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

SCIM

Use the REST API to automate user creation and team memberships with SCIM.

About SCIM

Note: SCIM for GitHub Enterprise Server is currently in private beta and is subject to change. For access to the beta, contact your account manager on GitHub's Sales team. Please provide feedback in the GitHub Community discussion.

Warning: The beta is exclusively for testing and feedback, and no support is available. GitHub recommends testing with a staging instance. For more information, see "Setting up a staging instance."

These endpoints only support authentication using a personal access token (classic). For more information, see "Managing your personal access tokens."

GitHub Enterprise Server provides endpoints for use by SCIM-enabled Identity Providers (IdPs). An integration on the IdP can use the REST API to automatically provision, manage, or deprovision user accounts on a GitHub Enterprise Server instance that uses SAML single sign-on (SSO) for authentication. For more information about SAML SSO, see "About SAML for enterprise IAM."

These endpoints are based on SCIM 2.0. For more information, refer to your IdP's documentation or see the specification on the IETF website.

Root URLs

An IdP can use the following root URL to communicate with the endpoints in this category for a GitHub Enterprise Server instance.

```
http(s)://HOSTNAME/api/v3/scim/v2/
```

Endpoints in this category are case-sensitive. For example, the first letter in the `Users` endpoint must be capitalized.

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

Authentication

The SCIM integration on the IdP performs actions on behalf of an enterprise owner for the GitHub Enterprise Server instance. For more information, see "Roles in an enterprise."

To authenticate API requests, the person who configures SCIM on the IdP must use a personal access token (classic) with `admin:enterprise` scope, which the IdP must provide in the request's `Authorization` header. For more information about personal access tokens (classic), see "Managing your personal access tokens".

Note: Enterprise owners must generate and use a personal access token (classic) for authentication of requests to endpoints in this category. Fine-grained personal access token and GitHub app callers are not supported at this time.

Mapping of SAML and SCIM data

The GitHub Enterprise Server instance links each user who authenticates successfully with SAML SSO to a SCIM identity. To link the identities successfully, the SAML IdP and the SCIM integration must use matching SAML `NameID` and SCIM `userName` values for each user.

Note: If the GitHub Enterprise Server uses Azure AD as a SAML IdP, GitHub Enterprise Server will also check the SCIM `externalId` claim and SAML `http://schemas.microsoft.com/identity/claims/objectidentifier` claim to match users first, instead of using `NameID` and `userName`.

Supported SCIM user attributes

`User` endpoints in this category support the following attributes within a request's parameters.

Name	Type	Description
`displayName`	String	Human-readable name for a user.
`name.formatted`	String	The user's full name, including all middle names, titles, and suffixes, formatted for display.
`name.givenName`	String	The first name of the user.
`name.familyName`	String	The last name of the user.
`userName`	String	The username for the user, generated by the IdP. Undergoes normalization before being used.
`emails`	Array	List of the user's emails.
`roles`	Array	List of the user's roles.
`externalId`	String	This identifier is generated by an IdP provider. You can find the `externalId` for a user either on the IdP, or by using the List SCIM provisioned identities endpoint and filtering on other known attributes, such as a user's username or email address on the GitHub Enterprise Server instance.
`id`	String	Identifier generated by the instance's SCIM endpoint.
`active`	Boolean	Indicates whether the identity is active (`true`) or should be suspended (`false`).

List provisioned SCIM groups for an enterprise

✓ Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Lists provisioned SCIM groups in an enterprise.

You can improve query search time by using the `excludedAttributes` query parameter with a value of `members` to exclude members from the response.

Parameters for "List provisioned SCIM groups for an enterprise"

Headers

`accept` string

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

Query parameters

`filter` string

If specified, only results that match the specified filter will be returned. Multiple filters are not supported. Possible filters are `externalId`, `id`, and `displayName`. For example, `?filter="externalId eq '9138790-10932-109120392-12321'"`.

