

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

User API

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.

Search Users

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

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

Action	Scope
users:read	global.users:*

Example Request:

```
GET /api/users HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Default value for the **perpage** parameter is 1000 and for the **page** parameter is 1. Requires basic authentication and that the authenticated user is a Grafana Admin.

Example Response:

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

```
[
  {
    "id": 1,
    "name": "Admin",
    "login": "admin",
    "email": "admin@mygraf.com",
    "isAdmin": true,
    "isDisabled": false,
```

```

    "lastSeenAt": "2020-04-10T20:29:27+03:00",
    "lastSeenAtAge": "2m",
    "authLabels": ["OAuth"]
  },
  {
    "id": 2,
    "name": "User",
    "login": "user",
    "email": "user@mygraf.com",
    "isAdmin": false,
    "isDisabled": false,
    "lastSeenAt": "2020-01-24T12:38:47+02:00",
    "lastSeenAtAge": "2M",
    "authLabels": []
  }
]

```

Search Users with Paging

GET /api/users/search?perpage=10&page=1&query=mygraf

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

Action	Scope
users:read	global.users:*

Example Request:

```

GET /api/users/search?perpage=10&page=1&query=mygraf HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=

```

Default value for the **perpage** parameter is 1000 and for the **page** parameter is 1. The **totalCount** field in the response can be used for pagination of the user list E.g. if **totalCount** is equal to 100 users and the **perpage** parameter is set to 10 then there are 10 pages of users. The **query** parameter is optional and it will return results where the query value is contained in one of the **name**, **login** or **email** fields. Query values with spaces need to be URL encoded e.g. **query=Jane%20Doe**.

Requires basic authentication and that the authenticated user is a Grafana Admin.

Example Response:

```

HTTP/1.1 200
Content-Type: application/json
{
  "totalCount": 2,
  "users": [
    {
      "id": 1,
      "name": "Admin",
      "login": "admin",
      "email": "admin@mygraf.com",
      "isAdmin": true,
      "isDisabled": false,
      "lastSeenAt": "2020-04-10T20:29:27+03:00",
      "lastSeenAtAge": "2m",
      "authLabels": ["OAuth"]
    },
    {
      "id": 2,
      "name": "User",
      "login": "user",
      "email": "user@mygraf.com",
      "isAdmin": false,
      "isDisabled": false,
      "lastSeenAt": "2020-01-24T12:38:47+02:00",
      "lastSeenAtAge": "2M",
      "authLabels": []
    }
  ],
  "page": 1,
  "perPage": 10
}

```

Get single user by Id

GET /api/users/:id

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

Action	Scope
users:read	users:*

Example Request:

```
GET /api/users/1 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Requires basic authentication and that the authenticated user is a Grafana Admin.

Example Response:

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

```
{
  "id": "1",
  "email": "user@mygraf.com",
  "name": "admin",
  "login": "admin",
  "theme": "light",
  "orgId": 1,
  "isGrafanaAdmin": true,
  "isDisabled": true,
  "isExternal": false,
  "authLabels": [],
  "updatedAt": "2019-09-09T11:31:26+01:00",
  "createdAt": "2019-09-09T11:31:26+01:00",
  "avatarUrl": ""
}
```

Get single user by Username(login) or Email

```
GET /api/users/lookup?loginOrEmail=user@mygraf.com
```

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

Action	Scope
users:read	global.users:*

Example Request using the email as option:

```
GET /api/users/lookup?loginOrEmail=user@mygraf.com HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

Example Request using the username as option:

```
GET /api/users/lookup?loginOrEmail=admin HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Requires basic authentication and that the authenticated user is a Grafana Admin.

Example Response:

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

```
{
  "id": 1,
  "email": "user@mygraf.com",
  "name": "admin",
  "login": "admin",
  "theme": "light",
  "orgId": 1,
  "isGrafanaAdmin": true,
  "isDisabled": false,
  "isExternal": false,
  "authLabels": null,
  "updatedAt": "2019-09-25T14:44:37+01:00",
  "createdAt": "2019-09-25T14:44:37+01:00",
  "avatarUrl": ""
}
```

User Update

```
PUT /api/users/:id
```

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

Action	Scope
users:write	users:*

Example Request:

```
PUT /api/users/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

```
{
  "email": "user@mygraf.com",
  "name": "User2",
  "login": "user",
  "theme": "light"
}
```

Requires basic authentication and that the authenticated user is a Grafana Admin.

Example Response:

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

```
{"message": "User updated"}
```

Get Organizations for user

```
GET /api/users/:id/orgs
```

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

Action	Scope
users:read	users:*

Example Request:

```
GET /api/users/1/orgs HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Requires basic authentication and that the authenticated user is a Grafana Admin.

