

Annotations API



- [Requirements](#)
 - [Find Annotations](#)
 - [Create Annotation](#)
 - [Create Annotation in Graphite format](#)
 - [Update Annotation](#)
 - [Patch Annotation](#)
 - [Delete Annotation By Id](#)
 - [Find Annotations Tags](#)
-

```
{ {< docs/shared lookup="developers/deprecated-apis.md" source="grafana"
version="<GRAFANA_VERSION>" > } }
```

Annotations are saved in the Grafana database (sqlite, mysql or postgres). Annotations can be organization annotations that can be shown on any dashboard by configuring an annotation data source - they are filtered by tags. Or they can be tied to a panel on a dashboard and are then only shown on that panel.

Requirements

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.

Find Annotations

```
GET /api/annotations?from=1506676478816&to=1507281278816&tags=tag1&tags=tag2&limit=100
```

Required permissions

See note in the [introduction](#) for an explanation.

Action	Scope
annotations:read	<ul style="list-style-type: none"> • annotations:* • annotations:type:* • dashboards:* • dashboards:uid:* • folders:* • folders:uid:*
{ .no-spacing-list }	

Example Request:

```
GET /api/annotations?from=1506676478816&to=1507281278816&tags=tag1&tags=tag2&limit=100
HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Query Parameters:

- **from**: epoch datetime in milliseconds. Optional.
- **to**: epoch datetime in milliseconds. Optional.
- **limit**: number. Optional - default is 100. Max limit for results returned.
- **alertId**: number. Optional. Find annotations for a specified alert.
- **dashboardId**: Deprecated. Use dashboardUID instead.
- **dashboardUID**: string. Optional. Find annotations that are scoped to a specific dashboard, when dashboardUID presents, dashboardId would be ignored.
- **panelId**: number. Optional. Find annotations that are scoped to a specific panel
- **userId**: number. Optional. Find annotations created by a specific user
- **type**: string. Optional. **alert** | **annotation** Return alerts or user created annotations
- **tags**: string. Optional. Use this to filter organization annotations. Organization annotations are annotations from an annotation data source that are not connected specifically to a dashboard or panel. To do an "AND" filtering with multiple tags, specify the tags parameter multiple times e.g. **tags=tag1&tags=tag2**.

Example Response:

```
HTTP/1.1 200
Content-Type: application/json
[
  {
    "id": 1124,
    "dashboardUID": "uGlb_lG7z",
    "panelId": 2,
```

```

    "userId": 1,
    "userName": "",
    "time": 1507266395000,
    "timeEnd": 1507266395000,
    "text": "test",
    "tags": [
      "tag1",
      "tag2"
    ],
    "data": {}
  },
  {
    "id": 1123,
    "dashboardUID": "jcIIG-07z",
    "panelId": 2,
    "userId": 1,
    "userName": "",
    "time": 1507265111000,
    "text": "test",
    "tags": [
      "tag1",
      "tag2"
    ],
    "data": {}
  }
]

```

{{< admonition type="note" >}} Alerting fields, `alertId`, `alertName`, `newState`, `prevState`, are omitted when the annotation isn't tied to an alert. A missing `alertId` means the same as `0`: not an alert annotation. {{< /admonition >}}

“ Starting in Grafana v6.4 regions annotations are now returned in one entity that now includes the `timeEnd` property.

Create Annotation

Creates an annotation in the Grafana database. The `dashboardUid` and `panelId` fields are optional. If they are not specified then an organization annotation is created and can be queried in any

dashboard that adds the Grafana annotations data source. When creating a region annotation include the timeEnd property.

The format for `time` and `timeEnd` should be epoch numbers in millisecond resolution.

`POST /api/annotations`

Required permissions

See note in the [introduction](#) for an explanation.

Action	Scope
<code>annotations:create</code>	<ul style="list-style-type: none"><code>annotations:*</code><code>annotations:type:*</code><code>dashboards:*</code><code>dashboards:uid:*</code><code>folders:*</code><code>folders:uid:*</code>
<code>{ .no-spacing-list }</code>	

Required JSON Body Fields

- `text`: description of the annotation.

Example Request:

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

{
  "dashboardUID": "jcIIG-07z",
  "time": 1507037197339,
  "timeEnd": 1507180805056,
  "tags": ["tag1", "tag2"],
  "text": "Annotation Description"
}
```

Example Response:

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

```
{
  "message": "Annotation added",
  "id": 1,
}
```

“ The response for this HTTP request is slightly different in versions prior to v6.4. In prior versions you would also get an `endId` if you were creating a region. But in 6.4 regions are represented using a single event with `time` and `timeEnd` properties.

