources:explore": [""],
  "datasources:query": ["datasources:uid:grafana"],
  "datasources:read": ["datasources:uid:grafana"],
  "orgs:read": [""]
}
```

### Status codes

| Code | Description |
| --- | --- |
| 200 | Set of assigned permissions is returned. |
| 403 | Access denied. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## List permissions assigned to a user

`GET /api/access-control/users/:userId/permissions`

Lists the permissions granted to a given user.

## Required permissions

| Action | Scope |
| --- | --- |
| users.permissions:read | users:id:`<user ID>` |

## Example request

http Copy

```
GET /api/access-control/users/1/permissions
Accept: application/json
```

## Example response

http Copy

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

[
    {
        "action": "ldap.status:read",
        "scope": ""
    },
    {
        "action": "ldap.user:read",
        "scope": ""
    }
]
```

## Status codes

| Code | Description |
| --- | --- |
| 200 | Set of assigned permissions is returned. |
| 403 | Access denied. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Add a user role assignment

`POST /api/access-control/users/:userId/roles`

Assign a role to a specific user.

For bulk updates consider Set user role assignments.

### Required permissions

`permissions:type:delegate` scope ensures that users can only assign roles which have same, or a subset of permissions which the user has. For example, if a user does not have required permissions for creating users, they won't be able to assign a role which will allow to do that. This is done to prevent escalation of privileges.

| Action | Scope |
| --- | --- |
| users.roles:add | permissions:type:delegate |

### Example request

http Copy

```
POST /api/access-control/users/1/roles
Accept: application/json
Content-Type: application/json

{
    "global": false,
    "roleUid": "XvHQJq57z"
}
```

### JSON body schema

| Field Name | Data Type | Required | Description |
| --- | --- | --- | --- |
| roleUid | string | Yes | UID of the role. |
| global | boolean | No | A flag indicating if the assignment is global or not. If set to `false`, the default org ID of the authenticated user will be used from the request to create organization local assignment. |

### Example response

http Copy

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

{
    "message": "Role added to the user."
}
```

### Status codes

| Code | Description |
| --- | --- |
| 200 | Role is assigned to a user. |
| 403 | Access denied. |
| 404 | Role not found. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Remove a user role assignment

`DELETE /api/access-control/users/:userId/roles/:roleUID`

Revoke a role from a user.

For bulk updates consider Set user role assignments.

## Required permissions

`permissions:type:delegate` scope ensures that users can only unassign roles which have same, or a subset of permissions which the user has. For example, if a user does not have required permissions for creating users, they won't be able to unassign a role which will allow to do that. This is done to prevent escalation of privileges.

| Action | Scope |
| --- | --- |
| users.roles:remove | permissions:type:delegate |

## Query parameters

| Param | Type | Required | Description |
| --- | --- | --- | --- |
| global | boolean | No | A flag indicating if the assignment is global or not. If set to `false`, the default org ID of the authenticated user will be used from the request to remove assignment. |

## Example request

http Copy

```
DELETE /api/access-control/users/1/roles/AFUXBHKnk
Accept: application/json
```

## Example response

http Copy

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

{
```

```
    "message": "Role removed from user."
}
```

### Status codes

| Code | Description |
| --- | --- |
| 200 | Role is unassigned. |
| 403 | Access denied. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Set user role assignments

`PUT /api/access-control/users/:userId/roles`

Update the user's role assignments to match the provided set of UIDs. This will remove any assigned roles that aren't in the request and add roles that are in the set but are not already assigned to the user.

If you want to add or remove a single role, consider using Add a user role assignment or Remove a user role assignment instead.

### Required permissions

`permissions:type:delegate` scope ensures that users can only assign or unassign roles which have same, or a subset of permissions which the user has. For example, if a user does not have required permissions for creating users, they won't be able to assign or unassign a role which will allow to do that. This is done to prevent escalation of privileges.

| Action | Scope |
| --- | --- |
| users.roles:add | permissions:type:delegate |
| users.roles:remove | permissions:type:delegate |

### Example request

```
http
```

Copy

```
PUT /api/access-control/users/1/roles
Accept: application/json
Content-Type: application/json

{
    "global": false,
    "roleUids": [
        "ZiHQJq5nk",
        "GzNQ1357k"
    ]
}
```

## JSON body schema

| Field Name | Date Type | Required | Description |
| --- | --- | --- | --- |
| global | boolean | No | A flag indicating if the assignment is global or not. If set to `false`, the default org ID of the authenticated user will be used from the request. |
| roleUids | list | Yes | List of role UIDs. |
| includeHidden | boolean | No | Specify whether the hidden role assignments should be updated. |

## Example response

http

Copy

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

{
    "message": "User roles have been updated."
}
```

## Status codes

| Code | Description |
| --- | --- |
| 200 | Roles have been assigned. |
| 403 | Access denied. |
| 404 | Role not found. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

# Create and remove service account role assignments

## List roles assigned to a service account

`GET /api/access-control/users/:serviceAccountId/roles`

Lists the roles that have been directly assigned to a given service account. The list does not include basic roles (Viewer, Editor, Admin or Grafana Admin).

Query Parameters:

- `includeHidden`: Optional. Set to `true` to include roles that are `hidden`.

### Required permissions

| Action | Scope |
| --- | --- |
| users.roles:read | users:id:`<service account ID>` |

### Example request

http Copy

```
GET /api/access-control/users/1/roles
Accept: application/json
```

### Example response

http Copy

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

[
    {
        "version": 4,
        "uid": "6dNwJq57z",
        "name": "fixed:reports:writer",
        "displayName": "Report writer",
        "description": "Create, read, update, or delete all reports and shared report se
        "group": "Reports",
        "updated": "2021-11-19T10:48:00+01:00",
        "created": "2021-11-19T10:48:00+01:00",
        "global": false
    }
]
```

### Status codes

| Code | Description |
| --- | --- |
| 200 | Set of assigned roles is returned. |
| 403 | Access denied. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## List permissions assigned to a service account

`GET /api/access-control/users/:serviceAccountId/permissions`

Lists the permissions that a given service account has.

### Required permissions

| Action | Scope |
| --- | --- |
| users.permissions:read | users:id:`<service account ID>` |

### Example request

```http
GET /api/access-control/users/1/permissions
Accept: application/json
```

### Example response

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

[
    {
        "action": "ldap.status:read",
        "scope": ""
    },
    {
        "action": "ldap.user:read",
        "scope": ""
    }
]
```

### Status codes

| Code | Description |
| --- | --- |
| 200 | Set of assigned permissions is returned. |
| 403 | Access denied. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Add a service account role assignment

`POST /api/access-control/users/:serviceAccountId/roles`

Assign a role to a specific service account.

For bulk updates consider Set service account role assignments.

## Required permissions

`permissions:type:delegate` scope ensures that users can only assign roles which have same, or a subset of permissions which the user has. For example, if a user does not have required permissions for creating users, they won't be able to assign a role which will allow to do that. This is done to prevent escalation of privileges.

| Action | Scope |
| --- | --- |
| users.roles:add | permissions:type:delegate |

## Example request

http

Copy

```
POST /api/access-control/users/1/roles
Accept: application/json
Content-Type: application/json

{
    "global": false,
    "roleUid": "XvHQJq57z"
}
```

## JSON body schema

| Field Name | Data Type | Required | Description |
| --- | --- | --- | --- |
| roleUid | string | Yes | UID of the role. |
| global | boolean | No | A flag indicating if the assignment is global or not. If set to `false`, the default org ID of the authenticated user will be used from the request to create organization local assignment. |

### Example response

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

{
    "message": "Role added to the user."
}
```

### Status codes

| Code | Description |
| --- | --- |
| 200 | Role is assigned to a user. |
| 403 | Access denied. |
| 404 | Role not found. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Remove a service account role assignment

`DELETE /api/access-control/users/:serviceAccountId/roles/:roleUID`

Revoke a role from a service account.

For bulk updates consider Set service account role assignments.

### Required permissions

`permissions:type:delegate` scope ensures that users can only unassign roles which have same, or a subset of permissions which the user has. For example, if a user does not have required permissions for creating users, they won't be able to unassign a role which will allow to do that. This is done to prevent escalation of privileges.

| Action | Scope |
| --- | --- |
| users.roles:remove | permissions:type:delegate |

## Query parameters

| Param | Type | Required | Description |
| --- | --- | --- | --- |
| global | boolean | No | A flag indicating if the assignment is global or not. If set to `false`, the default org ID of the authenticated user will be used from the request to remove assignment. |

## Example request

http Copy

```
DELETE /api/access-control/users/1/roles/AFUXBHKnk
Accept: application/json
```

## Example response

http Copy

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

{
    "message": "Role removed from user."
}
```

## Status codes

| Code | Description |
| --- | --- |
| 200 | Role is unassigned. |

| Code | Description |
| --- | --- |
| 403 | Access denied. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Set service account role assignments

`PUT /api/access-control/users/:serviceAccountId/roles`

Update the service accounts's role assignments to match the provided set of UIDs. This will remove any assigned roles that aren't in the request and add roles that are in the set but are not already assigned to the service account.

If you want to add or remove a single role, consider using Add a service account role assignment or Remove a service account role assignment instead.

### Required permissions

`permissions:type:delegate` scope ensures that users can only assign or unassign roles which have same, or a subset of permissions which the user has. For example, if a user does not have required permissions for creating users, they won't be able to assign or unassign a role which will allow to do that. This is done to prevent escalation of privileges.

| Action | Scope |
| --- | --- |
| users.roles:add | permissions:type:delegate |
| users.roles:remove | permissions:type:delegate |

### Example request

http

Copy

```
PUT /api/access-control/users/1/roles
Accept: application/json
Content-Type: application/json

{
    "global": false,
    "roleUids": [
        "ZiHQJq5nk",
```

```
        "GzNQ1357k"
    ]
}
```

## JSON body schema

| Field Name | Date Type | Required | Description |
|---|---|---|---|
| global | boolean | No | A flag indicating if the assignment is global or not. If set to `false`, the default org ID of the authenticated user will be used from the request. |
| roleUids | list | Yes | List of role UIDs. |
| includeHidden | boolean | No | Specify whether the hidden role assignments should be updated. |

## Example response

http

Copy

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

{
    "message": "User roles have been updated."
}
```

## Status codes

| Code | Description |
|---|---|
| 200 | Roles have been assigned. |
| 403 | Access denied. |
| 404 | Role not found. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

# Create and remove team role assignments

## List roles assigned to a team

`GET /api/access-control/teams/:teamId/roles`

Lists the roles that have been directly assigned to a given team.

Query Parameters:

- `includeHidden`: Optional. Set to `true` to include roles that are `hidden`.

### Required permissions

| Action | Scope |
| --- | --- |
| teams.roles:read | teams:id:`<team ID>` |

### Example request

```http
GET /api/access-control/teams/1/roles
Accept: application/json
```

### Example response

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

[
    {
        "version": 4,
        "uid": "j08ZBi-nk",
        "name": "fixed:licensing:reader",
        "displayName": "Licensing reader",
        "description": "Read licensing information and licensing reports.",
```

```
        "group": "Licenses",
        "updated": "2022-02-03T14:19:50+01:00",
        "created": "0001-01-01T00:00:00Z",
        "global": false
    }
]
```


### Status codes

| Code | Description |
| --- | --- |
| 200 | Set of assigned roles is returned. |
| 403 | Access denied. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Add a team role assignment

`POST /api/access-control/teams/:teamId/roles`

Assign a role to a specific team.

For bulk updates consider Set team role assignments.

### Required permissions

`permissions:type:delegate` scope ensures that users can only assign roles which have same, or a subset of permissions which the user has. For example, if a user does not have the permissions required to create users, they won't be able to assign a role that contains these permissions. This is done to prevent escalation of privileges.

| Action | Scope |
| --- | --- |
| teams.roles:add | permissions:type:delegate |

### Example request

http Copy

```
POST /api/access-control/teams/1/roles
Accept: application/json
Content-Type: application/json

{
    "roleUid": "XvHQJq57z"
}
```

## JSON body schema

| Field Name | Data Type | Required | Description |
| --- | --- | --- | --- |
| roleUid | string | Yes | UID of the role. |

## Example response

http Copy

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

{
    "message": "Role added to the team."
}
```

## Status codes

| Code | Description |
| --- | --- |
| 200 | Role is assigned to a team. |
| 403 | Access denied. |
| 404 | Role not found. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

# Remove a team role assignment

`DELETE /api/access-control/teams/:teams/roles/:roleUID`

Revoke a role from a team.

