

The REST API is now versioned. For more information, see "About API versioning."

SCIM

Use the REST API to control and manage your GitHub organization members access with SCIM.

About SCIM

SCIM Provisioning for Organizations

These endpoints are used by SCIM-enabled Identity Providers (IdPs) to automate provisioning of GitHub Enterprise Cloud organization membership and are based on version 2.0 of the [SCIM standard](#). IdPs should use the base URL `https://api.github.com/scim/v2/organizations/{org}/` for GitHub Enterprise Cloud SCIM endpoints.

Notes:

- These endpoints are only available for individual organizations that use GitHub Enterprise Cloud with SAML SSO enabled. For more information about SCIM, see "[About SCIM for organizations](#)." For more information about authorizing a token for a SAML SSO organization, see "[Authenticating to the REST API](#)."
- These endpoints cannot be used with an enterprise account or with an organization with managed users.

Authentication

You must authenticate as an owner of a GitHub Enterprise Cloud organization to use these endpoints. The REST API expects an OAuth 2.0 Bearer token (for example, a GitHub App user access token) to be included in the `Authorization` header. If you use a personal access token (classic) for authentication, it must have the `admin:org` scope and you must also [authorize it for use with your SAML SSO organization](#).

Mapping of SAML and SCIM data

The SAML IdP and the SCIM client must use matching `NameID` and `userName` values for each user. This allows a user authenticating through SAML to be linked to their provisioned SCIM identity.

Supported SCIM User attributes

Name	Type	Description
<code>userName</code>	<code>string</code>	The username for the user.
<code>name.givenName</code>	<code>string</code>	The first name of the user.
<code>name.familyName</code>	<code>string</code>	The last name of the user.
<code>emails</code>	<code>array</code>	List of user emails.
<code>externalId</code>	<code>string</code>	This identifier is generated by the SAML provider, and is used as a unique ID by the SAML provider to match against a GitHub

user. You can find the `externalID` for a user either at the SAML provider, or using the [List SCIM provisioned identities](#) endpoint and filtering on other known attributes, such as a user's GitHub username or email address.

id	string	Identifier generated by the GitHub SCIM endpoint.
active	boolean	Used to indicate whether the identity is active (true) or should be deprovisioned (false).

Note: These endpoints are case sensitive. For example, the first letter in the `Users` endpoint must be capitalized:

```
GET /scim/v2/organizations/{org}/Users/{scim_user_id}
```

List SCIM provisioned identities

✔ Works with [GitHub Apps](#)

Retrieves a paginated list of all provisioned organization members, including pending invitations. If you provide the `filter` parameter, the resources for all matching provisions members are returned.

When a user with a SAML-provisioned external identity leaves (or is removed from) an organization, the account's metadata is immediately removed. However, the returned list of user accounts might not always match the organization or enterprise member list you see on GitHub Enterprise Cloud. This can happen in certain cases where an external identity associated with an organization will not match an organization member:

- When a user with a SCIM-provisioned external identity is removed from an organization, the account's metadata is preserved to allow the user to re-join the organization in the future.
- When inviting a user to join an organization, you can expect to see their external identity in the results before they accept the invitation, or if the invitation is cancelled (or never accepted).
- When a user is invited over SCIM, an external identity is created that matches with the invitee's email address. However, this identity is only linked to a user account when the user accepts the invitation by going through SAML SSO.

The returned list of external identities can include an entry for a `null` user. These are unlinked SAML identities that are created when a user goes through the following Single Sign-On (SSO) process but does not sign in to their GitHub Enterprise Cloud account after completing SSO:

- 1 The user is granted access by the IdP and is not a member of the GitHub Enterprise Cloud organization.
- 2 The user attempts to access the GitHub Enterprise Cloud organization and initiates the SAML SSO process, and is not currently signed in to their GitHub Enterprise Cloud account.
- 3 After successfully authenticating with the SAML SSO IdP, the `null` external identity entry is created and the user is prompted to sign in to their GitHub Enterprise Cloud account:
 - If the user signs in, their GitHub Enterprise Cloud account is linked to this entry.
 - If the user does not sign in (or does not create a new account when prompted), they are not added to the GitHub Enterprise Cloud organization, and the external identity `null` entry remains in place.

