Folder APIs

- Requirements
- Endpoints
- Get all folders
- Get folder by uid
- Create folder
- Update folder
- Delete folder

{{< admonition type="note" >}} Available in Grafana 12 and later.

This API complies with the new Grafana API structure. To learn more refer to documentation about the API structure in Grafana.

This document may not contain the latest version of the API. For the most up-to-date list of available endpoints, refer to folder.grafana.app/v1 in Swagger.

{{< /admonition >}}

Requirements

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.

Endpoints

Method	URI	Summary
GET	/apis/folder.grafana.app/v1/namespaces/:namespace/folders	Get all folders

Method	URI	Summary
GET	/apis/folder.grafana.app/v1/namespaces/:namespace/folders/:uid	Get folder by uid
POST	/apis/folder.grafana.app/v1/namespaces/:namespace/folders	Create folder
PUT	/apis/folder.grafana.app/v1/namespaces/:namespace/folders/:uid	Update folder
DELETE	/apis/folder.grafana.app/v1/namespaces/:namespace/folders/:uid	Delete folder

Get all folders

`GET /apis/folder.grafana.app/v1/namespaces/:namespace/folders`

Returns all folders that the authenticated user has permission to view within the given organization. Use the `limit` query parameter to control the maximum number of dashboards returned. To retrieve additional dashboards, utilize the `continue` token provided in the response to fetch the next page.

- namespace: to read more about the namespace to use, see the API overview.

Query parameters:

- **`limit`** (optional): Maximum number of folders to return
- **`continue`** (optional): Continue token from a previous response to fetch the next page

Required permissions

See note in the [introduction]({{< ref "#folder-api" >}}) for an explanation.

Action	Scope
`folders:read`	`folders:*`

Example Request:

```
GET /apis/folder.grafana.app/v1/namespaces/default/folders?limit=1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>
```

Example Response:

```
HTTP/1.1 200
Content-Type: application/json
{
  "kind": "FolderList",
  "apiVersion": "folder.grafana.app/v1",
  "metadata": {
    "continue": "eyJvIjoxNTIsInYiOjE3NjE3MDQyMjQyMDcxODksInMiOmZhbHNlfQ=="
  },
  "items": [
    {
      "kind": "Folder",
      "apiVersion": "folder.grafana.app/v1",
      "metadata": {
        "name": "aef30vrzxs3y8d",
        "namespace": "default",
        "uid": "KCtv1FXDsJmTYQoTgcPnfuwZhDZge3uMpXOefaOHjb4X",
        "resourceVersion": "1741343686000",
        "creationTimestamp": "2025-03-07T10:34:46Z",
        "annotations": {
          "grafana.app/createdBy": "service-account:cef2t2rfm73lsb",
          "grafana.app/updatedBy": "service-account:cef2t2rfm73lsb",
          "grafana.app/updatedTimestamp": "2025-03-07T10:34:46Z"
        }
      },
      "spec": {
        "title": "example"
      }
    }
  ]
}
```

The `metadata.continue` field contains a token to fetch the next page.

Example subsequent request using continue token:

```
GET
/apis/folder.grafana.app/v1/namespaces/default/folders?limit=1&continue=eyJvIjoxNTIsInYiOjE3Nj
E3MDQyMjQyMDcxODksInMiOmZhbHNlfQ== HTTP/1.1
Accept: application/json
Content-Type: application/json
```

```
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>
```

Example subsequent response:

```
HTTP/1.1 200
Content-Type: application/json
{
  "kind": "FolderList",
  "apiVersion": "folder.grafana.app/v1",
  "items": [
    {
      "kind": "Folder",
      "apiVersion": "folder.grafana.app/v1",
      "metadata": {
        "name": "bef30vrzxs3y8e",
        "namespace": "default",
        "uid": "YCtv1FXDsJmTYQoTgcPnfuwZhDZge3uMpXOefaOHjb5Y",
        "resourceVersion": "1741343687000",
        "creationTimestamp": "2025-03-07T10:35:47Z",
        "annotations": {
          "grafana.app/createdBy": "service-account:cef2t2rfm73lsb",
          "grafana.app/updatedBy": "service-account:cef2t2rfm73lsb",
          "grafana.app/updatedTimestamp": "2025-03-07T10:35:47Z"
        }
      },
      "spec": {
        "title": "another folder"
      }
    }
  ]
}
```

Continue making requests with the updated `continue` token until you receive a response without a `continue` field in the metadata, indicating you've reached the last page.