For bulk updates consider Set team role assignments.

## Required permissions

`permissions:type:delegate` scope ensures that users can only unassign roles which have same, or a subset of permissions which the user has. For example, if a user does not have the permissions required to create users, they won't be able to assign a role that contains these permissions. This is done to prevent escalation of privileges.```

| Action | Scope |
| --- | --- |
| teams.roles:remove | permissions:type:delegate |

## Example request

http Copy

```
DELETE /api/access-control/teams/1/roles/AFUXBHKnk
Accept: application/json
```

## Example response

http Copy

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

{
    "message": "Role removed from team."
}
```

## Status codes

| Code | Description |
| --- | --- |
| 200 | Role is unassigned. |
| 403 | Access denied. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

## Set team role assignments

`PUT /api/access-control/teams/:teamId/roles`

Update the team's role assignments to match the provided set of UIDs. This will remove any assigned roles that aren't in the request and add roles that are in the set but are not already assigned to the user.

If you want to add or remove a single role, consider using Add a team role assignment or Remove a team role assignment instead.

### Required permissions

`permissions:type:delegate` scope ensures that users can only assign or unassign roles which have same, or a subset of permissions which the user has. For example, if a user does not have required permissions for creating users, they won't be able to assign or unassign a role to a team which will allow to do that. This is done to prevent escalation of privileges.

| Action | Scope |
| --- | --- |
| teams.roles:add | permissions:type:delegate |
| teams.roles:remove | permissions:type:delegate |

### Example request

http

Copy

```
PUT /api/access-control/teams/1/roles
Accept: application/json
Content-Type: application/json
```

```
{
    "roleUids": [
        "ZiHQJq5nk",
        "GzNQ1357k"
    ]
}
```

## JSON body schema

| Field Name | Date Type | Required | Description |
|---|---|---|---|
| roleUids | list | Yes | List of role UIDs. |
| includeHidden | boolean | No | Specify whether the hidden role assignments should be updated. |

## Example response

http Copy

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

{
    "message": "Team roles have been updated."
}
```

## Status codes

| Code | Description |
|---|---|
| 200 | Roles have been assigned. |
| 403 | Access denied. |
| 404 | Role not found. |
| 500 | Unexpected error. Refer to body and/or server logs for more details. |

# Reset basic roles to their default

`POST /api/access-control/roles/hard-reset`

`permissions:type:escalate` scope enables users to reset basic roles permissions. This could result in basic roles having permissions exceedind those of callers.

Reset basic roles permissions to their default.

## Required permissions

| Action | Scope |
| --- | --- |
| roles:write | permissions:type:escalate |

## Example request

http

```
POST /api/access-control/roles/hard-reset
Accept: application/json
Content-Type: application/json

{
    "BasicRoles": true
}
```

## JSON body schema

| Field Name | Data Type | Required | Description |
| --- | --- | --- | --- |
| BasicRoles | boolean | No | Option to reset basic roles permissions. |

## Example response

http

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

{
    "message": "Reset performed"
}
```

### Status codes

| Code | Description |
| --- | --- |
| 200 | Reset performed |
| 500 | Failed to reset basic roles |

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Grafana Role-based access control (RBAC)

Docs

Manage Grafana RBAC roles

Docs

Assign Grafana RBAC roles

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

### Company

- Partnerships
- Our team

### Localized content

- Japanese pages
- German pages

Careers

Success stories

Newsroom

Contact us

Getting help

Merch

Open positions WE'RE HIRING

The Story of Grafana

French pages

Spanish pages

Portuguese pages

---

Grafana Cloud status

Legal and Security

Terms of Service

Privacy Policy

Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Reporting API

This API allows you to interact programmatically with the Reporting feature.

The Reporting API is not stabilized yet, it is still in active development and may change without prior notice.

Reporting is only available in Grafana Enterprise. Read more about Grafana Enterprise.

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information.

## List all reports

`GET /api/reports`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| reports:read | reports:*<br>reports:id:* |

## Example request

http Copy

```
GET /api/reports HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## Example response

http Copy

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 1840

[
    {
        "id": 2,
        "userId": 1,
        "orgId": 1,
        "name": "Report 2",
        "recipients": "example-report@grafana.com",
        "replyTo": "",
        "message": "Hi, \nPlease find attached a PDF status report. If you have any ques
        "schedule": {
            "startDate": "2022-10-02T00:00:00+02:00",
            "endDate": null,
            "frequency": "once",
            "intervalFrequency": "",
            "intervalAmount": 0,
```

Expand code

### Status Codes

- **200** – OK
- **401** - Authentication failed, refer to Authentication API.
- **500** – Unexpected error or server misconfiguration. Refer to server logs for more details.

## Get a report

`GET /api/reports/:id`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| reports:read | reports:*<br>reports:id:*<br>reports:id:1(single report) |

## Example request

http Copy

```
GET /api/reports/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## Example response

http Copy

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 940
```

```
{
    "id": 2,
    "userId": 1,
    "orgId": 1,
    "name": "Report 2",
    "recipients": "example-report@grafana.com",
    "replyTo": "",
    "message": "Hi, \nPlease find attached a PDF status report. If you have any questions
    "schedule": {
        "startDate": "2022-10-02T00:00:00+02:00",
        "endDate": null,
        "frequency": "once",
        "intervalFrequency": "",
        "intervalAmount": 0,
        "workdaysOnly": false
```


Expand code

## Status Codes

- **200** – OK
- **400** – Bad request (invalid report ID).
- **401** - Authentication failed, refer to Authentication API.
- **403** – Forbidden (access denied to a report or a dashboard used in the report).
- **404** – Not found (such report does not exist).
- **500** – Unexpected error or server misconfiguration. Refer to server logs for more details.

# Create a report

`POST /api/reports`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| reports:create | n/a |

---

## Example request

http

Copy

```
POST /api/reports HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
    "name": "Report 4",
    "recipients": "texample-report@grafana.com",
    "replyTo": "",
    "message": "Hello, please, find the report attached",
    "schedule": {
        "startDate": "2022-10-02T10:00:00+02:00",
        "endDate": "2022-11-02T20:00:00+02:00",
        "frequency": "daily",
        "intervalFrequency": "",
        "intervalAmount": 0,
        "workdaysOnly": true,
        "timeZone": "Europe/Warsaw"
    },
```

Expand code

## Config JSON Body Schema

| Field name | Data type | Description |
|---|---|---|
| name | string | Name of the report that is used as an email subject. |
| recipients | string | Comma-separated list of emails to which to send the report to. |
| replyTo | string | Comma-separated list of emails used in a reply-to field of the report email. |
| message | string | Text message used for the body of the report email. |
| startDate | string | Report distribution starts from this date. |
| endDate | string | Report distribution ends on this date. |

| Field name | Data type | Description |
| --- | --- | --- |
| frequency | string | Specifies how often the report should be sent. Can be `once`, `hourly`, `daily`, `weekly`, `monthly`, `last` or `custom`.<br>`last` - schedules the report for the last day of month.<br>`custom` - schedules the report to be sent on a custom interval.<br>It requires `intervalFrequency` and `intervalAmount` to be specified: for example, every 2 weeks, where 2 is an `intervalAmount` and `weeks` is an `intervalFrequency`. |
| intervalFrequency | string | The type of the `custom` interval: `hours`, `days`, `weeks`, `months`. |
| intervalAmount | number | `custom` interval amount. |
| workdaysOnly | string | Send the report only on Monday-Friday. Applicable to `hourly` and `daily` types of schedule. |
| timeZone | string | Time zone used to schedule report execution. |
| orientation | string | Can be `portrait` or `landscape`. |
| layout | string | Can be `grid` or `simple`. |
| enableDashboardUrl | bool | Adds a dashboard url to the bottom of the report email. |
| formats | []string | Specified what kind of attachment to generate for the report - `csv`, `pdf`, `image`.<br>`pdf` is the default one.<br>`csv` attaches a CSV file for each table panel.<br>`image` embeds an image of a dashboard into the email's body. |
| dashboards | []object | Dashboards to generate a report for.<br>See "Report Dashboard Schema" section below. |

## Report Dashboard Schema

| Field name | Data type | Description |
| --- | --- | --- |
| dashboard.uid | string | Dashboard UID. |
| timeRange.from | string | Dashboard time range from. |
| timeRange.to | string | Dashboard time range to. |
| reportVariables. | string | Key-value pairs containing the template variables for this report, in JSON format. If empty, the template variables from |

| Field name | Data type | Description |
| --- | --- | --- |
| | | the report's dashboard will be used. |

### Example response

```http
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 35

{
    "id": 4,
    "message": "Report created"
}
```

### Status Codes

- **200** – OK
- **400** – Bad request (invalid json, missing or invalid fields values, etc.).
- **403** - Forbidden (access denied to a report or a dashboard used in the report).
- **500** - Unexpected error or server misconfiguration. Refer to server logs for more details

## Update a report

`PUT /api/reports/:id`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| reports:write | reports:*<br>reports:id:*<br>reports:1(single report) |

## Example request

See JSON body schema for fields description.

http Copy

```
GET /api/reports HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
    "name": "Updated Report",
    "recipients": "example-report@grafana.com",
    "replyTo": "",
    "message": "Hello, please, find the report attached",
    "schedule": {
        "frequency": "hourly",
        "timeZone": "Africa/Cairo",
        "workdaysOnly": true,
        "startDate": "2022-10-10T10:00:00+02:00",
        "endDate": "2022-11-20T19:00:00+02:00"
    },
    "options": {
        "orientation": "landscape",
```

Expand code

## Example response

http Copy

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 28

{
```

```
    "message": "Report updated"
}
```

### Status Codes

- **200** – OK
- **400** – Bad request (invalid json, missing or invalid fields values, etc.).
- **401** - Authentication failed, refer to Authentication API.
- **403** – Forbidden (access denied to a report or a dashboard used in the report).
- **404** – Not found (such report does not exist).
- **500** – Unexpected error or server misconfiguration. Refer to server logs for more details.