Parameters for "List SCIM provisioned identities"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

org string **Required**

The organization name. The name is not case sensitive.

Query parameters

startIndex integer

Used for pagination: the index of the first result to return.

count integer

Used for pagination: the number of results to return.

filter string

Filters results using the equals query parameter operator (`eq`). You can filter results that are equal to `id` , `userName` , `emails` , and `externalId` . For example, to search for an identity with the `userName` Octocat, you would use this query:
`?filter=userName%20eq%20\"Octocat\"` .
To filter results for the identity with the email `octocat@github.com` , you would use this query:
`?filter=emails%20eq%20\"octocat@github.com\"` .

HTTP response status codes for "List SCIM provisioned identities"

Status code	Description
200	OK
304	Not modified
400	Bad request
403	Forbidden
404	Resource not found
429	Too many requests

Code samples for "List SCIM provisioned identities"

Example 1: Status Code 200 ▾

GET

/scim/v2/organizations/{org}/Users

cURL

JavaScript

GitHub CLI

```
curl -L \ -H "Accept: application/scim+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ https://api.github.com/scim/v2/organizations/ORG/Users
```

Response with filter

Example response

Response schema

Status: 200

```
{ "schemas": [ "urn:ietf:params:scim:api:messages:2.0:ListResponse" ], "totalResults": 1, "itemsPerPage": 1, "startIndex": 1, "Resources": [ { "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:User" ], "id": "5fc0c238-1112-11e8-8e45-920c87bdbc75", "externalId": "00u1dhb1fkIGP7RL1d8", "userName": "octocat@github.com", "displayName": "Mona Octocat", "name": { "givenName": "Mona", "familyName": "Octocat", "formatted": "Mona Octocat" }, "emails": [ { "value": "octocat@github.com", "primary": true } ], "active": true, "meta": { "resourceType": "User", "created": "2018-02-13T15:05:24.000-08:00", "lastModified": "2018-02-13T15:05:55.000-08:00", "location": "https://api.github.com/scim/v2/organizations/octo-org/Users/5fc0c238-1112-11e8-8e45-
```

Provision and invite a SCIM user

Works with [GitHub Apps](#)

Provisions organization membership for a user, and sends an activation email to the email address. If the user was previously a member of the organization, the invitation will reinstate any former privileges that the user had. For more information about reinstating former members, see "[Reinstating a former member of your organization](#)."

Parameters for "Provision and invite a SCIM user"

Headers

accept string
Setting to `application/vnd.github+json` is recommended.

Path parameters

org string **Required**
The organization name. The name is not case sensitive.

Body parameters

userName string **Required**
Configured by the admin. Could be an email, login, or username

displayName string
The name of the user, suitable for display to end-users

name object **Required**

► Properties of `name`

emails array of objects **Required**
user emails

► Properties of `emails`

schemas	array of strings
externalId	string
groups	array of strings
active	boolean

HTTP response status codes for "Provision and invite a SCIM user"

Status code	Description
201	Created
304	Not modified
400	Bad request
403	Forbidden
404	Resource not found
409	Conflict
500	Internal server error

Code samples for "Provision and invite a SCIM user"

POST /scim/v2/organizations/{org}/Users

cURL

JavaScript

GitHub CLI

```
curl -L \ -X POST \ -H "Accept: application/scim+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ https://api.github.com/scim/v2/organizations/ORG/Users \ -d '{"userName":"octocat","name":"Monalisa Octocat","emails":[{"value":"mona.octocat@github.com","primary":true}]}'
```