`excludedAttributes` string

Excludes the specified attribute from being returned in the results. Using this parameter can speed up response time.

`startIndex` integer

Used for pagination: the starting index of the first result to return when paginating through values.

Default: `1`

`count` integer

Used for pagination: the number of results to return per page.

Default: `30`

HTTP response status codes for "List provisioned SCIM groups for an enterprise"

Status code	Description
`200`	Success, either groups were found or not found
`400`	Bad request
`401`	Authorization failure
`403`	Permission denied
`429`	Too many requests
`500`	Internal server error

Code samples for "List provisioned SCIM groups for an enterprise"

GET /scim/v2/Groups

cURL JavaScript GitHub CLI

```
 curl -L \ -H "Accept: application/scim+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \
http(s)://HOSTNAME/api/v3/scim/v2/Groups
```

Success, either groups were found or not found

Example response | Response schema

```
Status: 200
```

```
 { "schemas": [ "urn:ietf:params:scim:api:messages:2.0:ListResponse" ], "totalResults": 1, "Resources": [ { "schemas": [
"urn:ietf:params:scim:schemas:core:2.0:Group" ], "externalId": "8aa1a0c0-c4c3-4bc0-b4a5-2ef676900159", "id":
"927fa2c08dcb4a7fae9e", "displayName": "Engineering", "members": [ { "value": "879db59-3bdf-4490-ad68-ab880a2694745", "$+ref":
"https://api.github.localhost/scim/v2/Users/879db59-3bdf-4490-ad68-ab880a2694745", "displayName": "User 1" }, { "value": "0db508eb-
91e2-46e4-809c-30dcbda0c685", "$+ref": "https://api.github.localhost/scim/v2/Users/0db508eb-91e2-46e4-809c-30dcbda0c685",
"displayName": "User 2" } ], "meta": { "resourceType": "Group", "created": "2012-03-27T19:59:26.000Z", "lastModified": "2018-03-
```

Provision a SCIM enterprise group

Works with [GitHub Apps]

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Creates a SCIM group for an enterprise.

If members are included as part of the group provisioning payload, they will be created as external group members. It is up to a provider to store a mapping between the `externalId` and `id` of each user.

Parameters for "Provision a SCIM enterprise group"

Headers

`accept` string

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

Body parameters

`schemas` array of strings Required

The URIs that are used to indicate the namespaces of the SCIM schemas. Supported values are: `urn:ietf:params:scim:schemas:core:2.0:Group`

`externalId` string Required

A unique identifier for the resource as defined by the provisioning client.

`displayName` string Required

A human-readable name for a security group.

`members` array of objects Required

The group members.

▶ Properties of `members`

HTTP response status codes for "Provision a SCIM enterprise group"

Status code	Description
`201`	Group has been created
`400`	Bad request
`401`	Authorization failure
`403`	Permission denied
`409`	Duplicate record detected
`429`	Too many requests
`500`	Internal server error

Code samples for "Provision a SCIM enterprise group"

POST `/scim/v2/Groups`

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" \ http(s)://HOSTNAME/api/v3/scim/v2/Groups \ -d '{"schemas":
["urn:ietf:params:scim:schemas:core:2.0:Group"],"externalId":"8aa1a0c0-c4c3-4bc0-b4a5-2ef676900159","displayName":"Engineering"}'
```

Group has been created

Example response Response schema