## Delete a report

`DELETE /api/reports/:id`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| reports:delete | reports:*<br>reports:id:*<br>reports:1(single report) |

### Example request

http Copy

```
GET /api/reports/6 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example response

```http
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 39

{
    "message": "Report config was removed"
}
```

### Status Codes

- **200** – OK
- **400** – Bad request (invalid report ID).
- **401** - Authentication failed, refer to Authentication API.
- **404** - Not found (report with this ID does not exist).
- **500** - Unexpected error or server misconfiguration. Refer to server logs for more details

## Send a report

`POST /api/reports/email`

Generate and send a report. This API waits for the report to be generated before returning. We recommend that you set the client's timeout to at least 60 seconds.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| reports:send | n/a |

## Example request

```http
POST /api/reports/email HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "id":"3",
  "useEmailsFromReport": true
}
```

## JSON Body Schema

| Field name | Data type | Description |
| --- | --- | --- |
| id | string | ID of the report to send. It is the same as in the URL when editing a report, not to be confused with the ID of the dashboard. Required. |
| emails | string | Comma-separated list of emails to which to send the report to. Overrides the emails from the report. Required if `useEmailsFromReport` is not present. |
| useEmailsFromReport | boolean | Send the report to the emails specified in the report. Required if `emails` is not present. |

## Example response

```http
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 29

{"message":"Report was sent"}
```

### Status Codes

- **200** – Report was sent.
- **400** – Bad request (invalid json, missing content-type, missing or invalid fields, etc.).
- **401** - Authentication failed, refer to Authentication API.
- **403** - Forbidden (access denied to a report or a dashboard used in the report).
- **404** - Report not found.
- **500** - Unexpected error or server misconfiguration. Refer to server logs for more details.

## Get reports branding settings

`GET /api/reports/settings`

Returns reports branding settings that are global and used across all the reports.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| reports.settings:read | n/a |

## Example request

http Copy

```
GET /api/reports/settings HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

## Example response

http Copy

```
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 181

{
    "id": 1,
    "userId": 1,
    "orgId": 1,
    "branding": {
        "reportLogoUrl": "",
        "emailLogoUrl": "",
        "emailFooterMode": "sent-by",
        "emailFooterText": "Grafana Labs",
        "emailFooterLink": "https://grafana.com/"
    }
}
```

### Status Codes

- **200** – OK
- **401** - Authentication failed, refer to Authentication API.
- **500** - Unexpected error or server misconfiguration. Refer to server logs for more detail

## Save reports branding settings

`POST /api/reports/settings`

Creates settings if they don't exist, otherwise updates them. These settings are global and used across all the reports.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| reports.settings:write | n/a |

## Example request

```http
POST /api/reports/settings HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
    "branding": {
        "reportLogoUrl": "https://grafana.com/reportLogo.jpg",
        "emailLogoUrl": "https://grafana.com/emailLogo.jpg",
        "emailFooterMode": "sent-by",
        "emailFooterText": "Grafana Labs",
        "emailFooterLink": "https://grafana.com/"
    }
}
```

## JSON Body Schema

| Field name | Data type | Description |
|---|---|---|
| branding.reportLogoUrl | string | URL of an image used as a logo on every page of the report. |
| branding.emailLogoUrl | string | URL of an image used as a logo in the email. |
| branding.emailFooterMode | string | Can be `sent-by` or `none`. `sent-by` adds a “Sent by `branding.emailFooterText`” footer link to the email. Requires specifying values in the `branding.emailFooterText` and `branding.emailFooterLink` fields. `none` suppresses adding a “Sent by” footer link to the email. |
| branding.emailFooterText | string | Text of a URL added to the email “Sent by” footer. |
| branding.emailFooterLink | string | URL address value added to the email “Sent by” footer. |

### Example response

```http
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 35

{
    "message": "Report settings saved"
}
```

### Status Codes

- **200** – OK
- **400** – Bad request (invalid json, missing or invalid fields values, etc.).
- **401** - Authentication failed, refer to Authentication API.
- **500** - Unexpected error or server misconfiguration. Refer to server logs for more detail

## Send a test email

`POST /api/reports/test-email`

Sends a test email with a report without persisting it in the database.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| reports:send | n/a |

### Example request

See JSON body schema for fields description.

```http
POST /api/reports/test-email HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{{
    "name": "Report 4",
    "recipients": "example-report@grafana.com",
    "replyTo": "",
    "message": "Hello, please, find the report attached",
    "schedule": {
        "startDate": "2022-10-02T10:00:00+02:00",
        "endDate": "2022-11-02T20:00:00+02:00",
        "frequency": "daily",
        "intervalFrequency": "",
        "intervalAmount": 0,
        "workdaysOnly": true,
        "timeZone": "Europe/Warsaw"
    },
```

Copy

Expand code

## Example response

```http
HTTP/1.1 200 OK
Content-Type: application/json
Content-Length: 29

{
    "message": "Test email sent"
}
```

Copy

## Status Codes

- **200** – OK
- **400** – Bad request (invalid json, missing or invalid fields values, etc.).

---

- **401** - Authentication failed, refer to Authentication API.
- **403** - Forbidden (access denied to a report or a dashboard used in the report).
- **500** - Unexpected error or server misconfiguration. Refer to server logs for more details.

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Grafana Enterprise

Docs

Configure Grafana Enterprise

Docs

Grafana Enterprise license

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

**5 Mar**

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

**60 min**

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

**60 min**

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

## Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

## Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

## Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

## Company

- Partnerships
- Our team
- Careers
- Success stories
- Newsroom
- Contact us
- Getting help
- Merch
- Open positions WE'RE HIRING
- The Story of Grafana

## Localized content

- Japanese pages
- German pages
- French pages
- Spanish pages
- Portuguese pages

Grafana Cloud status

Legal and Security

Terms of Service

Privacy Policy

Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Service account API

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information. For Grafana Cloud instances, please use a Bearer token to authenticate. The examples within this section reference Basic authentication which is for On-Prem Grafana instances.

## Search service accounts with Paging

`GET /api/serviceaccounts/search?perpage=10&page=1&query=myserviceaccount`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:read | n/a |

### Example Request:

```http
```

Copy

```
GET /api/serviceaccounts/search?perpage=10&page=1&query=mygraf HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Default value for the `perpage` parameter is `1000` and for the `page` parameter is `1`. The `totalCount` field in the response can be used for pagination of the user list E.g. if `totalCount` is equal to 100 users and the `perpage` parameter is set to 10 then there are 10 pages of users. The `query` parameter is optional and it will return results where the query value is contained in one of the `name`. Query values with spaces need to be URL encoded e.g. `query=Jane%20Doe`.

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json
{
    "totalCount": 2,
    "serviceAccounts": [
        {
            "id": 1,
            "name": "grafana",
            "login": "sa-grafana",
            "orgId": 1,
            "isDisabled": false,
            "role": "Viewer",
            "tokens": 0,
            "avatarUrl": "/avatar/85ec38023d90823d3e5b43ef35646af9",
            "accessControl": {
                "serviceaccounts:delete": true,
                "serviceaccounts:read": true,
                "serviceaccounts:write": true
            }
```

Expand code

## Create service account

`POST /api/serviceaccounts`

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:create | n/a |

**Example Request:**

```http
POST /api/serviceaccounts HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=

{
  "name": "grafana",
  "role": "Viewer",
  "isDisabled": false
}
```

**Example Response:**

```http
HTTP/1.1 201
Content-Type: application/json

{
    "id": 1,
    "name": "test",
    "login": "sa-test",
    "orgId": 1,
    "isDisabled": false,
    "createdAt": "2022-03-21T14:35:33Z",
    "updatedAt": "2022-03-21T14:35:33Z",
    "avatarUrl": "/avatar/8ea890a677d6a223c591a1beea6ea9d2",
    "role": "Viewer",
```

```
    "teams": []
}
```

## Get a service account by ID

`GET /api/serviceaccounts/:id`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:read | serviceaccounts:id:* |

### Example Request:

http Copy

```
GET /api/serviceaccounts/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

### Example Response:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
    "id": 1,
    "name": "test",
    "login": "sa-test",
    "orgId": 1,
    "isDisabled": false,
    "createdAt": "2022-03-21T14:35:33Z",
    "updatedAt": "2022-03-21T14:35:33Z",
    "avatarUrl": "/avatar/8ea890a677d6a223c591a1beea6ea9d2",
```

```
    "role": "Viewer",
    "teams": []
}
```

## Update service account

`PATCH /api/serviceaccounts/:id`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:write | serviceaccounts:id:* |

### Example Request:

http Copy

```
PATCH /api/serviceaccounts/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=

{
  "name": "test",
    "role": "Editor"
}
```

### Example Response:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
    "id": 2,
    "name": "test",
```

```
    "login": "sa-grafana",
    "orgId": 1,
    "isDisabled": false,
    "createdAt": "2022-03-21T14:35:44Z",
    "updatedAt": "2022-03-21T14:35:44Z",
    "avatarUrl": "/avatar/8ea890a677d6a223c591a1beea6ea9d2",
    "role": "Editor",
    "teams": []
}
```


## Delete service account

`DELETE /api/serviceaccounts/:id`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:delete | serviceaccounts:id:* |

### Example Request:

http Copy

```
DELETE /api/serviceaccounts/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

### Example Response:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
```

```
    "message": "Service account deleted"
}
```

## Migrate API keys to service accounts

`POST /api/serviceaccounts/migrate`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:write | serviceaccounts:* |

**Example Request:**

```http
POST /api/serviceaccounts/migrate HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{
    "message": "API keys migrated to service accounts"
}
```

## Migrate API key to service account

`POST /api/serviceaccounts/migrate/:keyId`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:write | serviceaccounts:* |

### Example Request:

```http
POST /api/serviceaccounts/migrate/4 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json

{
    "message": "Service accounts migrated"
}
```

## Get API key to service account migration status

`GET /api/serviceaccounts/migrationstatus`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:read | serviceaccounts:* |

### Example Request:

```http
POST /api/serviceaccounts/migrationstatus HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json

{
    "migrated": true
}
```

## Hide the API keys tab

`GET /api/serviceaccounts/hideApiKeys`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:write | serviceaccounts:* |

### Example Request:

```http
POST /api/serviceaccounts/hideApiKeys HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json

{
    "message": "API keys hidden"
}
```

## Get service account tokens

`GET /api/serviceaccounts/:id/tokens`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:read | serviceaccounts:id:* |

### Example Request:

```http
GET /api/serviceaccounts/2/tokens HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json

[
```

```
    {
        "id": 1,
        "name": "grafana",
        "role": "Viewer",
        "created": "2022-03-23T10:31:02Z",
        "expiration": null,
        "secondsUntilExpiration": 0,
        "hasExpired": false
    }
]
```

## Create service account tokens

`POST /api/serviceaccounts/:id/tokens`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:write | serviceaccounts:id:* |

### Example Request:

http Copy

```
POST /api/serviceaccounts/2/tokens HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=

{
    "name": "grafana"
}
```

### Example Response:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
    "id": 7,
    "name": "grafana",
    "key": "eyJrIjoiVjFxTHZ6dGdPSjg5Um92MjN1RlhjMkNqYkZUbm9jYkwiLCJuIjoiZ3JhZmFuYSIsImlk
}
```

## Delete service account tokens

`DELETE /api/serviceaccounts/:id/tokens/:tokenId`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:write | serviceaccounts:id:* |

**Example Request:**

http Copy

```
DELETE /api/serviceaccounts/2/tokens/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
```

```
    "message": "API key deleted"
}
```

## Revert service account token to API key

`DELETE /api/serviceaccounts/:serviceAccountId/revert/:keyId`

This operation will delete the service account and create a legacy API Key for the given `keyId`.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| serviceaccounts:delete | serviceaccounts:id:* |

**Example Request:**

http Copy

```
DELETE /api/serviceaccounts/1/revert/glsa_VVQjot0nijQ59lun6pMZRtsdBXxnFQ9M_77c34a79 HTTP/
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
    "message": "Reverted service account to API key"
}
```

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Graphite HTTP API

Docs

Grafana Mimir HTTP API

Docs

API keys

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

**Products**

Grafana Cloud

Grafana Enterprise Stack

**Open Source**

Grafana

Grafana Alloy

**Learn**

Documentation

Grafana Labs blog

Grafana visualizations
Plugins
Grafana Cloud Logs
Grafana Cloud Metrics
Grafana Cloud Traces
Grafana Cloud Profiles
AI/ML tools for observability
Application Observability
Frontend Observability
Grafana Cloud Asserts
Grafana IRM
Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Grafana Beyla
Grafana Faro
Grafana k6
Grafana Loki
Grafana Mimir
Grafana OnCall
Grafana Pyroscope
Grafana Tanka
Grafana Tempo
Graphite
OpenTelemetry
Prometheus
Dashboard templates
Downloads
GitHub

Webinars and videos
Tutorials
Workshops
Events
Community
Community forums
Community Slack
Grafana Champions
Community organizers
Observability Survey
OSS vs Cloud
Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



# Short URL API

Use this API to create shortened URLs. A short URL represents a longer URL containing complex query parameters in a smaller and simpler format.

