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

Folder API

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.

A note about the General folder

The General folder (id=0) is special and is not part of the Folder API which means that you cannot use this API for retrieving information about the General folder.

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.

Example Request:

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

Example Response:

```
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.

Example Request:

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

Example Response:

```
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

Create folder

`POST /api/folders`

Creates a new folder.

Example Request:

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

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

JSON Body schema:

- **uid** – Optional unique identifier.
- **title** – The title of the folder.

Example Response:

```
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** – 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.

Example Request:

```
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:

- **uid** – Provide another unique identifier than stored to change the unique identifier.
- **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/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
}
```

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/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 8 Alerts]({{< relref "../alerting/unified-alerting/_index.md" >}}) are 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 8 Alerts. However, if this parameter is set to `true` then it will delete any Grafana 8 Alerts under this folder.

Example Request:

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

Example Response:

```
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.

Example Request:

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

Example Response:

```
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