Create Annotation in Graphite format

Creates an annotation by using Graphite-compatible event format. The `when` and `data` fields are optional. If `when` is not specified then the current time will be used as annotation's timestamp. The `tags` field can also be in prior to Graphite `0.10.0` format (string with multiple tags being separated by a space).

POST /api/annotations/graphite

Required permissions

See note in the [introduction](#) for an explanation.

Action	Scope
annotations:create	annotations:type:organization

Example Request:

POST /api/annotations/graphite HTTP/1.1

Accept: application/json

Content-Type: application/json

```
{
  "what": "Event - deploy",
  "tags": ["deploy", "production"],
  "when": 1467844481,
```

```
"data": "deploy of main branch happened at Wed Jul 6 22:34:41 UTC 2016"
}
```

Example Response:

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

{
  "message": "Graphite annotation added",
  "id": 1
}
```

Update Annotation

PUT /api/annotations/:id

Updates all properties of an annotation that matches the specified id. To only update certain property, consider using the [Patch Annotation](#) operation.

Required permissions

See note in the [introduction](#) for an explanation.

Action	Scope
annotations:write	<ul style="list-style-type: none">annotations:*annotations:type:*dashboards:*dashboards:uid:*folders:*folders:uid:*
{ .no-spacing-list }	

Example Request:

```
PUT /api/annotations/1141 HTTP/1.1
Accept: application/json
Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>
Content-Type: application/json
```

```
{
  "time":1507037197339,
  "timeEnd":1507180805056,
  "text":"Annotation Description",
  "tags":["tag3","tag4","tag5"]
}
```

Example Response:

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

{
  "message":"Annotation updated"
}
```

Patch Annotation

PATCH /api/annotations/:id

Updates one or more properties of an annotation that matches the specified id.

This operation currently supports updating of the `text`, `tags`, `time` and `timeEnd` properties.

Required permissions

See note in the [introduction](#) for an explanation.

Action	Scope
<code>annotations:write</code>	<ul style="list-style-type: none"><code>annotations:*</code><code>annotations:type:*</code><code>dashboards:*</code><code>dashboards:uid:*</code><code>folders:*</code><code>folders:uid:*</code>
<code>{ .no-spacing-list }</code>	

Example Request:

```
PATCH /api/annotations/1145 HTTP/1.1
Accept: application/json
```

Authorization: Bearer <SERVICE_ACCOUNT_TOKEN>

Content-Type: application/json

```
{
  "text": "New Annotation Description",
  "tags": ["tag6", "tag7", "tag8"]
}
```

Example Response:

HTTP/1.1 200

Content-Type: application/json

```
{
  "message": "Annotation patched"
}
```

Delete Annotation By Id

DELETE /api/annotations/:id

Deletes the annotation that matches the specified id.

Required permissions

See note in the [introduction](#) for an explanation.

Action	Scope
annotations:delete	<ul style="list-style-type: none">annotations:*annotations:type:*dashboards:*dashboards:uid:*folders:*folders:uid:*
{ .no-spacing-list }	

Example Request:

DELETE /api/annotations/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

```
{
  "message": "Annotation deleted"
}
```

Find Annotations Tags

GET /api/annotations/tags

Find all the event tags created in the annotations.

Required permissions

See note in the [introduction](#) for an explanation.

Action	Scope
annotations:read	N/A

Example Request:

GET /api/annotations/tags?tag=out HTTP/1.1

Accept: application/json

Content-Type: application/json

Authorization: Basic YWRtaW46YWRtaW4=

Query Parameters:

- `tag`: Optional. A string that you can use to filter tags.
- `limit`: Optional. A number, where the default is 100. Max limit for results returned.

Example Response:

HTTP/1.1 200

Content-Type: application/json

```
{
  "result": {
    "tags": [
      {
        "tag": "outage",
        "count": 1
      }
    ]
  }
}
```

#2

□ ### ### 2026-10-31 16:06:20 UTC
□ ### ### 2026-10-31 16:06:22 UTC