## Create short URL

`POST /api/short-urls`

Creates a short URL.

**Example request:**

http Copy

```
POST /api/short-urls HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "path": "d/TxKARsmGz/new-dashboard?orgId=1&from=1599389322894&to=1599410922894"
}
```

JSON body schema:

- **path** – The path to shorten, relative to the Grafana root_url.

**Example response:**

```http
HTTP/1.1 200
Content-Type: application/json

{
  "uid": AT76wBvGk,
  "url": http://localhost:3000/goto/AT76wBvGk?orgId=1
}
```

Status codes:

- **200** – Created
- **400** – Errors (invalid JSON, missing or invalid fields)

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Graphite HTTP API

Docs

Grafana Mimir HTTP API

Docs

Use the HTTP API to manage alerting resources

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

### Company

- Partnerships
- Our team
- Careers
- Success stories

### Localized content

- Japanese pages
- German pages
- French pages
- Spanish pages

Newsroom

Contact us

Getting help

Merch

Open positions WE'RE HIRING

The Story of Grafana

Portuguese pages

---

Grafana Cloud status

Legal and Security

Terms of Service

Privacy Policy

Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# SSO Settings API

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information.

**NOTE**
Available in Public Preview in Grafana 10.4 and on Grafana Cloud behind the `ssoSettingsApi` feature toggle.

The API can be used to create, update, delete, get, and list SSO Settings.

## List SSO Settings

`GET /api/v1/sso-settings`

Lists the SSO Settings for all providers.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `settings:read` | `settings:auth.{provider}:*` |

**Example Request:**

```http
GET /api/v1/sso-settings HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json
[
  {
    "id":       "1",
    "provider":  "github",
    "settings": {
      "apiUrl": "https://api.github.com/user",
      "clientId": "my_github_client",
      "clientSecret": "*********",
      "enabled": true,
      "scopes": "user:email,read:org"
      // rest of the settings
    },
    "source":    "system",
  },
  {
    "id":       "2",
    "provider":  "azuread",
```

Expand code

Status Codes:

- **200** – SSO Settings found
- **400** – Bad Request
- **401** – Unauthorized
- **403** – Access Denied

# Get SSO Settings

`GET /api/v1/sso-settings/:provider`

Gets the SSO Settings for a provider.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `settings:read` | `settings:auth.{provider}:*` |

### Example Request:

```http
GET /api/v1/sso-settings/github HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json
ETag: db87f729761898ee
{
  "id":        "1",
  "provider":  "github",
  "settings": {
    "apiUrl": "https://api.github.com/user",
    "clientId": "my_github_client",
    "clientSecret": "*********",
    "enabled": true,
    "scopes": "user:email,read:org"
    // rest of the settings
  },
```

```
  "source":    "system",
}
```

Status Codes:

- **200** – SSO Settings found
- **400** – Bad Request
- **401** – Unauthorized
- **403** – Access Denied
- **404** – SSO Settings not found

## Update SSO Settings

`PUT /api/v1/sso-settings/:provider`

Updates the SSO Settings for a provider.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `settings:write` | `settings:auth.{provider}:*` |

**Example Request**:

http Copy

```
PUT /api/v1/sso-settings/github HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "settings": {
    "apiUrl": "https://api.github.com/user",
    "clientId": "my_github_client",
    "clientSecret": "my_github_secret",
    "enabled": true,
    "scopes": "user:email,read:org"
```

```
  }
}
```

**Example Response:**

```http
HTTP/1.1 204
Content-Type: application/json
```

Status Codes:

- **204** – SSO Settings updated
- **400** – Bad Request
- **401** – Unauthorized
- **403** – Access Denied

## Delete SSO Settings

`DELETE /api/v1/sso-settings/:provider`

Deletes an existing SSO Settings entry for a provider.

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| `settings:write` | `settings:auth.{provider}:*` |

**Example Request:**

```http
DELETE /api/v1/sso-settings/azuread HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 204
Content-Type: application/json
```

Status Codes:

- **204** – SSO Settings deleted
- **400** – Bad Request
- **401** – Unauthorized
- **403** – Access Denied
- **404** – SSO Settings not found

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

SSO Settings UI and Terraform resource for configuring OAuth providers

Docs

Settings updates at runtime

Docs

API and Terraform support for SAML configuration in Grafana

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

## Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

## Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

## Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

## Company

- Partnerships
- Our team
- Careers
- Success stories
- Newsroom

## Localized content

- Japanese pages
- German pages
- French pages
- Spanish pages
- Portuguese pages

Contact us

Getting help

Merch

Open positions WE'RE HIRING

The Story of Grafana

---

Grafana Cloud status

Legal and Security

Terms of Service

Privacy Policy

Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Team API

This API can be used to manage Teams and Team Memberships.

Access to these API endpoints is restricted as follows:

- All authenticated users are able to view details of teams they are a member of.
- Organization Admins are able to manage all teams and team members.
- If you enable `editors_can_admin` configuration flag, then Organization Editors can create teams and manage teams where they are Admin.
  - If you enable `editors_can_admin` configuration flag, Editors can find out whether a team that they are not members of exists by trying to create a team with the same name.

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information.

## Team Search With Paging

`GET /api/teams/search?perpage=50&page=1&query=myteam&sort=memberCount-desc`

or

`GET /api/teams/search?name=myteam`

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams:read | teams:* |

**Example Request:**

```http
GET /api/teams/search?perpage=10&page=1&query=mytestteam HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{
  "totalCount": 1,
  "teams": [
    {
      "id": 1,
      "orgId": 1,
      "name": "MyTestTeam",
      "email": "",
      "avatarUrl": "\/avatar\/3f49c15916554246daa714b9bd0ee398",
      "memberCount": 1
    }
  ],
  "page": 1,
  "perPage": 1000
}
```

## Using the query parameter

Default value for the `perpage` parameter is `1000` and for the `page` parameter is `1`.

The `totalCount` field in the response can be used for pagination of the teams list E.g. if `totalCount` is equal to 100 teams and the `perpage` parameter is set to 10 then there are 10 pages of teams.

The `query` parameter is optional and it will return results where the query value is contained in the `name` field. Query values with spaces need to be URL encoded e.g. `query=my%20team`.

The `sort` param is an optional comma separated list of options to order the search result. Accepted values for the sort filter are: `name-asc`, `name-desc`, `email-asc`, `email-desc`, `memberCount-asc`, `memberCount-desc`. By default, if `sort` is not specified, the teams list will be ordered by `name` in ascending order.

### Using the name parameter

The `name` parameter returns a single team if the parameter matches the `name` field.

#### Status Codes:

- **200** - Ok
- **400** - Bad Request
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found (if searching by name)

## Get Team By Id

`GET /api/teams/:id`

#### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams:read | teams:* |

**Example Request**:

```http
```

```
GET /api/teams/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
  "orgId": 1,
  "name": "MyTestTeam",
  "email": "",
  "created": "2017-12-15T10:40:45+01:00",
  "updated": "2017-12-15T10:40:45+01:00"
}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found

## Add Team

The Team `name` needs to be unique. `name` is required and `email`,`orgId` is optional.

`POST /api/teams`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams:create | N/A |

**Example Request:**

```http
POST /api/teams HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=

{
  "name": "MyTestTeam",
  "email": "email@test.com",
  "orgId": 2
}
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Team created","teamId":2}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **409** - Team name is taken

## Update Team

There are two fields that can be updated for a team: `name` and `email`.

`PUT /api/teams/:id`

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams:write | teams:* |

**Example Request**:

```http
PUT /api/teams/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=

{
  "name": "MyTestTeam",
  "email": "email@test.com"
}
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Team updated"}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found
- **409** - Team name is taken

## Delete Team By Id

`DELETE /api/teams/:id`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams:delete | teams:* |

### Example Request:

```http
DELETE /api/teams/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Team deleted"}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Failed to delete Team. ID not found

## Get Team Members

`GET /api/teams/:teamId/members`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams.permissions:read | teams:* |

**Example Request**:

```http
GET /api/teams/1/members HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

[
  {
    "orgId": 1,
    "teamId": 1,
    "userId": 3,
    "email": "user1@email.com",
    "login": "user1",
    "avatarUrl": "\/avatar\/1b3c32f6386b0185c40d359cdc733a79"
  },
  {
    "orgId": 1,
    "teamId": 1,
    "userId": 2,
    "email": "user2@email.com",
    "login": "user2",
    "avatarUrl": "\/avatar\/cad3c68da76e45d10269e8ef02f8e73e"
```

Expand code

Status Codes:

- **200** - Ok

- **401** - Unauthorized
- **403** - Permission denied

## Add Team Member

`POST /api/teams/:teamId/members`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams.permissions:write | teams:* |

**Example Request:**

http Copy

```
POST /api/teams/1/members HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=

{
  "userId": 2
}
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{"message":"Member added to Team"}
```

Status Codes:

- **200** - Ok
- **400** - User is already added to this team

- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found

## Remove Member From Team

`DELETE /api/teams/:teamId/members/:userId`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams.permissions:write | teams:* |

**Example Request**:

```http
DELETE /api/teams/2/members/3 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Team Member removed"}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found/Team member not found

# Get Team Preferences

`GET /api/teams/:teamId/preferences`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams:read | teams:* |

### Example Request:

```http
GET /api/teams/2/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

### Example Response:

```http
HTTP/1.1 200
Content-Type: application/json

{
  "theme": "",
  "homeDashboardId": 0,
  "timezone": ""
}
```

# Update Team Preferences

`PUT /api/teams/:teamId/preferences`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams:write | teams:* |

**Example Request**:

```http
PUT /api/teams/2/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "theme": "dark",
  "homeDashboardId": 39,
  "timezone": "utc"
}
```

JSON Body Schema:

- **theme** - One of: `light`, `dark`, or an empty string for the default theme
- **homeDashboardId** - The numerical `:id` of a dashboard, default: `0`
- **timezone** - One of: `utc`, `browser`, or an empty string for the default

Omitting a key will cause the current value to be replaced with the system default value.

**Example Response**:

```http
HTTP/1.1 200
Content-Type: text/plain; charset=utf-8

{
  "message":"Preferences updated"
}
```

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Configure Grafana Teams

Docs

Administer Grafana Teams

Docs

Grafana OnCall teams HTTP API

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

**Products**

Grafana Cloud

Grafana Enterprise Stack

**Open Source**

Grafana

Grafana Alloy

**Learn**

Documentation

Grafana Labs blog

Grafana visualizations
Plugins
Grafana Cloud Logs
Grafana Cloud Metrics
Grafana Cloud Traces
Grafana Cloud Profiles
AI/ML tools for observability
Application Observability
Frontend Observability
Grafana Cloud Asserts
Grafana IRM
Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Grafana Beyla
Grafana Faro
Grafana k6
Grafana Loki
Grafana Mimir
Grafana OnCall
Grafana Pyroscope
Grafana Tanka
Grafana Tempo
Graphite
OpenTelemetry
Prometheus
Dashboard templates
Downloads
GitHub

Webinars and videos
Tutorials
Workshops
Events
Community
Community forums
Community Slack
Grafana Champions
Community organizers
Observability Survey
OSS vs Cloud
Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Team Sync API

Team Sync is only available in Grafana Enterprise. Read more about Grafana Enterprise.

