Library Element API

- Identifier (id) vs unique identifier (uid)
- Get all library elements
- Get library element by uid
- Get library element by name
- Get library element connections
- Create library element
- Update library element
- Delete library element

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

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.
- `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`: **Deprecated.** A comma separated list of folder IDs to filter the elements by. Use `folderFilterUIDs` instead.
- `folderFilterUIDs`: A comma separated list of folder UIDs 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:

```
GET /api/library-elements?perPage=10 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

{
  "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": {...},
            "version": 1,
            "meta": {
                "folderName": "General",
                "folderUid": "",
                "connectedDashboards": 1,
                "created": "2021-09-27T09:56:17+02:00",
                "updated": "2021-09-27T09:56:17+02:00",
                "createdBy": {
                    "id": 1,
                    "name": "admin",
                    "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
                },
                "updatedBy": {
                    "id": 1,
                    "name": "admin",
                    "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
                }
            }
        },
        {...}
        {...}
    ],
  }
}
```


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:

```
GET /api/library-elements/V--OrYHnz 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

{
    "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,
          "created": "2021-09-27T09:56:17+02:00",
          "updated": "2021-09-27T09:56:17+02:00",
          "createdBy": {
              "id": 1,
              "name": "admin",
              "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
          },
          "updatedBy": {
              "id": 1,
              "name": "admin",
              "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
          }
```

```
      }
    }
}
```

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:

```
GET /api/library-elements/name/API docs Example 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

{
    "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,
                "created": "2021-09-27T09:56:17+02:00",
                "updated": "2021-09-27T09:56:17+02:00",
                "createdBy": {
                    "id": 1,
                    "name": "admin",
                    "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
                },
                "updatedBy": {
                    "id": 1,
                    "name": "admin",
                    "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
                }
            }
        }
    ]
}
```


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:

```
GET /api/library-elements/V--OrYHnz/connections 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

{
    "result": [
        {
            "id": 148, // Deprecated: will be removed in the future.
            "kind": 1,
            "elementId": 25,
            "connectionId": 527,
            "connectionUid": "dHEquNzGz",
            "created": "2021-09-27T10:00:07+02:00",
            "createdBy": {
                "id": 1,
                "name": "admin",
                "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
            }
        }
    ]
}
```

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.
- `uid`: Optional, the unique identifier.

Example Request:

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

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

Example Response:

```
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": "",
```

```
            "connectedDashboards": 0,
            "created": "2021-09-30T09:14:22.378307+02:00",
            "updated": "2021-09-30T09:14:22.378307+02:00",
            "createdBy": {
                "id": 1,
                "name": "admin",
                "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
            },
            "updatedBy": {
                "id": 1,
                "name": "admin",
                "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
            }
        }
    }
}
```


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.
- `version`: Version of the library element you are updating.

- `uid`: Optional, the unique identifier.

Example Request:

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

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

Example Response:

```
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": ""
        },
        "version": 2,
        "meta": {
            "folderName": "General",
            "folderUid": "",
            "connectedDashboards": 0,
```

```
            "created": "2021-09-30T09:14:22+02:00",
            "updated": "2021-09-30T09:25:57.697214+02:00",
            "createdBy": {
                "id": 1,
                "name": "admin",
                "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
            },
            "updatedBy": {
                "id": 1,
                "name": "admin",
                "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
            }
        }
    }
}
```

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.

{{< admonition type="note" >}} You cannot delete a library element that is connected. This operation cannot be reverted. {{< /admonition >}}

Example Request:

```
DELETE /api/library-elements/nErXDvCkzz 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": "Library element deleted",
    "id": 28
}
```

Status Codes:

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

□□□ #2
□ □□□ □□□ 2026-10-31 16:07:15 UTC
□ □□□ □□□ 2026-10-31 16:07:18 UTC