```
Status: 201
```

```
{ "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:Group" ], "id": "abcd27f8-a9aa-11ea-8221-f59b2be9cccc", "externalId":
"8aa1a0c0-c4c3-4bc0-b4a5-2ef676900159", "displayName": "Engineering", "members": [ { "value": "879db59-3bdf-4490-ad68-
ab880a2694745", "$+ref": "https://api.github.localhost/scim/v2/Users/879db59-3bdf-4490-ad68-ab880a2694745", "displayName": "User 1"
}, { "value": "0db508eb-91e2-46e4-809c-30dcbda0c685", "$+ref": "https://api.github.localhost/scim/v2/Users/0db508eb-91e2-46e4-809c-
30dcbda0c685", "displayName": "User 2" } ], "meta": { "resourceType": "Group", "created": "2012-03-27T19:59:26.000Z",
"lastModified": "2018-03-27T19:59:26.000Z", "location": "https://api.github.localhost/scim/v2/Groups/927fa2c08dcb4a7fae9e" } }
```

Get SCIM provisioning information for an enterprise group

✓ Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Gets information about a SCIM group.

Parameters for "Get SCIM provisioning information for an enterprise group"

Headers

`accept` string

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

Path parameters

`scim_group_id` string Required

A unique identifier of the SCIM group.

Query parameters

`excludedAttributes` string

Excludes the specified attribute from being returned in the results. Using this parameter can speed up response time.

HTTP response status codes for "Get SCIM provisioning information for an enterprise group"

Status code	Description
`200`	Success, a group was found
`400`	Bad request
`401`	Authorization failure
`403`	Permission denied
`404`	Resource not found
`429`	Too many requests
`500`	Internal server error

Code samples for "Get SCIM provisioning information for an enterprise group"

GET `/scim/v2/Groups/{scim_group_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" \
http(s)://HOSTNAME/api/v3/scim/v2/Groups/SCIM_GROUP_ID
```

Success, a group was found

Example response Response schema

```
Status: 200
```

```
 { "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:Group" ], "id": "abcd27f8-a9aa-11ea-8221-f59b2be9cccc", "externalId":
"8aa1a0c0-c4c3-4bc0-b4a5-2ef676900159", "displayName": "Engineering", "members": [ { "value": "879db59-3bdf-4490-ad68-
ab880a2694745", "$+ref": "https://api.github.localhost/scim/v2/Users/879db59-3bdf-4490-ad68-ab880a2694745", "displayName": "User 1"
}, { "value": "0db508eb-91e2-46e4-809c-30dcbda0c685", "$+ref": "https://api.github.localhost/scim/v2/Users/0db508eb-91e2-46e4-809c-
30dcbda0c685", "displayName": "User 2" } ], "meta": { "resourceType": "Group", "created": "2012-03-27T19:59:26.000Z",
"lastModified": "2018-03-27T19:59:26.000Z", "location": "https://api.github.localhost/scim/v2/Groups/927fa2c08dcb4a7fae9e" } }
```

Set SCIM information for a provisioned enterprise group

Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Replaces an existing provisioned group’s information.

You must provide all the information required for the group as if you were provisioning it for the first time. Any existing group information that you don't provide will be removed, including group membership. If you want to only update a specific attribute, use the Update an attribute for a SCIM enterprise group endpoint instead.

Parameters for "Set SCIM information for a provisioned enterprise group"

Headers

`accept` string

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

Path parameters

`scim_group_id` string Required

A unique identifier of the SCIM group.

Body parameters

`schemas` array of strings Required

The URIs that are used to indicate the namespaces of the SCIM schemas. Supported values are: `urn:ietf:params:scim:schemas:core:2.0:Group`

`externalId` string Required

A unique identifier for the resource as defined by the provisioning client.

`displayName` string Required

A human-readable name for a security group.

`members` array of objects Required

The group members.

▶ Properties of `members`

HTTP response status codes for "Set SCIM information for a provisioned enterprise group"

Status code	Description
`200`	Group was updated
`400`	Bad request
`401`	Authorization failure
`403`	Permission denied

`404`	Resource not found
`409`	Duplicate record detected
`429`	Too many requests
`500`	Internal server error

Code samples for "Set SCIM information for a provisioned enterprise group"