Status Codes:

- **200** – OK
- **401** – Unauthorized
- **403** – Access Denied

Get folder by uid

`GET /apis/folder.grafana.app/v1/namespaces/:namespace/folders/:uid`

Will return the folder given the folder uid.

- namespace: to read more about the namespace to use, see the API overview.
- uid: the unique identifier of the folder to update. this will be the *name* in the folder response

Required permissions

See note in the [introduction]({{< ref "#folder-api" >}}) for an explanation.

Action	Scope
`folders:read`	`folders:*`

Example Request:

```
GET /apis/folder.grafana.app/v1/namespaces/default/folders/aef30vrzxs3y8d HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>
```

Example Response:

```
HTTP/1.1 200
Content-Type: application/json
{
  "kind": "Folder",
  "apiVersion": "folder.grafana.app/v1",
  "metadata": {
    "name": "aef30vrzxs3y8d",
    "namespace": "default",
    "uid": "KCtv1FXDsJmTYQoTgcPnfuwZhDZge3uMpXOefaOHjb4X",
    "resourceVersion": "1741343686000",
    "creationTimestamp": "2025-03-07T10:34:46Z",
    "annotations": {
      "grafana.app/createdBy": "service-account:cef2t2rfm73lsb",
      "grafana.app/updatedBy": "service-account:cef2t2rfm73lsb",
      "grafana.app/updatedTimestamp": "2025-03-07T10:34:46Z",
```

```
      "grafana.app/folder": "fef30w4jaxla8b"
    }
  },
  "spec": {
    "title": "test"
  }
}
```

Note the annotation `grafana.app/folder` which contains the uid of the parent folder.

Status Codes:

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

Create folder

`POST /apis/folder.grafana.app/v1/namespaces/:namespace/folders`

Creates a new folder.

- namespace: to read more about the namespace to use, see the API overview.

Required permissions

See note in the [introduction]({{< ref "#folder-api" >}}) for an explanation.

`folders:create` allows creating folders and subfolders. If granted with scope `folders:uid:general`, allows creating root level folders. Otherwise, allows creating subfolders under the specified folders.

Action	Scope
`folders:create`	`folders:*`
`folders:write`	`folders:*`

Example Request:

```
POST /apis/folder.grafana.app/v1/namespaces/default/folders HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>
```

```
{
  "metadata": {
    "name": "aef30vrzxs3y8d",
    "annotations": {
      "grafana.app/folder": "fef30w4jaxla8b"
    }
  },
  "spec": {
    "title": "child-folder"
  }
}
```

JSON Body schema:

- **metadata.name** – The Grafana [unique identifier]({{< ref "#identifier-id-vs-unique-identifier-uid" >}}). If you do not want to provide this, set metadata.generateName to the prefix you would like for the uid.
- **metadata.annotations.grafana.app/folder** - Optional field, the unique identifier of the parent folder under which the folder should be created. Requires nested folders to be enabled.
- **spec.title** – The title of the folder.

{{< admonition type="note" >}} Custom labels and annotations in the metadata field are supported on some instances, with full support planned for all instances when these APIs reach general availability. If they are not yet supported on your instance, they will be ignored. {{< /admonition >}}

Example Response:

```
HTTP/1.1 200
Content-Type: application/json
{
  "kind": "Folder",
  "apiVersion": "folder.grafana.app/v1",
  "metadata": {
    "name": "eef33r1fprd34d",
    "namespace": "default",
    "uid": "X8momvVZnsXdOqvLD9I4ngqLVif2CgRWXHy9xb2UgjQX",
    "resourceVersion": "1741320415009",
    "creationTimestamp": "2025-03-07T04:06:55Z",
    "labels": {
```

```
      "grafana.app/deprecatedInternalID": "1159"
    },
    "annotations": {
      "grafana.app/folder": "fef30w4jaxla8b",
      "grafana.app/createdBy": "service-account:cef2t2rfm73lsb"
    }
  },
  "spec": {
    "title": "child-folder"
  }
}
```

Status Codes:

- **201** – Created
- **400** – Errors (invalid json, missing or invalid fields, etc)
- **401** – Unauthorized
- **403** – Access denied
- **409** – Conflict (folder with the same uid already exists)

Update folder

`PUT /apis/folder.grafana.app/v1/namespaces/:namespace/folders/:uid`

Updates an existing folder identified by uid.