Response

Example response

Response schema

Status: 201

```
{ "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:User" ], "id": "edefdfedf-050c-11e7-8d32", "externalId": "a7d0f98382", "userName": "mona.octocat@okta.example.com", "displayName": "Monalisa Octocat", "name": { "givenName": "Monalisa", "familyName": "Octocat", "formatted": "Monalisa Octocat" }, "emails": [ { "value": "mona.octocat@okta.example.com", "primary": true }, { "value": "monalisa@octocat.github.com" } ], "active": true, "meta": { "resourceType": "User", "created": "2017-03-09T16:11:13-05:00", "lastModified": "2017-03-09T16:11:13-05:00", "location": "https://api.github.com/scim/v2/organizations/octo-org/Users/edefdfedf-050c-11e7-8d32" } }
```

Get SCIM provisioning information for a user

Parameters for "Get SCIM provisioning information for a user"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

org string **Required**

The organization name. The name is not case sensitive.

scim_user_id string **Required**

The unique identifier of the SCIM user.

HTTP response status codes for "Get SCIM provisioning information for a user"

Status code	Description
200	OK
304	Not modified
403	Forbidden
404	Resource not found

Code samples for "Get SCIM provisioning information for a user"

GET

/scim/v2/organizations/{org}/Users/{scim_user_id}

cURL

JavaScript

GitHub CLI

```
curl -L \ -H "Accept: application/scim+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ https://api.github.com/scim/v2/organizations/ORG/Users/SCIM_USER_ID
```

Response

Example response

Response schema

Status: 200

```
{ "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:User" ], "id": "edefdfedf-050c-11e7-8d32", "externalId": "a7d0f98382", "userName": "mona.octocat@okta.example.com", "displayName": "Monalisa Octocat", "name": { "givenName": "Monalisa", "familyName": "Octocat", "formatted": "Monalisa Octocat" }, "emails": [ { "value": "mona.octocat@okta.example.com", "primary": true }, { "value": "monalisa@octocat.github.com" } ], "active": true, "meta": { "resourceType": "User", "created": "2017-03-09T16:11:13-05:00", "lastModified": "2017-03-09T16:11:13-05:00", "location": "https://api.github.com/scim/v2/organizations/octo-org/Users/edefdfedf-050c-11e7-8d32" } }
```

Update a provisioned organization membership

✔ Works with [GitHub Apps](#)

Replaces an existing provisioned user's information. You must provide all the information required for the user as if you were provisioning them for the first time. Any existing user information that you don't provide will be removed. If you want to only update a specific attribute, use the [Update an attribute for a SCIM user](#) endpoint instead.

You must at least provide the required values for the user: `userName` , `name` , and `emails` .

Warning: Setting `active: false` removes the user from the organization, deletes the external identity, and deletes the associated `{scim_user_id}` .

Parameters for "Update a provisioned organization membership"

Headers

accept string
Setting to `application/vnd.github+json` is recommended.

Path parameters

org string **Required**
The organization name. The name is not case sensitive.

scim_user_id string **Required**
The unique identifier of the SCIM user.

Body parameters

schemas array of strings

displayName string
The name of the user, suitable for display to end-users

externalId string

groups array of strings

active boolean

userName string **Required**
Configured by the admin. Could be an email, login, or username

name object **Required**

► Properties of `name`

emails array of objects **Required**
user emails

► Properties of `emails`

HTTP response status codes for "Update a provisioned organization membership"

Status code	Description
200	OK
304	Not modified
403	Forbidden
404	Resource not found

Code samples for "Update a provisioned organization membership"

PUT

/scim/v2/organizations/{org}/Users/{scim_user_id}

cURL

JavaScript

GitHub CLI

```
curl -L \ -X PUT \ -H "Accept: application/scim+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ https://api.github.com/scim/v2/organizations/ORG/Users/SCIM_USER_ID \ -d '{"userName":"octocat","name":"Monalisa Octocat","emails":[{"value":"mona.octocat@github.com","primary":true}]}'
```