Group ⬍

PUT `/scim/v2/Groups/{scim_group_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" \ http(s)://HOSTNAME/api/v3/scim/v2/Groups/SCIM_GROUP_ID \ -d '{"schemas":
["urn:ietf:params:scim:schemas:core:2.0:Group"],"externalId":"8aa1a0c0-c4c3-4bc0-b4a5-2ef676900159","displayName":"Engineering"}'
```

Group was updated

Example response Response schema

```
Status: 200
```

```
{ "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:Group" ], "id": "abcd27f8-a9aa-11ea-8221-f59b2be9cccc", "externalId":
"8aa1a0c0-c4c3-4bc0-b4a5-2ef676900159", "displayName": "Engineering", "members": [ { "value": "879db59-3bdf-4490-ad68-
ab880a2694745", "$+ref": "https://api.github.localhost/scim/v2/Users/879db59-3bdf-4490-ad68-ab880a2694745", "displayName": "User 1"
}, { "value": "0db508eb-91e2-46e4-809c-30dcbda0c685", "$+ref": "https://api.github.localhost/scim/v2/Users/0db508eb-91e2-46e4-809c-
30dcbda0c685", "displayName": "User 2" } ], "meta": { "resourceType": "Group", "created": "2012-03-27T19:59:26.000Z",
"lastModified": "2018-03-27T19:59:26.000Z", "location": "https://api.github.localhost/scim/v2/Groups/927fa2c08dcb4a7fae9e" } }
```

Update an attribute for a SCIM enterprise group

✓ Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Update a provisioned group’s individual attributes.

To change a group’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. Update can also be used to add group memberships.

Group memberships can be sent one at a time or in batches for faster performance. **Note**: The memberships are referenced through a local user `id`, and the user will need to be created before they are referenced here.

Parameters for "Update an attribute for a SCIM enterprise group"

Headers

`accept` string

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

Path parameters

`scim_group_id` string Required

A unique identifier of the SCIM group.

Body parameters

`Operations` array of objects Required

patch operations list

▶ Properties of `Operations`

`schemas` array of strings Required

undefinedSupported values are: `urn:ietf:params:scim:api:messages:2.0:PatchOp`

HTTP response status codes for "Update an attribute for a SCIM enterprise group"

Status code	Description
`200`	Success, group was updated
`204`	No Content
`400`	Bad request
`401`	Authorization failure
`403`	Permission denied
`404`	Resource not found
`409`	Duplicate record detected
`429`	Too many requests
`500`	Internal server error

Code samples for "Update an attribute for a SCIM enterprise group"

Update Group ⬍

PATCH `/scim/v2/Groups/{scim_group_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" \ http(s)://HOSTNAME/api/v3/scim/v2/Groups/SCIM_GROUP_ID \ -d '{"schemas":["urn:ietf:params:scim:api:messages:2.0:PatchOp"],"Operations":[{"op":"replace","path":"displayName","value":"Employees"}]}'
```

Success, group was updated

Success, group was updated

Example response | Response schema

```
Status: 200
```

```
{ "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:Group" ], "id": "abcd27f8-a9aa-11ea-8221-f59b2be9cccc", "externalId": "8aa1a0c0-c4c3-4bc0-b4a5-2ef676900159", "displayName": "Engineering", "members": [ { "value": "879db59-3bdf-4490-ad68-ab880a2694745", "$+ref": "https://api.github.localhost/scim/v2/Users/879db59-3bdf-4490-ad68-ab880a2694745", "displayName": "User 1" }, { "value": "0db508eb-91e2-46e4-809c-30dcbda0c685", "$+ref": "https://api.github.localhost/scim/v2/Users/0db508eb-91e2-46e4-809c-30dcbda0c685", "displayName": "User 2" } ], "meta": { "resourceType": "Group", "created": "2012-03-27T19:59:26.000Z", "lastModified": "2018-03-27T19:59:26.000Z", "location": "https://api.github.localhost/scim/v2/Groups/927fa2c08dcb4a7fae9e" } }
```

Delete a SCIM group from an enterprise

✓ Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Deletes a SCIM group from an enterprise.