- namespace: to read more about the namespace to use, see the API overview.
- uid: the unique identifier of the folder to update. this will be the *name* in the folder response

Required permissions

See note in the [introduction]({{< ref "#folder-api" >}}) for an explanation.

Action	Scope
`folders:write`	`folders:*`

Example Request:

```
PUT /apis/folder.grafana.app/v1/namespaces/default/folders/fef30w4jaxla8b HTTP/1.1
Accept: application/json
```

```
Content-Type: application/json
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>

{
  "metadata": {
    "name": "aef30vrzxs3y8d",
    "annotations": {
      "grafana.app/folder": "xkj92m5pqw3vn4"
    }
  },
  "spec": {
    "title": "updated title"
  }
}
```

JSON Body schema:

- **metadata.name** – The [unique identifier]({{< ref "#identifier-id-vs-unique-identifier-uid" >}}) of the folder.
- **metadata.annotations.grafana.app/folder** - Optional field, the unique identifier of the parent folder under which the folder should be - update this to move the folder under a different parent folder. Requires nested folders to be enabled.
- **spec.title** – The title of the folder.

{{< admonition type="note" >}} Custom labels and annotations in the metadata field are supported on some instances, with full support planned for all instances when these APIs reach general availability. If they are not yet supported on your instance, they will be ignored. {{< /admonition >}}

Example Response:

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

{
  "kind": "Folder",
  "apiVersion": "folder.grafana.app/v1",
  "metadata": {
    "name": "fef30w4jaxla8b",
    "namespace": "default",
    "uid": "YaWLsFrMwEaTlIQwX2iMnhHlJuZHtZugps50BQoyjXEX",
    "resourceVersion": "1741345736000",
```

```
    "creationTimestamp": "2025-03-07T11:08:56Z",
    "annotations": {
      "grafana.app/folder": "xkj92m5pqw3vn4",
      "grafana.app/createdBy": "service-account:cef2t2rfm73lsb",
      "grafana.app/updatedBy": "service-account:cef2t2rfm73lsb",
      "grafana.app/updatedTimestamp": "2025-03-07T11:08:56Z"
    }
  },
  "spec": {
    "title": "updated title"
  }
}
```

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 folder has been changed by someone else). With this status code, 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 /apis/folder.grafana.app/v1/namespaces/:namespace/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.

- namespace: to read more about the namespace to use, see the API overview.
- uid: the unique identifier of the folder to delete. this will be the *name* in the folder response

Required permissions

See note in the [introduction]({{< ref "#folder-api" >}}) for an explanation.

Action	Scope
`folders:delete`	`folders:*`

Example Request:

```
DELETE /apis/folder.grafana.app/v1/namespaces/default/folders/fef30w4jaxla8b HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>
```

Example Response:

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

{
  "kind": "Folder",
  "apiVersion": "folder.grafana.app/v1",
  "metadata": {
    "name": "fef30w4jaxla8b",
    "namespace": "default",
    "uid": "YaWLsFrMwEaTlIQwX2iMnhHlJuZHtZugps50BQoyjXEX",
    "resourceVersion": "1741345736000",
    "creationTimestamp": "2025-03-07T11:08:56Z",
    "annotations": {
      "grafana.app/folder": "xkj92m5pqw3vn4",
      "grafana.app/createdBy": "service-account:cef2t2rfm73lsb",
      "grafana.app/updatedBy": "service-account:cef2t2rfm73lsb",
```

```
      "grafana.app/updatedTimestamp": "2025-03-07T11:08:56Z"
    }
  },
  "spec": {
    "title": "updated title"
  }
}
```

Status Codes:

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

□□□ #2
□ □□□ □□□ 2026-10-31 16:09:44 UTC
□ □□□ □□□ 2026-10-31 16:09:47 UTC