Resource History HTTP API

- List resource history
  - Dashboard example
- Pagination
- Retrieve a specific version
- Status codes

{{< 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.

{{< /admonition >}}

The new Grafana APIs track version history for resources. You can retrieve the history of any resource by using the standard List endpoint with specific query parameters. This page documents how to list resource history, using dashboards as an example. The same pattern applies to any resource.

Resource history is available on all stable (GA) API versions (for example, `v1`, `v2`). Alpha or beta versions may not support it.

List resource history

`GET /apis/<group>/<version>/namespaces/<namespace>/<resource>?labelSelector=grafana.app/get-history=true&fieldSelector=metadata.name=<NAME>`

Lists the version history of a specific resource. This uses the standard List endpoint with two required query parameters:

- **`labelSelector`**: Must be set to `grafana.app/get-history=true` to request history instead of a normal list.
- **`fieldSelector`**: Must be set to `metadata.name=<NAME>` to identify the specific resource. `<NAME>` is the `metadata.name` field of the resource (the Grafana UID).

You can control pagination through additional query parameters:

- `limit` (optional): Maximum number of history entries to return per page.
- `continue` (optional): Token from a previous response to fetch the next page.

History entries are returned in reverse chronological order (newest first).

Dashboard example

The following request retrieves the version history for a dashboard with `metadata.name` of `production-overview` in the `default` namespace:

Example request:

```
GET
/apis/dashboard.grafana.app/v1/namespaces/default/dashboards?labelSelector=grafana.app/get-history=true&fieldSelector=metadata.name=production-overview HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>
```

Example response:

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

{
  "kind": "DashboardList",
  "apiVersion": "dashboard.grafana.app/v1",
  "metadata": {
    "resourceVersion": "1758777451428472",
    "continue": "eyJvIjoxNTIsInYiOjE3NjE3MDQyMjQyMDcxODksInMiOmZhbHNlfQ=="
  },
  "items": [
    {
      "kind": "Dashboard",
      "apiVersion": "dashboard.grafana.app/v1",
      "metadata": {
        "name": "production-overview",
        "namespace": "default",
        "uid": "a1b2c3d4-e5f6-7890-g1h2-i3j4k5l6m7n8",
```

```
        "resourceVersion": "1758777451428472",
        "generation": 3,
        "creationTimestamp": "2026-03-15T14:22:10Z",
        "annotations": {
          "grafana.app/updatedBy": "user:u000000001",
          "grafana.app/message": "Added latency panel"
        }
      },
      "spec": {
        "title": "Production Overview",
        "schemaVersion": 41,
        ...
      }
    },
    {
      "kind": "Dashboard",
      "apiVersion": "dashboard.grafana.app/v1",
      "metadata": {
        "name": "production-overview",
        "namespace": "default",
        "uid": "a1b2c3d4-e5f6-7890-g1h2-i3j4k5l6m7n8",
        "resourceVersion": "1758777451428100",
        "generation": 2,
        "creationTimestamp": "2026-03-10T09:15:30Z",
        "annotations": {
          "grafana.app/updatedBy": "user:u000000001",
          "grafana.app/message": "Updated thresholds"
        }
      },
      "spec": {
        "title": "Production Overview",
        "schemaVersion": 41,
        ...
      }
    }
  ]
}
```


Each item in the `items` array represents a historical version of the resource. The following metadata fields are especially relevant for history:

- `metadata.generation`: The version number. This increments each time the resource's spec changes.
- `metadata.resourceVersion`: A unique identifier for this specific version. Use this for pagination and change detection.
- `metadata.creationTimestamp`: When this specific version was saved.
- `metadata.annotations.grafana.app/updatedBy`: The user who saved this version, in the format `<user-type>:<uid>`.
- `metadata.annotations.grafana.app/message`: An optional commit message set during the update. You can set this annotation when updating a resource.
- `spec`: The full resource spec as it existed at that version.

Pagination

You can paginate through history using `limit` and `continue`, the same way you paginate a standard List request.

Example request with pagination:

```
GET
/apis/dashboard.grafana.app/v1/namespaces/default/dashboards?labelSelector=grafana.app/get-
history=true&fieldSelector=metadata.name=production-overview&limit=2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>
```

If the response includes a `metadata.continue` field, pass it as the `continue` query parameter in the next request to fetch the next page. When the response doesn't include a `continue` field, you've reached the last page.

Example request for the next page:

```
GET
/apis/dashboard.grafana.app/v1/namespaces/default/dashboards?labelSelector=grafana.app/get-
history=true&fieldSelector=metadata.name=production-
overview&limit=2&continue=eyJvIjoxNTIsInYiOjE3NjE3MDQyMjQyMDcxODksInMiOmZhbHNlfQ== HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>
```

Retrieve a specific version

To retrieve the full resource at a specific historical version, use the standard Get endpoint with the `resourceVersion` query parameter set to the `metadata.resourceVersion` from the history list.

Example request:

```
GET /apis/dashboard.grafana.app/v1/namespaces/default/dashboards/production-
overview?resourceVersion=1758777451428100 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>
```

This returns the complete resource as it existed at that version.

Status codes

- **200**: OK
- **400**: Bad request (missing `fieldSelector`, invalid `labelSelector`, etc.)
- **401**: Unauthorized
- **403**: Access denied

□□□ #2
□ □□□ □□□ 2026-10-31 16:09:59 UTC
□ □□□ □□□ 2026-10-31 16:10:01 UTC