Parameters for "Delete a SCIM group from an enterprise"

Headers

`accept` string

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

Path parameters

`scim_group_id` string Required

A unique identifier of the SCIM group.

HTTP response status codes for "Delete a SCIM group from an enterprise"

Status code	Description
`204`	Group was deleted, no content
`400`	Bad request
`401`	Authorization failure
`403`	Permission denied
`404`	Resource not found
`429`	Too many requests
`500`	Internal server error

Code samples for "Delete a SCIM group from an enterprise"

```
DELETE /scim/v2/Groups/{scim_group_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" \ http(s)://HOSTNAME/api/v3/scim/v2/Groups/SCIM_GROUP_ID
```

Group was deleted, no content

```
Status: 204
```

List SCIM provisioned identities for an enterprise

✓ Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Lists provisioned SCIM enterprise members.

When a user with a SCIM-provisioned external identity is removed from an enterprise through a `patch` with `active` flag set to `false`, the account's metadata is preserved to allow the user to re-join the enterprise in the future. However, the user's account will be suspended and the user will not be able to sign-in. In order to permanently suspend the users account with no ability to re-join the enterprise in the future, use the `delete` request. Users that were not permanently deleted will be visible in the returned results.

Parameters for "List SCIM provisioned identities for an enterprise"

Headers

`accept` string

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

Query parameters

`filter` string

If specified, only results that match the specified filter will be returned. Multiple filters are not supported. Possible filters are `userName`, `externalId`, `id`, and `displayName`. For example, `?filter="externalId eq '9138790-10932-109120392-12321'"`.

`startIndex` integer

Used for pagination: the starting index of the first result to return when paginating through values.

Default: `1`

`count` integer

Used for pagination: the number of results to return per page.

Default: `30`

HTTP response status codes for "List SCIM provisioned identities for an enterprise"

Status code	Description

`200`	Success, either users were found or not found
`400`	Bad request
`401`	Authorization failure
`403`	Permission denied
`429`	Too many requests
`500`	Internal server error

Code samples for "List SCIM provisioned identities for an enterprise"

GET /scim/v2/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" \
http(s)://HOSTNAME/api/v3/scim/v2/Users
```

Success, either users were found or not found

Example response Response schema

```
Status: 200
```

```
 { "schemas": [ "urn:ietf:params:scim:api:messages:2.0:ListResponse" ], "totalResults": 1, "Resources": [ { "schemas": [
"urn:ietf:params:scim:schemas:core:2.0:User" ], "externalId": "E012345", "id": "7fce0092-d52e-4f76-b727-3955bd72c939", "active":
true, "userName": "E012345", "name": { "formatted": "Ms. Mona Lisa Octocat", "familyName": "Octocat", "givenName": "Mona",
"middleName": "Lisa" }, "displayName": "Mona Lisa", "emails": [ { "value": "mlisa@example.com", "type": "work", "primary": true } ],
"roles": [ { "value": "User", "primary": false } ] } ], "startIndex": 1, "itemsPerPage": 20 }
```

Provision a SCIM enterprise user

✓ Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Creates an external identity for a new SCIM enterprise user.

SCIM does not authenticate users, it only provisions them. The authentication of users is done by SAML. However, when SCIM is enabled, all users need to be provisioned through SCIM before a user can sign in through SAML. The matching of a user to a SCIM provisioned user is done when the SAML assertion is consumed. The user will be matched on SAML response `NameID` to SCIM `userName`.

When converting existing enterprise to use SCIM, the user handle (`userName`) from the SCIM payload will be used to match the provisioned user to an already existing user in the enterprise. Since the new identity record is created for newly provisioned users the matching for those records is done using a user's handle. Currently the matching will be performed to all of the users no matter if they were SAML JIT provisioned or created as local users.

Parameters for "Provision a SCIM enterprise user"

Headers

`accept` string

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

Body parameters

`schemas` array of strings Required

The URIs that are used to indicate the namespaces of the SCIM schemas. Supported values are: `urn:ietf:params:scim:schemas:core:2.0:User`

`externalId` string Required

A unique identifier for the resource as defined by the provisioning client.

`active` boolean Required

Whether the user active in the IdP.

`userName` string Required

The username for the user.

`name` object

▶ Properties of `name`

`displayName` string Required

A human-readable name for the user.

`emails` array of objects Required

The emails for the user.

▶ Properties of `emails`

`roles` array of objects

The roles assigned to the user.

▶ Properties of `roles`

HTTP response status codes for "Provision a SCIM enterprise user"

Status code	Description
`201`	User has been created
`400`	Bad request
`401`	Authorization failure
`403`	Permission denied
`409`	Duplicate record detected
`429`	Too many requests

`500`	Internal server error

Code samples for "Provision a SCIM enterprise user"

User ⬍

POST /scim/v2/Users

cURL JavaScript