Response

Example response

Response schema

Status: 200

```
{ "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:User" ], "id": "edefdfedf-050c-11e7-8d32", "externalId": "a7d0f98382", "userName": "mona.octocat@okta.example.com", "displayName": "Monalisa Octocat", "name": { "givenName": "Monalisa", "familyName": "Octocat", "formatted": "Monalisa Octocat" }, "emails": [ { "value": "mona.octocat@okta.example.com", "primary": true }, { "value": "monalisa@octocat.github.com" } ], "active": true, "meta": { "resourceType": "User", "created": "2017-03-09T16:11:13-05:00", "lastModified": "2017-03-09T16:11:13-05:00", "location": "https://api.github.com/scim/v2/organizations/octo-org/Users/edefdfedf-050c-11e7-8d32" } }
```

Update an attribute for a SCIM user

✔ Works with [GitHub Apps](#)

Allows you to change a provisioned user's individual attributes. To change a user's values, you must provide a specific `Operations` JSON format that contains at least one of the `add`, `remove`, or `replace` operations. For examples and more information on the SCIM operations format, see the [SCIM specification](#).

Note: Complicated SCIM `path` selectors that include filters are not supported. For example, a `path` selector defined as `"path": "emails[type eq \"work\"]"` will not work.

Warning: If you set `active:false` using the `replace` operation (as shown in the JSON example below), it removes the user from the organization, deletes the external identity, and deletes the associated `:scim_user_id`.


```
{
  "Operations":[{
    "op":"replace",
    "value":{"
      "active":false
    }
  }]
}
```

Parameters for "Update an attribute for a SCIM user"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

org string **Required**

The organization name. The name is not case sensitive.

scim_user_id string **Required**

The unique identifier of the SCIM user.

Body parameters

schemas array of strings

Operations array of objects **Required**

Set of operations to be performed

► Properties of `Operations`

HTTP response status codes for "Update an attribute for a SCIM user"

Status code	Description
200	OK
304	Not modified
400	Bad request
403	Forbidden
404	Resource not found
429	Too Many Requests

Code samples for "Update an attribute for a SCIM user"

PATCH `/scim/v2/organizations/{org}/Users/{scim_user_id}`

cURL

JavaScript

GitHub CLI

```
curl -L \ -X PATCH \ -H "Accept: application/scim+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ https://api.github.com/scim/v2/organizations/ORG/Users/SCIM_USER_ID \ -d '{"Operations":[{"op":"replace","value":{"displayName":"Octocat"}}]}'
```

Response

Example responseResponse schema

Status: 200

{ "schemas": ["urn:ietf:params:scim:schemas:core:2.0:User"], "id": "edefdfedf-050c-11e7-8d32", "externalId": "a7d0f98382", "userName": "mona.octocat@okta.example.com", "displayName": "Monalisa Octocat", "name": { "givenName": "Monalisa", "familyName": "Octocat", "formatted": "Monalisa Octocat" }, "emails": [{ "value": "mona.octocat@okta.example.com", "primary": true }, { "value": "monalisa@octocat.github.com" }], "active": true, "meta": { "resourceType": "User", "created": "2017-03-09T16:11:13-05:00", "lastModified": "2017-03-09T16:11:13-05:00", "location": "https://api.github.com/scim/v2/organizations/octo-org/Users/edefdfedf-050c-11e7-8d32" } }

Delete a SCIM user from an organization

✔ Works with [GitHub Apps](#)

Parameters for "Delete a SCIM user from an organization"

Headers

accept string

Setting to `application/vnd.github+json` is recommended.

Path parameters

org string **Required**

The organization name. The name is not case sensitive.

scim_user_id string **Required**

The unique identifier of the SCIM user.

HTTP response status codes for "Delete a SCIM user from an organization"

Status code	Description
204	No Content
304	Not modified
403	Forbidden
404	Resource not found

Code samples for "Delete a SCIM user from an organization"

DELETE

/scim/v2/organizations/{org}/Users/{scim_user_id}

cURL

JavaScript

GitHub CLI

```
curl -L \ -X DELETE \ -H "Accept: application/vnd.github+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-API-Version: 2022-11-28" \ https://api.github.com/scim/v2/organizations/ORG/Users/SCIM_USER_ID
```

Response

Status: 204

Legal