

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

Organization API

The Organization HTTP API is divided in two resources, `/api/org` (current organization) and `/api/orgs` (admin organizations). One big difference between these are that the admin of all organizations API only works with basic authentication, see [Admin Organizations API](#) for more information.

If you are running Grafana Enterprise and have [Fine-grained access control]({{< relref "../enterprise/access-control/_index.md" >}}) enabled, for some endpoints you would need to have relevant permissions. Refer to specific resources to understand what permissions are required.

Current Organization API

Get current Organization

GET `/api/org/`

Required permissions

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

Action	Scope
orgs:read	N/A

Example Request:

```
GET /api/org/ 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,
  "name":"Main Org."
}
```

Get all users within the current organization

GET `/api/org/users`

Returns all org users within the current organization. Accessible to users with org admin role.

Required permissions

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

Action	Scope
org.users:read	users:*

Example Request:

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

Example Response:

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

[
  {
    "orgId": 1,
    "userId": 1,
    "email": "admin@localhost",
    "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56",
    "login": "admin",
    "role": "Admin",
    "lastSeenAt": "2019-08-09T11:02:49+02:00",
    "lastSeenAtAge": "< 1m"
  }
]
```

Get all users within the current organization (lookup)

```
GET /api/org/users/lookup
```

Returns all org users within the current organization, but with less detailed information. Accessible to users with org admin role, admin in any folder or admin of any team. Mainly used by Grafana UI for providing list of users when adding team members and when editing folder/dashboard permissions.

Required permissions

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

Action	Scope
org.users:read	users:*

Example Request:

```
GET /api/org/users/lookup HTTP/1.1
Accept: application/json
```

```
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

Example Response:

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

[
  {
    "userId": 1,
    "login": "admin",
    "avatarUrl": "/avatar/46d229b033af06a191ff2267bca9ae56"
  }
]
```

Updates the given user

```
PATCH /api/org/users/:userId
```

Required permissions

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

Action	Scope
org.users.role:update	users:*

Example Request:

```
PATCH /api/org/users/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "role": "Viewer",
}
```

Example Response:

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

{"message":"Organization user updated"}
```

Delete user in current organization

```
DELETE /api/org/users/:userId
```

Required permissions

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

Action	Scope
org.users:remove	users:*

Example Request:

```
DELETE /api/org/users/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXde9ZWmNrMkZYbk
```

Example Response:

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

{"message":"User removed from organization"}
```

Update current Organization

PUT /api/org

Required permissions

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

Action	Scope
orgs:write	N/A

Example Request:

```
PUT /api/org HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXde9ZWmNrMkZYbk

{
  "name":"Main Org."
}
```

Example Response:

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

{"message":"Organization updated"}
```

Add a new user to the current organization

POST /api/org/users

Adds a global user to the current organization.

Required permissions

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

Action	Scope
org.users:add	users:*

Example Request:

```
POST /api/org/users HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "role": "Admin",
  "loginOrEmail": "admin"
}
```

Example Response:

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

{"message":"User added to organization","userId":1}
```

Admin Organizations API

The Admin Organizations HTTP API does not currently work with an API Token. API Tokens are currently only linked to an organization and an organization role. They cannot be given the permission of server admin, only users can be given that permission. So in order to use these API calls you will have to use Basic Auth and the Grafana user must have the Grafana Admin permission (The default admin user is called `admin` and has permission to use this API).

Get Organization by Id

GET /api/orgs/:orgId

Only works with Basic Authentication (username and password), see [introduction](#).

Required permissions

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

Action	Scope
orgs:read	N/A

Example Request:

```
GET /api/orgs/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
```

Example Response:

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

{
  "id":1,
  "name":"Main Org.",
  "address":{
    "address1":"",
    "address2":"",
    "city":"",
    "zipCode":"",
    "state":"",
    "country":""
  }
}
```

Get Organization by Name

```
GET /api/orgs/name/:orgName
```

Only works with Basic Authentication (username and password), see [introduction](#).

Required permissions

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

Action	Scope	Note
orgs:read	N/A	Needs to be assigned globally.