Example Response:

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

```
[
  {
    "orgId": 1,
    "name": "Main Org.",
    "role": "Admin"
```

```
}
]
```

Get Teams for user

GET /api/users/:id/teams

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

Action	Scope
users.teams:read	users:*

Example Request:

```
GET /api/users/1/teams HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Requires basic authentication and that the authenticated user is a Grafana Admin.

Example Response:

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

```
[
  {
    "id":1,
    "orgId":1,
    "name":"team1",
    "email":"",
    "avatarUrl":"/avatar/3fcfe295eae3bcb67a49349377428a66",
    "memberCount":1
  }
]
```

User

Actual User

GET /api/user

Example Request:

```
GET /api/user HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Requires basic authentication.

Example Response:

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

```
{
  "id":1,
  "email":"admin@mygraf.com",
  "name":"Admin",
  "login":"admin",
  "theme":"light",
  "orgId":1,
  "isGrafanaAdmin":true,
  "isDisabled":false
  "isExternal": false,
  "authLabels": [],
  "updatedAt": "2019-09-09T11:31:26+01:00",
  "createdAt": "2019-09-09T11:31:26+01:00",
  "avatarUrl": ""
}
```

Change Password

```
PUT /api/user/password
```

Changes the password for the user. Requires basic authentication.

Example Request:

```
PUT /api/user/password HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

```
{
  "oldPassword": "old_password",
  "newPassword": "new_password"
}
```

Example Response:

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



```
{"message": "User password changed"}
```

Change Password with a Script

If you need to change a password with a script, here is an example of changing the Admin password using curl with basic auth:

```
curl -X PUT -H "Content-Type: application/json" -d '{
  "oldPassword": "oldpass",
  "newPassword": "newpass",
  "confirmNew": "newpass"
}' http://admin:oldpass@<your_grafana_host>:3000/api/user/password
```

Switch user context for a specified user

POST /api/users/:userId/using/:organizationId

Switch user context to the given organization. Requires basic authentication and that the authenticated user is a Grafana Admin.

Example Request:

POST /api/users/7/using/2 HTTP/1.1
Authorization: Basic YWRtaW46YWRtaW4=

Example Response:

HTTP/1.1 200
Content-Type: application/json

```
{"message": "Active organization changed"}
```

Switch user context for signed in user

POST /api/user/using/:organizationId

Switch user context to the given organization.

Example Request:

POST /api/user/using/2 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

Example Response:

HTTP/1.1 200
Content-Type: application/json

```
{"message": "Active organization changed"}
```

Organizations of the actual User

GET /api/user/orgs

Return a list of all organizations of the current user. Requires basic authentication.

Example Request:

```
GET /api/user/orgs HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Example Response:

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

```
[
  {
    "orgId":1,
    "name":"Main Org.",
    "role":"Admin"
  }
]
```

Teams that the actual User is member of

GET /api/user/teams

Return a list of all teams that the current user is member of.

Example Request:

```
GET /api/user/teams HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

Example Response:

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

```
[
  {
    "id": 1,
    "orgId": 1,
    "name": "MyTestTeam",
    "email": "",
  }
]
```

```

        "avatarUrl": "\\avatar\\3f49c15916554246daa714b9bd0ee398",
        "memberCount": 1
    }
]

```

Star a dashboard

POST /api/user/stars/dashboard/:dashboardId

Stars the given Dashboard for the actual user.

Example Request:

```

POST /api/user/stars/dashboard/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":"Dashboard starred!"}

```

Unstar a dashboard

DELETE /api/user/stars/dashboard/:dashboardId

Deletes the starring of the given Dashboard for the actual user.

Example Request:

```

DELETE /api/user/stars/dashboard/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":"Dashboard unstarred"}

```

Auth tokens of the actual User

GET /api/user/auth-tokens

Return a list of all auth tokens (devices) that the actual user currently have logged in from.

Example Request:

```
GET /api/user/auth-tokens HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

Example Response:

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

```
[
  {
    "id": 361,
    "isActive": true,
    "clientIp": "127.0.0.1",
    "browser": "Chrome",
    "browserVersion": "72.0",
    "os": "Linux",
    "osVersion": "",
    "device": "Other",
    "createdAt": "2019-03-05T21:22:54+01:00",
    "seenAt": "2019-03-06T19:41:06+01:00"
  },
  {
    "id": 364,
    "isActive": false,
    "clientIp": "127.0.0.1",
    "browser": "Mobile Safari",
    "browserVersion": "11.0",
    "os": "iOS",
    "osVersion": "11.0",
    "device": "iPhone",
    "createdAt": "2019-03-06T19:41:19+01:00",
    "seenAt": "2019-03-06T19:41:21+01:00"
  }
]
```

Revoke an auth token of the actual User

```
POST /api/user/revoke-auth-token
```

Revokes the given auth token (device) for the actual user. User of issued auth token (device) will no longer be logged in and will be required to authenticate again upon next activity.

Example Request:

```
POST /api/user/revoke-auth-token HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pUlY2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

```
{
  "authTokenId": 364
}
```

Example Response:

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

```
{
  "message": "User auth token revoked"
}
```