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information.

## Get External Groups

`GET /api/teams/:teamId/groups`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams.permissions:read | teams:* |

**Example Request:**

```
http
```

Copy

```
GET /api/teams/1/groups HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "orgId": 1,
    "teamId": 1,
    "groupId": "cn=editors,ou=groups,dc=grafana,dc=org"
  }
]
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied

## Add External Group

`POST /api/teams/:teamId/groups`

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams.permissions:write | teams:* |

**Example Request:**

```http
POST /api/teams/1/groups HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=

{
  "groupId": "cn=editors,ou=groups,dc=grafana,dc=org"
}
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Group added to Team"}
```

Status Codes:

- **200** - Ok
- **400** - Group is already added to this team
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found

## Remove External Group

`DELETE /api/teams/:teamId/groups?groupId=external-group-id`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| teams.permissions:write | teams:* |

**Example Request:**

```http
DELETE /api/teams/1/groups?groupId=cn%3Deditors%2Cou%3Dgroups%2Cdc%3Dgrafana%2Cdc%3Dorg H
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Team Group removed"}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found/Group not found

---

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Configure Grafana Teams

Docs

Team HTTP API

Docs

Grafana Teams

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud

Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Prometheus
Dashboard templates
Downloads
GitHub

Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# User API

The Users HTTP API does not currently work with an API Token. API Tokens are linked to an organization and an organization role. They cannot be given the permission of server users access, only users can be given that permission. To use these API calls you can use Basic Auth and the Grafana user must have the Grafana Admin role.

API Tokens can be used with Organization HTTP API to get users of specific organization.

If you are running Grafana Enterprise, for some endpoints you'll need to have specific permissions. Refer to Role-based access control permissions for more information.

## Search Users

`GET /api/users?perpage=10&page=1&sort=login-asc,email-asc`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| users:read | global.users:* |

**Example Request:**

```
http
```

Copy

```
GET /api/users HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Default value for the `perpage` parameter is `1000` and for the `page` parameter is `1`. Requires basic authentication and that the authenticated user is a Grafana Admin.

The `sort` param is an optional comma separated list of options to order the search result. Accepted values for the sort filter are: `login-asc`, `login-desc`, `email-asc`, `email-desc`, `name-asc`, `name-desc`, `lastSeenAtAge-asc`, `lastSeenAtAge-desc`. By default, if `sort` is not specified, the user list will be ordered by `login`, `email` in ascending order.

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id": 1,
    "name": "Admin",
    "login": "admin",
    "email": "admin@mygraf.com",
    "isAdmin": true,
    "isDisabled": false,
    "lastSeenAt": "2020-04-10T20:29:27+03:00",
    "lastSeenAtAge": "2m",
    "authLabels": ["OAuth"]
  },
  {
    "id": 2,
    "name": "User",
    "login": "user",
```

Expand code

## Search Users with Paging

`GET /api/users/search?perpage=10&page=1&query=mygraf&sort=login-asc,email-asc`

**Required permissions**

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| users:read | global.users:* |

**Example Request**:

```http
GET /api/users/search?perpage=10&page=1&query=mygraf HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Default value for the `perpage` parameter is `1000` and for the `page` parameter is `1`. The `totalCount` field in the response can be used for pagination of the user list E.g. if `totalCount` is equal to 100 users and the `perpage` parameter is set to 10 then there are 10 pages of users. The `query` parameter is optional and it will return results where the query value is contained in one of the `name`, `login` or `email` fields. Query values with spaces need to be URL encoded e.g. `query=Jane%20Doe`.

The `sort` param is an optional comma separated list of options to order the search result. Accepted values for the sort filter are: `login-asc`, `login-desc`, `email-asc`, `email-desc`, `name-asc`, `name-desc`, `lastSeenAtAge-asc`, `lastSeenAtAge-desc`. By default, if `sort` is not specified, the user list will be ordered by `login`, `email` in ascending order.

Requires basic authentication and that the authenticated user is a Grafana Admin.

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json
{
  "totalCount": 2,
  "users": [
    {
      "id": 1,
      "name": "Admin",
      "login": "admin",
```

```
      "email": "admin@mygraf.com",
      "isAdmin": true,
      "isDisabled": false,
      "lastSeenAt": "2020-04-10T20:29:27+03:00",
      "lastSeenAtAge': "2m",
      "authLabels": ["OAuth"]
    },
    {
      "id": 2,
```


Expand code

## Get single user by Id

`GET /api/users/:id`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| users:read | global.users:* |

### Example Request:

http

Copy

```
GET /api/users/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Requires basic authentication and that the authenticated user is a Grafana Admin.

### Example Response:

http

Copy

```
HTTP/1.1 200
Content-Type: application/json

{
```

```
  "id": "1",
  "email": "user@mygraf.com",
  "name": "admin",
  "login": "admin",
  "theme": "light",
  "orgId": 1,
  "isGrafanaAdmin": true,
  "isDisabled": true,
  "isExternal": false,
  "authLabels": [],
  "updatedAt": "2019-09-09T11:31:26+01:00",
  "createdAt": "2019-09-09T11:31:26+01:00",
  "avatarUrl": ""
}
```


## Get single user by Username(login) or Email

`GET /api/users/lookup?loginOrEmail=user@mygraf.com`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| users:read | global.users:* |

**Example Request using the email as option:**

```http
GET /api/users/lookup?loginOrEmail=user@mygraf.com HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Request using the username as option:**

```http
```

```
GET /api/users/lookup?loginOrEmail=admin HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Requires basic authentication and that the authenticated user is a Grafana Admin.

**Example Response:**

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{
  "id": 1,
  "email": "user@mygraf.com",
  "name": "admin",
  "login": "admin",
  "theme": "light",
  "orgId": 1,
  "isGrafanaAdmin": true,
  "isDisabled": false,
  "isExternal": false,
  "authLabels": null,
  "updatedAt": "2019-09-25T14:44:37+01:00",
  "createdAt": "2019-09-25T14:44:37+01:00",
  "avatarUrl":""
}
```

## User Update

`PUT /api/users/:id`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| users:write | global.users:* |

**Example Request**:

```http
PUT /api/users/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=

{
  "email":"user@mygraf.com",
  "name":"User2",
  "login":"user",
  "theme":"light"
}
```

Requires basic authentication and that the authenticated user is a Grafana Admin.

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"User updated"}
```

## Get Organizations for user

`GET /api/users/:id/orgs`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| users:read | global.users:* |

**Example Request**:

```http
GET /api/users/1/orgs HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Requires basic authentication and that the authenticated user is a Grafana Admin.

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

[
  {
    "orgId":1,
    "name":"Main Org.",
    "role":"Admin"
  }
]
```

## Get Teams for user

`GET /api/users/:id/teams`

### Required permissions

See note in the introduction for an explanation.

| Action | Scope |
| --- | --- |
| users:read | global.users:* |

| Action | Scope |
| --- | --- |
| teams:read | teams:* |

**Example Request:**

```http
GET /api/users/1/teams HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Requires basic authentication and that the authenticated user is a Grafana Admin.

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id":1,
    "orgId":1,
    "name":"team1",
    "email":"",
    "avatarUrl":"/avatar/3fcfe295eae3bcb67a49349377428a66",
    "memberCount":1
  }
]
```

# User

# Actual User

`GET /api/user`

**Example Request:**

```http
GET /api/user HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Requires basic authentication.

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{
  "id":1,
  "email":"admin@mygraf.com",
  "name":"Admin",
  "login":"admin",
  "theme":"light",
  "orgId":1,
  "isGrafanaAdmin":true,
  "isDisabled":false
  "isExternal": false,
  "authLabels": [],
  "updatedAt": "2019-09-09T11:31:26+01:00",
  "createdAt": "2019-09-09T11:31:26+01:00",
  "avatarUrl": ""
}
```

## Change Password

`PUT /api/user/password`

Changes the password for the user. Requires basic authentication.

**Example Request**:

```http
```

```
PUT /api/user/password HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=

{
  "oldPassword": "old_password",
  "newPassword": "new_password"
}
```

**Example Response**:

http Copy

```
HTTP/1.1 200
Content-Type: application/json

{"message":"User password changed"}
```

### Change Password with a Script

If you need to change a password with a script, here is an example of changing the Admin password using curl with basic auth:

bash Copy

```
curl -X PUT -H "Content-Type: application/json" -d '{
  "oldPassword": "oldpass",
  "newPassword": "newpass",
  "confirmNew": "newpass"
}' http://admin:oldpass@<your_grafana_host>:3000/api/user/password
```

## Switch user context for a specified user

`POST /api/users/:userId/using/:organizationId`

Switch user context to the given organization. Requires basic authentication and that the authenticated user is a Grafana Admin.

**Example Request**:

```http
POST /api/users/7/using/2 HTTP/1.1
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Active organization changed"}
```

## Switch user context for signed in user

`POST /api/user/using/:organizationId`

Switch user context to the given organization.

**Example Request:**

```http
POST /api/user/using/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Active organization changed"}
```

## Organizations of the actual User

`GET /api/user/orgs`

Return a list of all organizations of the current user. Requires basic authentication.

**Example Request**:

```http
GET /api/user/orgs HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

[
  {
    "orgId":1,
    "name":"Main Org.",
    "role":"Admin"
  }
]
```

## Teams that the actual User is member of

`GET /api/user/teams`

Return a list of all teams that the current user is member of.

**Example Request**:

```http
GET /api/user/teams HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id": 1,
    "orgId": 1,
    "name": "MyTestTeam",
    "email": "",
    "avatarUrl": "\/avatar\/3f49c15916554246daa714b9bd0ee398",
    "memberCount": 1
  }
]
```

## Star a dashboard

`POST /api/user/stars/dashboard/:dashboardId`

Stars the given Dashboard for the actual user.

**Example Request:**

```http
POST /api/user/stars/dashboard/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Dashboard starred!"}
```

## Unstar a dashboard

`DELETE /api/user/stars/dashboard/:dashboardId`

Deletes the starring of the given Dashboard for the actual user.

**Example Request**:

```http
DELETE /api/user/stars/dashboard/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

{"message":"Dashboard unstarred"}
```

## Auth tokens of the actual User

`GET /api/user/auth-tokens`

Return a list of all auth tokens (devices) that the actual user currently have logged in from.

**Example Request**:

```http
GET /api/user/auth-tokens HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

**Example Response**:

```http
HTTP/1.1 200
Content-Type: application/json

[
  {
    "id": 361,
    "isActive": true,
    "clientIp": "127.0.0.1",
    "browser": "Chrome",
    "browserVersion": "72.0",
    "os": "Linux",
    "osVersion": "",
    "device": "Other",
    "createdAt": "2019-03-05T21:22:54+01:00",
    "seenAt": "2019-03-06T19:41:06+01:00"
  },
  {
    "id": 364,
    "isActive": false,
```

Copy

Expand code

## Revoke an auth token of the actual User

`POST /api/user/revoke-auth-token`

Revokes the given auth token (device) for the actual user. User of issued auth token (device) will no longer be logged in and will be required to authenticate again upon next activity.

**Example Request**:

```http
POST /api/user/revoke-auth-token HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
```

Copy

```
  "authTokenId": 364
}
```

**Example Response:**

```http
HTTP/1.1 200
Content-Type: application/json

{
  "message": "User auth token revoked"
}
```

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Graphite HTTP API

Docs

Grafana Mimir HTTP API

Docs

Escalation HTTP API

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

### Company

- Partnerships
- Our team
- Careers
- Success stories

### Localized content

- Japanese pages
- German pages
- French pages
- Spanish pages

Newsroom

Contact us

Getting help

Merch

Open positions WE'RE HIRING

The Story of Grafana

Portuguese pages

---

Grafana Cloud status

Legal and Security

Terms of Service

Privacy Policy

Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



# Contribute to Grafana

This page lists resources for developers who want to contribute to the Grafana software ecosystem or build plugins for Grafana.