Example Request:

```
GET /api/orgs/name/Main%20Org HTTP/1.1
Accept: application/json
Content-Type: application/json
```

Example Response:

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

{
  "id":1,
```

```
"name": "Main Org.",
"address": {
  "address1": "",
  "address2": "",
  "city": "",
  "zipCode": "",
  "state": "",
  "country": ""
}
}
```

Create Organization

POST /api/orgs

Only works with Basic Authentication (username and password), see [introduction](#).

Required permissions

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

Action	Scope	Note
orgs:create	N/A	Needs to be assigned globally.

Example Request:

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

{
  "name": "New Org."
}
```

Note: The api will work in the following two ways

1. Need to set GF_USERS_ALLOW_ORG_CREATE=true
2. Set the config value users.allow_org_create to true in ini file

Example Response:

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

{
  "orgId": "1",
  "message": "Organization created"
}
```

Search all Organizations

```
GET /api/orgs?perpage=10&page=1
```

Only works with Basic Authentication (username and password), see [introduction](#).

Required permissions

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

Action	Scope	Note
orgs:read	N/A	Needs to be assigned globally.

Example Request:

```
GET /api/orgs HTTP/1.1
Accept: application/json
Content-Type: application/json
```

Note: The api will only work when you pass the admin name and password to the request HTTP URL, like <http://admin:admin@localhost:3000/api/orgs>

Default value for the `perpage` parameter is `1000` and for the `page` parameter is `0`.

Example Response:

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

[
  {
    "id":1,
    "name":"Main Org."
  }
]
```

Update Organization

```
PUT /api/orgs/:orgId
```

Update Organization, fields *Address 1*, *Address 2*, *City* are not implemented yet. Only works with Basic Authentication (username and password), see [introduction](#).

Required permissions

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

Action	Scope
orgs:write	N/A

Example Request:


```
PUT /api/orgs/1 HTTP/1.1
Accept: application/json
Content-Type: application/json

{
  "name": "Main Org 2."
}
```

Example Response:

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

{"message": "Organization updated"}
```

Delete Organization

```
DELETE /api/orgs/:orgId
```

Only works with Basic Authentication (username and password), see [introduction](#).

Required permissions

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

Action	Scope
orgs:delete	N/A

Example Request:

```
DELETE /api/orgs/1 HTTP/1.1
Accept: application/json
```

Example Response:

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

{"message": "Organization deleted"}
```

Get Users in Organization

```
GET /api/orgs/:orgId/users
```

Only works with Basic Authentication (username and password), see [introduction](#).

Required permissions

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

--	--

Action	Scope
org.users:read	users:*

Example Request:

```
GET /api/orgs/1/users HTTP/1.1
Accept: application/json
Content-Type: application/json
```

Note: The api will only work when you pass the admin name and password to the request HTTP URL, like <http://admin:admin@localhost:3000/api/orgs/1/users>

Example Response:

```
HTTP/1.1 200
Content-Type: application/json
[
  {
    "orgId":1,
    "userId":1,
    "email":"admin@mygraf.com",
    "login":"admin",
    "role":"Admin"
  }
]
```

Add User in Organization

```
POST /api/orgs/:orgId/users
```

Only works with Basic Authentication (username and password), see [introduction](#).

Required permissions

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

Action	Scope
org.users:add	users:*

Example Request:

```
POST /api/orgs/1/users HTTP/1.1
Accept: application/json
Content-Type: application/json

{
  "loginOrEmail":"user",
  "role":"Viewer"
}
```

Example Response:

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

{"message":"User added to organization", "userId": 1}
```

Update Users in Organization

PATCH /api/orgs/:orgId/users/:userId

Only works with Basic Authentication (username and password), see [introduction](#).

Required permissions

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

Action	Scope
org.users.role:update	users:*

Example Request:

```
PATCH /api/orgs/1/users/2 HTTP/1.1
Accept: application/json
Content-Type: application/json

{
  "role": "Admin"
}
```

Example Response:

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

{"message":"Organization user updated"}
```

Delete User in Organization

DELETE /api/orgs/:orgId/users/:userId

Only works with Basic Authentication (username and password), see [introduction](#).

Required permissions

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

Action	Scope
org.users.remove	users:*

Example Request:

```
DELETE /api/orgs/1/users/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
```

Example Response:

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

{"message":"User removed from organization"}
```