

```
+++ title = "Team HTTP API " description = "Grafana Team HTTP API" keywords = ["grafana", "http",  
"documentation", "api", "team", "teams", "group"] aliases = ["/docs/grafana/latest/http_api/team/"] +++
```

Team API

This API can be used to manage Teams and Team Memberships.

Access to these API endpoints is restricted as follows:

- All authenticated users are able to view details of teams they are a member of.
- Organization Admins are able to manage all teams and team members.
- If you enable `editors_can_admin` configuration flag, then Organization Editors can create teams and manage teams where they are Admin.
 - If you enable `editors_can_admin` configuration flag, Editors can find out whether a team that they are not members of exists by trying to create a team with the same name.

If you are running Grafana Enterprise and have [Fine-grained access control]({{< relref "../enterprise/access-control/_index.md" >}}) enabled, access to endpoints will be controlled by Fine-grained access control permissions. Refer to specific endpoints to understand what permissions are required.

Team Search With Paging

```
GET /api/teams/search?perpage=50&page=1&query=myteam
```

or

```
GET /api/teams/search?name=myteam
```

Required permissions

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

Action	Scope
teams:read	teams:*

Example Request:

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

Example Response:

```
HTTP/1.1 200  
Content-Type: application/json  
  
{  
  "totalCount": 1,  
  "teams": [  
    {
```

```

    "id": 1,
    "orgId": 1,
    "name": "MyTestTeam",
    "email": "",
    "avatarUrl": "\/avatar\/3f49c15916554246daa714b9bd0ee398",
    "memberCount": 1
  }
],
"page": 1,
"perPage": 1000
}

```

Using the query parameter

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 teams list E.g. if `totalCount` is equal to 100 teams and the `perPage` parameter is set to 10 then there are 10 pages of teams.

The `query` parameter is optional and it will return results where the query value is contained in the `name` field. Query values with spaces need to be URL encoded e.g. `query=my%20team`.

Using the name parameter

The `name` parameter returns a single team if the parameter matches the `name` field.

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found (if searching by name)

Get Team By Id

```
GET /api/teams/:id
```

Required permissions

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

Action	Scope
teams:read	teams:*

Example Request:

```

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

```

Example Response:

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

{
  "id": 1,
  "orgId": 1,
  "name": "MyTestTeam",
  "email": "",
  "created": "2017-12-15T10:40:45+01:00",
  "updated": "2017-12-15T10:40:45+01:00"
}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found

Add Team

The Team `name` needs to be unique. `name` is required and `email` , `orgId` is optional.

POST `/api/teams`

Required permissions

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

Action	Scope
teams:create	N/A

Example Request:

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

{
  "name": "MyTestTeam",
  "email": "email@test.com",
  "orgId": 2
}
```

Example Response:

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

{"message":"Team created","teamId":2}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **409** - Team name is taken

Update Team

There are two fields that can be updated for a team: `name` and `email` .

```
PUT /api/teams/:id
```

Required permissions

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

Action	Scope
teams:write	teams:*

Example Request:

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

{
  "name": "MyTestTeam",
  "email": "email@test.com"
}
```

Example Response:

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

{"message":"Team updated"}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found
- **409** - Team name is taken

Delete Team By Id

```
DELETE /api/teams/:id
```

Required permissions

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

Action	Scope
teams:delete	teams:*

Example Request:

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

Example Response:

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

{"message": "Team deleted"}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Failed to delete Team. ID not found

Get Team Members

```
GET /api/teams/:teamId/members
```

Required permissions

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

Action	Scope
teams.permissions:read	teams:*

Example Request:

```
GET /api/teams/1/members 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,
    "teamId": 1,
    "userId": 3,
    "email": "user1@email.com",
    "login": "user1",
    "avatarUrl": "\/avatar\/1b3c32f6386b0185c40d359cdc733a79"
  },
  {
    "orgId": 1,
    "teamId": 1,
    "userId": 2,
    "email": "user2@email.com",
    "login": "user2",
    "avatarUrl": "\/avatar\/cad3c68da76e45d10269e8ef02f8e73e"
  }
]
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied

Add Team Member

POST /api/teams/:teamId/members

Required permissions

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

Action	Scope
teams.permissions:write	teams:*

Example Request:

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

{
  "userId": 2
}
```

Example Response:

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

{"message":"Member added to Team"}
```

Status Codes:

- **200** - Ok
- **400** - User is already added to this team
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found

Remove Member From Team

```
DELETE /api/teams/:teamId/members/:userId
```

Required permissions

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

Action	Scope
teams.permissions:write	teams:*

Example Request:

```
DELETE /api/teams/2/members/3 HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Basic YWRtaW46YWRtaW4=
```

Example Response:

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

{"message":"Team Member removed"}
```

Status Codes:

- **200** - Ok
- **401** - Unauthorized
- **403** - Permission denied
- **404** - Team not found/Team member not found

Get Team Preferences

```
GET /api/teams/:teamId/preferences
```

Required permissions

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

Action	Scope
teams:read	teams:*

Example Request:

```
GET /api/teams/2/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk
```

Example Response:

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

{
  "theme": "",
  "homeDashboardId": 0,
  "timezone": ""
}
```

Update Team Preferences

PUT /api/teams/:teamId/preferences

Required permissions

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

Action	Scope
teams:write	teams:*

Example Request:

```
PUT /api/teams/2/preferences HTTP/1.1
Accept: application/json
Content-Type: application/json
Authorization: Bearer eyJrIjoiT0tTcG1pU1Y2RnVKZTFVaDFsNFZXdE9ZWmNrMkZYbk

{
  "theme": "dark",
  "homeDashboardId": 39,
  "timezone": "utc"
}
```

JSON Body Schema:

- **theme** - One of: `light` , `dark` , or an empty string for the default theme
- **homeDashboardId** - The numerical `id` of a dashboard, default: `0`
- **timezone** - One of: `utc` , `browser` , or an empty string for the default

Omitting a key will cause the current value to be replaced with the system default value.

Example Response:

```
HTTP/1.1 200
Content-Type: text/plain; charset=utf-8

{
  "message": "Preferences updated"
}
```