## General resources

These resources are useful for all developers.

- Contributing to Grafana: Start here to learn how you can contribute your skills to make Grafana even better.
- Developer guide: A guide to help you get started developing Grafana software, includes instructions for how to configure Grafana for development.
- Contributing to documentation: A guide to help you contribute to Grafana documentation, includes links to beginner-friendly issues.
- Architecture guides: These guides explain Grafana's background architecture.
- Create a pull request: A guide for new contributors about how to create your first Grafana pull request.
- REST APIs allow you to interact programmatically with the Grafana backend.

## Best practices and style

Our style guides outline Grafana style for frontend, backend, documentation, and more, including best practices. Please read through them before you start editing or coding!

- Backend style guide explains how we want to write Go code in the future.

- Documentation style guide applies to all documentation created for Grafana products.
- End to end test framework provides guidance for Grafana e2e tests.
- Frontend style guide provides rules and guidance on developing in React for Grafana.
- Redux framework explains how Grafana handles Redux boilerplate code.
- Styling Grafana expands on styling React components with Emotion.
- Theming Grafana explains how to use themes and ThemeContext in Grafana code.

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Developer resources

Docs

Developers

Docs

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

## Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

## Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

## Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

## Company

- Partnerships
- Our team
- Careers
- Success stories
- Newsroom
- Contact us

## Localized content

- Japanese pages
- German pages
- French pages
- Spanish pages
- Portuguese pages

Getting help

Merch

Open positions WE'RE HIRING

The Story of Grafana

---

Grafana Cloud status

Legal and Security

Terms of Service

Privacy Policy

Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



# Grafana Labs Software Grant and Contributor License Agreement (“Agreement”)

This agreement is based on the Apache Software Foundation Contributor License Agreement. (v r190612)

Thank you for your interest in software projects stewarded by Raintank, Inc. dba Grafana Labs (“Grafana Labs”). In order to clarify the intellectual property license granted with Contributions from any person or entity, Grafana Labs must have a Contributor License Agreement (CLA) on file that has been agreed to by each Contributor, indicating agreement to the license terms below. This license is for your protection as a Contributor as well as the protection of Grafana Labs and its users; it does not change your rights to use your own Contributions for any other purpose. This Agreement allows an individual to contribute to Grafana Labs on that individual’s own behalf, or an entity (the “Corporation”) to submit Contributions to Grafana Labs, to authorize Contributions submitted by its designated employees to Grafana Labs, and to grant copyright and patent licenses thereto.

You accept and agree to the following terms and conditions for Your present and future Contributions submitted to Grafana Labs. Except for the license granted herein to Grafana Labs and recipients of software distributed by Grafana Labs, You reserve all right, title, and interest in and to Your Contributions.

1. Definitions. “You” (or “Your”) shall mean the copyright owner or legal entity authorized by the copyright owner that is making this Agreement with Grafana Labs. For legal entities, the entity making a Contribution and all other entities that control, are controlled by, or are under common control with that entity are considered to be a single Contributor. For the purposes of this definition, “control” means (i) the power, direct or indirect, to cause the direction or management of such entity,

whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity. “Contribution” shall mean any work, as well as any modifications or additions to an existing work, that is intentionally submitted by You to Grafana Labs for inclusion in, or documentation of, any of the products owned or managed by Grafana Labs (the “Work”). For the purposes of this definition, “submitted” means any form of electronic, verbal, or written communication sent to Grafana Labs or its representatives, including but not limited to communication on electronic mailing lists, source code control systems (such as GitHub), and issue tracking systems that are managed by, or on behalf of, Grafana Labs for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by You as “Not a Contribution.”

2. Grant of Copyright License. Subject to the terms and conditions of this Agreement, You hereby grant to Grafana Labs and to recipients of software distributed by Grafana Labs a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare derivative works of, publicly display, publicly perform, sublicense, and distribute Your Contributions and such derivative works.

3. Grant of Patent License. Subject to the terms and conditions of this Agreement, You hereby grant to Grafana Labs and to recipients of software distributed by Grafana Labs a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by You that are necessarily infringed by Your Contribution(s) alone or by combination of Your Contribution(s) with the Work to which such Contribution(s) were submitted. If any entity institutes patent litigation against You or any other entity (including a cross-claim or counterclaim in a lawsuit) alleging that your Contribution, or the Work to which you have contributed, constitutes direct or contributory patent infringement, then any patent licenses granted to that entity under this Agreement for that Contribution or Work shall terminate as of the date such litigation is filed.

4. You represent that You are legally entitled to grant the above license. If You are an individual, and if Your employer(s) has rights to intellectual property that you create that includes Your Contributions, you represent that You have received permission to make Contributions on behalf of that employer, or that Your employer has waived such rights for your Contributions to Grafana Labs. If You are a Corporation, any individual who makes a contribution from an account associated with You will be considered authorized to Contribute on Your behalf.

5. You represent that each of Your Contributions is Your original creation (see section 7 for submissions on behalf of others).

6. You are not expected to provide support for Your Contributions,except to the extent You desire to provide support. You may provide support for free, for a fee, or not at all. Unless required by applicable law or agreed to in writing, You provide Your Contributions on an “AS IS” BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE.

7. Should You wish to submit work that is not Your original creation, You may submit it to Grafana Labs separately from any Contribution, identifying the complete details of its source and of any license or other restriction (including, but not limited to, related patents, trademarks, and license agreements) of which you are personally aware, and conspicuously marking the work as “Submitted on behalf of a third-party: [named here]”.

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Contributor License Agreement (CLA) assistant

Docs

Developers

Docs

Build and review documentation

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

## Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

## Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

## Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

## Company

- Partnerships
- Our team
- Careers
- Success stories
- Newsroom

## Localized content

- Japanese pages
- German pages
- French pages
- Spanish pages
- Portuguese pages

Contact us

Getting help

Merch

Open positions WE'RE HIRING

The Story of Grafana

---

Grafana Cloud status

Legal and Security

Terms of Service

Privacy Policy

Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



# Angular support deprecation

Angular plugin support is deprecated and will be removed in a future release. There are legacy core Grafana visualizations and external plugins that rely on Grafana's Angular plugin support to work. The same is likely true for private plugins that have been developed by Grafana users for use on their own instances over the years. From Grafana v9 and onwards, there is a server configuration option that's global to the entire instance and controls whether Angular plugin support is available or not. In Grafana 11, we will change the default value for the configuration to remove support.

Warning messages are displayed if a dashboard depends on an a panel visualization or data source which requires AngularJS as shown in the following video:

To avoid disruption:

- Ensure that you are running the latest version of plugins by following this guide on updating. Many panels and data sources have migrated from AngularJS.
- If you are using legacy Core Grafana visualizations such as Graph or Table-old, migrate to their replacements using the provided automatic migrations.
- Review the list of current Angular plugins to discover which Core and external plugins are impacted, and whether an update or alternative is required.

## Why are we deprecating Angular support?

AngularJS is an old frontend framework whose active development stopped many years ago. Therefore, it poses a security risk. As Grafana itself had already started migrating to React in v5, this presented the most logical framework for our plugin platform. AngularJS also requires unsafe-eval in the CSP (Content Security Policy) settings, which also reduces the security of running JavaScript in the browser.

## When will Angular plugins stop working?

In Grafana 11, which will be released in preview in April 2024 and generally available in May, we will change the default behavior of the angular_support_enabled configuration parameter to turn off support for AngularJS based plugins. In case you still rely on AngularJS-based plugins developed internally or by the community, you will need to enable this option to continue using them.

New Grafana Cloud users will be unable to request for support to be added to their instance.

## When will we remove Angular support completely?

Our current plan is to completely remove any remaining support for Angular plugins in version 12. Including the removal of the angular_support_enabled configuration parameter.

## A dashboard I use is displaying a warning, what do I need to do?

A dashboard displays warnings when one or more panel visualizations or data sources in the dashboard have a dependency on Angular. Contact your system administrator to advise them of the issue or follow the preceding guidance on avoiding disruption.

## How do I migrate an Angular plugin to React?

Depending on if it's a data source plugin, panel plugin, or app plugin the process will differ.

For panels, the rendering logic could in some cases be easily preserved, but all options need to be redone to use the declarative options framework. For data source plugins the query editor and config options will likely need a total rewrite.

## How do I encourage a community plugin to migrate?

We encourage you to locate the repository of the corresponding plugin and create or upvote an Issue within it if you are using a plugin that is still based on Angular.

### Links

- Migrate Angular to React
- Build a panel plugin
- Build a data source plugin
- List of current Angular plugins

---

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

---

## Related documentation

Docs

Developer resources

Docs

Developers

Docs

Provision Grafana Cloud with infrastructure as code

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM
- Grafana k6
- Grafana SLO
- Synthetic Monitoring
- Solutions
- Pricing

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry
- Prometheus
- Dashboard templates
- Downloads
- GitHub

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud
- Writers' Toolkit
- Plugin development
- Load testing
- OpenTelemetry Report
- Log monitoring

## Company

- Partnerships
- Our team
- Careers
- Success stories
- Newsroom
- Contact us
- Getting help
- Merch
- Open positions WE'RE HIRING
- The Story of Grafana

## Localized content

- Japanese pages
- German pages
- French pages
- Spanish pages
- Portuguese pages

---

- Grafana Cloud status
- Legal and Security
- Terms of Service
- Privacy Policy
- Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



Enterprise Open source

# Plugins using AngularJS

The use of AngularJS in Grafana has been deprecated in favor of React. Support for AngularJS will be turned off by default in Grafana 11.

This page explains how Grafana users might be impacted by the removal of Angular support based on plugins dependent on this legacy framework. You will also see if there is a migration option available for a given plugin.

**NOTE**
We are greatly appreciative of the developers who have contributed plugins to the Grafana ecosystem. Guidance on migrating a plugin to React can be found in our migration guide.

## What should I do with the list of AngularJS plugins?

Refer to the table below and take the appropriate action for you.

- Consider the advice on whether to update, migrate to a listed alternative, or explore the Grafana plugins catalog to find the most suitable option for your use case.
- Use our detect-angular-dashboards open source tooling to list dashboards which have a dependency on Angular plugins.
- Check your Grafana instances for usage of these plugins. Refer to the documentation on browsing installed plugins.
- Customers of Grafana Enterprise and users of Grafana Cloud can also leverage usage insights to prioritize any migration efforts.

- Review the plugin source repositories to add your support to any migration issues or consider forking the repo.

> **NOTE**
> If you want to add any specific migration guidance for your plugin here or update our assessment, please open a PR by clicking **Suggest an edit** at the bottom of this page.

## Private plugins

Grafana OSS and Grafana Enterprise support the creation of private plugins for use on local instances. These plugins may have a dependency on AngularJS and require an update.

The `detect-angular-dashboards` tool listed above will include private plugins in its report **if the Grafana version is v10.1.0 or later**.

Additionally, warning icons and messages will be displayed when browsing the catalog via **Administration** > **Plugins and Data** > **Plugins** in your local instance.

## Automatic migration of plugins

Certain legacy Grafana panel plugins automatically update to their React-based replacements when Angular support is disabled. This migration is usually available within the panel options, as shown in the screenshot below for World Map. Automatic migration can be triggered by setting the feature toggle `autoMigrateOldPanels` to `true`.

Automatic migration is supported for the plugins shown in the following table. Each of the target plugins are included in Grafana as Core plugins which don't require installation.

| Plugin | Migration target |
| --- | --- |
| Graph (old) | Time Series |
| Singlestat | Stat |
| Stat (old) | Stat |
| Table (old) | Table |
| Worldmap | Geomap |

A dashboard must still be saved with the new plugin ID to persist the change.

# AngularJS-based plugins

This table lists plugins which we have detected as having a dependency on AngularJS. For alternatives, consider included Visualizations and Data sources, as well as external plugins from the catalog.

| Plugin ID | Name | Action |
|---|---|---|
| grafana-worldmap-panel | Worldmap Panel | Migrate - Geomap (core) replaced Worldmap - Note this should migrate when Angular is disabled. |
| natel-discrete-panel | Discrete | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| vonage-status-panel | Status Panel | Migrate - Consider Stat (core) or Polystat as potential alternatives. |
| grafana-simple-json-datasource | SimpleJson | Migrate - Consider Infinity as a potential alternative. |
| natel-plotly-panel | Plotly | Migrate - Consider alternative nline-plotlyjs-panel plugin. |
| agenty-flowcharting-panel | FlowCharting | Migrate - Consider Canvas (core) or Diagram as potential alternatives. |
| camptocamp-prometheus-alertmanager-datasource | Prometheus AlertManager | Update - Note the minimum version for React is 2.0.0. |
| briangann-gauge-panel | D3 Gauge | Update - Note the minimum version for React is 2.0.0. We recommend the latest. |
| yesoreyeram-boomtable-panel | Boom Table | Migrate - Consider Table (core) and transformations as appropriate. |
| briangann-datatable-panel | Datatable Panel | Wait - New version with React migration is planned. |
| flant-statusmap-panel | Statusmap | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| michaeldmoore-multistat-panel | Multistat | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |

| Plugin ID | Name | Action |
| --- | --- | --- |
| pr0ps-trackmap-panel | TrackMap | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| snuids-trafficlights-panel | Traffic Lights | Migrate - Consider Traffic Light as a potential alternative. |
| vertamedia-clickhouse-datasource | Altinity plugin for ClickHouse | Update - Note the minimum version for React is 3.0.0. |
| petrslavotinek-carpetplot-panel | Carpet plot | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| ryantxu-ajax-panel | AJAX | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| michaeldmoore-annunciator-panel | Annunciator | Migrate - Consider Stat (core). |
| marcuscalidus-svg-panel | SVG | Migrate - Consider alternatives such as Canvas (core), Colored SVG, or others. |
| neocat-cal-heatmap-panel | Cal-HeatMap | Migrate - Consider Heatmap (core) visualization. |
| blackmirror1-singlestat-math-panel | Singlestat Math | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| scadavis-synoptic-panel | SCADAvis Synoptic Panel | Update - Note the minimum version for React is 2.0. |
| farski-blendstat-panel | Blendstat | Migrate - Consider Stat (core) and transformations as appropriate. |
| savantly-heatmap-panel | Heatmap | Migrate - Consider Heatmap (core) visualization. |
| mtanda-histogram-panel | Histogram | Migrate - Consider included Histogram visualization. |
| snuids-radar-panel | Radar Graph | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| fatcloud-windrose-panel | WindRose | Migrate - Consider alternative Operator Windrose plugin. |
| bessler-pictureit-panel | PictureIt | Migrate - Consider alternative ePict plugin. |

| Plugin ID | Name | Action |
| --- | --- | --- |
| digrich-bubblechart-panel | Bubble Chart | Update - Note the minimum version for React is 2.0.1. We recommend the latest. |
| corpglory-progresslist-panel | Progress List | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| aidanmountford-html-panel | HTML | Migrate - Consider alternatives such as Text (core), HTML, or others. |
| fifemon-graphql-datasource | GraphQL Data Source | Wait - Removal of AngularJS is planned. Consider Infinity plugin as alternative. |
| goshposh-metaqueries-datasource | MetaQueries | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| mxswat-separator-panel | Separator | Migrate - Consider alternative Text panel (core) which can be empty and used as a separator. |
| natel-influx-admin-panel | Influx Admin | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| doitintl-bigquery-datasource | Google BigQuery | Migrate - Consider Grafana Big Query plugin. |
| satellogic-3d-globe-panel | 3D Globe Panel | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| foursquare-clouderamanager-datasource | Cloudera Manager | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| grafana-splunk-datasource | Splunk | Update - Note the minimum version for React is 4.1.0. We recommend the latest. |
| grafana-singlestat-panel | Singlestat | Migrate - Consider Stat panel (core). |
| blackmirror1-statusbygroup-panel | Status By Group Panel | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| novalabs-annotations-panel | Annotation Panel | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |

| Plugin ID | Name | Action |
| --- | --- | --- |
| jasonlashua-prtg-datasource | PRTG | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| ryantxu-annolist-panel | Annotation List | Migrate - Consider annotations list (core). |
| cloudflare-app | Cloudflare Grafana App | Migrate - Consider using the Cloudflare Dashboard or DNS Analytics API. |
| smartmakers-trafficlight-panel | TrafficLight | Migrate - Consider Traffic Light as a potential alternative. |
| zuburqan-parity-report-panel | Parity Report | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| citilogics-geoloop-panel | GeoLoop | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| xginn8-pagerduty-datasource | Pagerduty | Wait - We are developing an Enterprise plugin for Pagerduty targeted for availability in Q1 2024. Note that all roadmap items are subject to change. |
| gretamosa-topology-panel | Topology Panel | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| fzakaria-simple-annotations-datasource | Simple Annotations | Migrate - Check for annotations support within your data sources to remove dependency on this plugin. |
| oci-metrics-datasource | Oracle Cloud Infrastructure Metrics | Update - Note the minimum version for React is 5.0.0. |
| abhisant-druid-datasource | Druid | Migrate - Migrate to replacement Druid plugin. |
| devopsprodigy-kubegraf-app | DevOpsProdigy KubeGraf | Migrate - Consider Grafana Kubernetes Monitoring (Grafana Cloud only). |
| mtanda-heatmap-epoch-panel | HeatmapEpoch | Migrate - Consider Heatmap (core) visualization. |
| alexandra-trackmap-panel | Track Map | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |

| Plugin ID | Name | Action |
| --- | --- | --- |
| gnocchixyz-gnocchi-datasource | Gnocchi | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| tencentcloud-monitor-app | Tencent Cloud Monitor | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| andig-darksky-datasource | DarkSky | Remove - Note that support for the DarkSky API was ended by Apple in March 2023. |
| mtanda-google-calendar-datasource | GoogleCalendar | Wait - Migration to React is planned. |
| ntop-ntopng-datasource | ntopng | Migrate - Consider InfluxDB (core), with additional guidance available here. |
| ayoungprogrammer-finance-datasource | Finance | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| grafana-kairosdb-datasource | KairosDB | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| fastweb-openfalcon-datasource | Open-Falcon | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| praj-ams-datasource | Ambari Metrics | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| monasca-datasource | Monasca | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| grafana-strava-datasource | Strava | Update - Note the minimum version for React is 1.6.0. We recommend the latest. |
| gridprotectionalliance-osisoftpi-datasource | OSIsoft-PI | Update - Note the minimum version for React is 4.0.0. We recommend the latest. |
| monitoringartist-monitoringart-datasource | Monitoring Art | Migrate - Browse included visualizations and plugins catalog for potential alternatives. |
| hawkular-datasource | Hawkular | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| ovh-warp10-datasource | Warp 10 | Migrate - Browse included data sources and plugins catalog for |

| Plugin ID | Name | Action |
| --- | --- | --- |
| | | potential alternatives. |
| natel-usgs-datasource | USGS Water Services | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| radensolutions-netxms-datasource | NetXMS | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| ibm-apm-datasource | IBM APM | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| cognitedata-datasource | Cognite Data Fusion | Update - Note the minimum version for React is 4.0.0. We recommend the latest. |
| linksmart-sensorthings-datasource | LinkSmart SensorThings | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| pue-solr-datasource | Solr | Migrate - Consider this guidance on using solr-exporter for prometheus. |
| paytm-kapacitor-datasource | KapacitorSimpleJson | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| oci-logs-datasource | Oracle Cloud Infrastructure Logs | Update - Note the minimum version for React is 4.0.0. |
| gridprotectionalliance-openhistorian-datasource | openHistorian | Wait - Note that new version with React migration is planned. |
| devicehive-devicehive-datasource | DeviceHive | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| rackerlabs-blueflood-datasource | Blueflood | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| udoprog-heroic-datasource | Heroic | Migrate - Note that Heroic DB has been discontinued. |
| akumuli-datasource | Akumuli | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| bmchelix-ade-datasource | BMC Helix | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| sidewinder-datasource | Sidewinder | Migrate - Browse included data sources and plugins catalog for |

| Plugin ID | Name | Action |
| --- | --- | --- |
| | | potential alternatives. |
| linksmart-hds-datasource | LinkSmart HDS Datasource | Migrate - browse included data sources and plugins catalog for potential alternatives. |
| skydive-datasource | Skydive | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| quasardb-datasource | QuasarDB | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| spotify-heroic-datasource | Heroic | Migrate - Note that Heroic DB has been discontinued. |
| grafana-es-open-distro-datasource | Open Distro for Elasticsearch | Migrate - Note that OpenSearch replaced Open Distro for Elasticseach. |
| humio-datasource | Humio | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| yeya24-chaosmesh-datasource | Chaos Mesh | Migrate - Note that plugin was replaced by chaosmeshorg-datasource. |
| kentik-connect-app | Kentik Connect Pro | Update - Note the minimum version for React is 1.7.0. |
| chaosmeshorg-datasource | Chaos Mesh | Update - Note the minimum version for React is 3.0.0. |
| aquaqanalytics-kdbadaptor-datasource | kdb+ | Migrate - Note that kdb+ backend replaced kdb.+. |
| alexanderzobnin-zabbix-app | Zabbix | Update - Note the minimum version for React is 4.3.0. We recommend the latest. Recently brought under Grafana signature. |
| bosun-app | Bosun | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| belugacdn-app | BelugaCDN | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| grafana-azure-data-explorer-datasource | Azure Data Explorer Datasource | Update - The minimum supported version is 3.5.1. We recommend the latest. |
| ddurieux-glpi-app | glpi | Migrate - Browse included data sources and plugins catalog for |

| Plugin ID | Name | Action |
| --- | --- | --- |
| | | potential alternatives. |
| fetzerch-sunandmoon-datasource | Sun and Moon | Update - Note the minimum version for React is 0.3.0. |
| grafana-clock-panel | Clock | Update - Note the minimum version for React is 1.1.0. We recommend the latest. |
| grafana-github-datasource | GitHub | Update - Note the minimum version for React is 1.3.3. We recommend the latest. |
| grafana-datadog-datasource | Datadog | Update - Note the minimum version for React is 3.0.0. We recommend the latest. |
| grafana-gitlab-datasource | Gitlab | Update - Note the minimum version for React is 1.1.0. We recommend the latest. |
| grafana-iot-twinmaker-app | AWS IoT TwinMaker App | Update - Note the minimum version for React is 1.6.3. We recommend the latest. |
| grafana-newrelic-datasource | New Relic | Update - Note the minimum version for React is 3.0.0. We recommend the latest. |
| grafana-opensearch-datasource | Opensearch | Update - Note the minimum version for React is 2.0.0. We recommend the latest. |
| grafana-oracle-datasource | Oracle | Update - Note the minimum version for React is 2.0.6. We recommend the latest. |
| grafana-piechart-panel | Pie Chart | Migrate - Note that Pie Chart (core) replaced Pie Chart. |
| grafana-polystat-panel | Polystat | Update - Note the minimum version for React is 2.0.0. We recommend the latest. |
| grafana-servicenow-datasource | ServiceNow | Update - Note the minimum version for React is 2.0.2. We recommend the latest. |
| grafana-synthetic-monitoring-app | Synthetic Monitoring | Update - Note the minimum version for React is 0.7.3. We recommend the latest. |
| grafana-wavefront-datasource | Wavefront | Update - Note the minimum version for React is 2.0.0. We recommend the latest. |

| Plugin ID | Name | Action |
| --- | --- | --- |
| hadesarchitect-cassandra-datasource | Apache Cassandra | Update - Note the minimum version for React is 2.1.1. We recommend the latest. |
| instana-datasource | Instana | Update - Note the minimum version for React is 3.0.0. We recommend the latest. |
| jdbranham-diagram-panel | Diagram | Update - Note the minimum version for React is 1.7.1. We recommend the latest. |
| larona-epict-panel | ePict | Update - Note the minimum version for React is 2.0.0. We recommend the latest. |
| moogsoft-aiops-app | Moogsoft AIOps | Update - Note the minimum version for React is 9.0.0. |
| opennms-helm-app | OpenNMS Helm | Migrate - Note that OpenNMS Plugin for Grafana replaced OpenNMS Helm. |
| percona-percona-app | Percona | Migrate - Consider use of Percona dashboards. |
| novatec-sdg-panel | Service Dependency Graph | Update - Note the minimum version for React is 4.0.3. |
| pierosavi-imageit-panel | ImageIt | Migrate - Consider ePict or browse plugins catalog for potential alternatives. |
| redis-app | Redis Application | Update - Note the minimum version for React is 1.2.0. We recommend the latest. |
| sbueringer-consul-datasource | Consul | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| simpod-json-datasource | JSON | Update - Note the minimum version for React is 0.3.0. We recommend the latest. |
| singlestat | Singlestat | Migrate - Note that Stat (core) replaced Singlestat. |
| sni-pnp-datasource | PNP | Update - Note the minimum version for React is 2.0.0. We recommend the latest. |
| sni-thruk-datasource | Thruk | Update - Note the minimum version for React is 2.0.0. We recommend the latest. |

| Plugin ID | Name | Action |
| --- | --- | --- |
| stagemonitor-elasticsearch-app | stagemonitor Elasticsearch | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| tdengine-datasource | TDengine Datasource | Update - Note the minimum version for React is 3.3.0. We recommend the latest. |
| vertica-grafana-datasource | Vertica | Update - Note the minimum version for React is 2.0.0. We recommend the latest. |
| voxter-app | Voxter VoIP Platform Metrics | Migrate - Browse included data sources and plugins catalog for potential alternatives. |
| graph | Graph (old) | Migrate - Note that this is replaced by Time Series (core) - This plugin should migrate when Angular is disabled. Also consider Bar Chart or Histogram if appropriate. |
| table-old | Table (old) | Migrate - Note that this is replaced by Table (core) - This plugin should migrate when AngularJS is disabled. |
| shorelinesoftware-shoreline-datasource | Shoreline Data Source | Update - Note the minimum version for React is 1.2.1. We recommend the latest. |

## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

Angular support deprecation

Docs

AngularJS plugin warnings in dashboards

Docs

Migrate from Grafana OSS/Enterprise to Grafana Cloud using the Grafana Cloud Migration Assistant

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud

Grafana k6
Grafana SLO
Synthetic Monitoring
Solutions
Pricing

Prometheus
Dashboard templates
Downloads
GitHub

Writers' Toolkit
Plugin development
Load testing
OpenTelemetry Report
Log monitoring

## Company

Partnerships
Our team
Careers
Success stories
Newsroom
Contact us
Getting help
Merch
Open positions WE'RE HIRING
The Story of Grafana

## Localized content

Japanese pages
German pages
French pages
Spanish pages
Portuguese pages

---

Grafana Cloud status
Legal and Security
Terms of Service
Privacy Policy
Trademark Policy

Copyright 2025 © Grafana Labs

Important: This documentation is about an older version. It's relevant only to the release noted, many of the features and functions have been updated or replaced. Please view the current version.



# Grafana CLI

Grafana CLI is a small executable that is bundled with Grafana server. It can be executed on the same machine Grafana server is running on. Grafana CLI has `plugins` and `admin` commands, as well as global options.

To list all commands and options:

```
grafana cli -h
```

Copy

## Invoking Grafana CLI

To invoke Grafana CLI, add the path to the grafana binaries in your `PATH` environment variable. Alternately, if your current directory is the `bin` directory, use `./grafana cli`. Otherwise, you can specify full path to the CLI. For example, on Linux `/usr/share/grafana/bin/grafana` and on Windows `C:\Program Files\GrafanaLabs\grafana\bin\grafana.exe`, and invoke it with `grafana cli`.

**NOTE**

Some commands, such as installing or removing plugins, require `sudo` on Linux. If you are on Windows, run Windows PowerShell as Administrator.

## Grafana CLI command syntax

The general syntax for commands in Grafana CLI is:

```bash
grafana cli [global options] command [command options] [arguments...]
```

## Global options

Grafana CLI allows you to temporarily override certain Grafana default settings. Except for `--help` and `--version`, most global options are only used by developers.

Each global option applies only to the command in which it is used. For example, `--pluginsDir value` does not permanently change where Grafana saves plugins. It only changes it for command in which you apply the option.

### Display Grafana CLI help

`--help` or `-h` displays the help, including default paths and Docker configuration information.

**Example:**

```bash
grafana cli -h
```

### Display Grafana CLI version

`--version` or `-v` prints the version of Grafana CLI currently running.

**Example:**

```bash
grafana cli -v
```

### Override default plugin directory

`--pluginsDir value` overrides the path to where your local Grafana instance stores plugins. Use this option if you want to install, update, or remove a plugin somewhere other than the default directory ("/var/lib/grafana/plugins") [$GF_PLUGIN_DIR].

**Example:**

```bash
grafana cli --pluginsDir "/var/lib/grafana/devplugins" plugins install <plugin-id>
```

## Override default plugin repo URL

`--repo value` allows you to download and install or update plugins from a repository other than the default Grafana repo.

**Example:**

```bash
grafana cli --repo "https://example.com/plugins" plugins install <plugin-id>
```

## Override default plugin .zip URL

`--pluginUrl value` allows you to download a .zip file containing a plugin from a local URL instead of downloading it from the default Grafana source.

**Example:**

```bash
grafana cli --pluginUrl https://company.com/grafana/plugins/<plugin-id>-<plugin-version>
```

## Override Transport Layer Security

**Warning:** Turning off TLS is a significant security risk. We do not recommend using this option.

`--insecure` allows you to turn off Transport Layer Security (TLS) verification (insecure). You might want to do this if you are downloading a plugin from a non-default source.

**Example:**

```bash
grafana cli --insecure --pluginUrl https://company.com/grafana/plugins/<plugin-id>-<plug
```

## Enable debug logging

`--debug` or `-d` enables debug logging. Debug output is returned and shown in the terminal.

**Example:**

```bash
grafana cli --debug plugins install <plugin-id>
```

## Override a configuration setting

`--configOverrides` is a command line argument that acts like an environmental variable override.

For example, you can use it to redirect logging to another file (maybe to log plugin installations in Grafana Cloud) or when resetting the admin password and you have non-default values for some important configuration value (like where the database is located).

**Example:**

```bash
grafana cli --configOverrides cfg:default.paths.log=/dev/null plugins install <plugin-id>
```

## Override homepath value

Sets the path for the Grafana install/home path, defaults to working directory. You do not need to use this if you are in the Grafana installation directory when using the CLI.

**Example:**

```bash
grafana cli --homepath "/usr/share/grafana" admin reset-admin-password <new password>
```

## Override config file

`--config value` overrides the default location where Grafana expects the configuration file. Refer to Configuration for more information about configuring Grafana and default configuration file locations.

Example:

```bash
grafana cli --config "/etc/configuration/" admin reset-admin-password mynewpassword
```

## Plugins commands

Grafana CLI allows you to install, upgrade, and manage your Grafana plugins. For more information about installing plugins, refer to plugins page.

All listed commands apply to the Grafana default repositories and directories. You can override the defaults with Global Options.

### List available plugins

```bash
grafana cli plugins list-remote
```

### Install the latest version of a plugin

```bash
grafana cli plugins install <plugin-id>
```

### Install a specific version of a plugin

```bash
grafana cli plugins install <plugin-id> <version>
```

### List installed plugins

```bash
grafana cli plugins ls
```

### Update all installed plugins

```bash
grafana cli plugins update-all
```

### Update one plugin

```bash
grafana cli plugins update <plugin-id>
```

### Remove one plugin

```bash
grafana cli plugins remove <plugin-id>
```

## Admin commands

Admin commands are only available in Grafana 4.1 and later.

### Show all admin commands

```bash
grafana cli admin
```

### Reset admin password

`grafana cli admin reset-admin-password <new password>` resets the password for the admin user using the CLI. You might need to do this if you lose the admin password.

If there are two flags being used to set the homepath and the config file path, then running the command returns this error:

> Could not find config defaults, make sure homepath command line parameter is set or working directory is homepath

To correct this, use the `--homepath` global option to specify the Grafana default homepath for this command:

```bash
grafana cli --homepath "/usr/share/grafana" admin reset-admin-password <new password>
```

If you have not lost the admin password, we recommend that you change the user password either in the User Preferences or in the Server Admin > User tab.

If you need to set the password in a script, then you can use the Grafana User API.

## Migrate data and encrypt passwords

`data-migration` runs a script that migrates or cleans up data in your database.

`encrypt-datasource-passwords` migrates passwords from unsecured fields to secure_json_data field. Returns `ok` unless there is an error. Safe to execute multiple times.

**Example:**

```bash
grafana cli admin data-migration encrypt-datasource-passwords
```


## Was this page helpful?

👍 Yes 👎 No

Create a GitHub issue

Email docs@grafana.com

Help and support

Community

## Related documentation

Docs

The Grafana Alloy command-line interface

Docs

The run command

Docs

The convert command

## Related resources from Grafana Labs

Additional helpful documentation, links, and articles:

5 Mar

Getting started with managing your metrics, logs, and traces using Grafana

In this webinar, we'll demo how to get started using the LGTM Stack: Loki for logs, Grafana for visualization, Tempo for traces, and Mimir for metrics.

60 min

Getting started with Grafana dashboard design

In this webinar, you'll learn how to design stylish and easily accessible Grafana dashboards that tell a story.

60 min

Building advanced Grafana dashboards

In this webinar, we'll demo how to build and format Grafana dashboards.

## Sign up for Grafana stack updates

Email

Subscribe

Note: By signing up, you agree to be emailed related product-level information.

Grafana Labs

### Products

- Grafana Cloud
- Grafana Enterprise Stack
- Grafana visualizations
- Plugins
- Grafana Cloud Logs
- Grafana Cloud Metrics
- Grafana Cloud Traces
- Grafana Cloud Profiles
- AI/ML tools for observability
- Application Observability
- Frontend Observability
- Grafana Cloud Asserts
- Grafana IRM

### Open Source

- Grafana
- Grafana Alloy
- Grafana Beyla
- Grafana Faro
- Grafana k6
- Grafana Loki
- Grafana Mimir
- Grafana OnCall
- Grafana Pyroscope
- Grafana Tanka
- Grafana Tempo
- Graphite
- OpenTelemetry

### Learn

- Documentation
- Grafana Labs blog
- Webinars and videos
- Tutorials
- Workshops
- Events
- Community
- Community forums
- Community Slack
- Grafana Champions
- Community organizers
- Observability Survey
- OSS vs Cloud