```
curl -L \ -X POST \ -H "Accept: application/scim+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-28" \ http(s)://HOSTNAME/api/v3/scim/v2/Users \ -d '{"schemas":["urn:ietf:params:scim:schemas:core:2.0:User"],"externalId":"E012345","active":true,"userName":"E012345","name":{"formatted":"Ms. Mona Lisa Octocat","familyName":"Octocat","givenName":"Mona","middleName":"Lisa"},"displayName":"Mona Lisa","emails":[{"value":"mlisa@example.com","type":"work","primary":true}],"roles":[{"value":"User","primary":false}]}'
```

User has been created

Example response Response schema

```
Status: 201
```

```
{ "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:User" ], "id": "7fce0092-d52e-4f76-b727-3955bd72c939", "externalId": "E012345", "active": true, "userName": "E012345", "name": { "formatted": "Ms. Mona Lisa Octocat", "familyName": "Octocat", "givenName": "Mona", "middleName": "Lisa" }, "displayName": "Mona Lisa", "emails": [ { "value": "mlisa@example.com", "type": "work", "primary": true } ], "roles": [ { "value": "User", "primary": false } ], "meta": { "resourceType": "User", "created": "2012-03-27T19:59:26.000Z", "lastModified": "2018-03-27T19:59:26.000Z", "location": "https://api.github.localhost/scim/v2/Users/7fce0092-d52e-4f76-b727-3955bd72c939" } }
```

Get SCIM provisioning information for an enterprise user

✓ Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Gets information about a SCIM user.

Parameters for "Get SCIM provisioning information for an enterprise user"

Headers

`accept` string

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

Path parameters

`scim_user_id` string Required

The unique identifier of the SCIM user.

HTTP response status codes for "Get SCIM provisioning information for an enterprise user"

Status code	Description
`200`	Success, a user was found
`400`	Bad request
`401`	Authorization failure
`403`	Permission denied
`404`	Resource not found
`429`	Too many requests
`500`	Internal server error

Code samples for "Get SCIM provisioning information for an enterprise user"

GET `/scim/v2/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" \
http(s)://HOSTNAME/api/v3/scim/v2/Users/SCIM_USER_ID
```

Success, a user was found

Example response Response schema

```
Status: 200
```

```
 { "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:User" ], "id": "7fce0092-d52e-4f76-b727-3955bd72c939", "externalId":
"E012345", "active": true, "userName": "E012345", "name": { "formatted": "Ms. Mona Lisa Octocat", "familyName": "Octocat",
"givenName": "Mona", "middleName": "Lisa" }, "displayName": "Mona Lisa", "emails": [ { "value": "mlisa@example.com", "type":
"work", "primary": true } ], "roles": [ { "value": "User", "primary": false } ], "meta": { "resourceType": "User", "created":
"2012-03-27T19:59:26.000Z", "lastModified": "2018-03-27T19:59:26.000Z", "location":
"https://api.github.localhost/scim/v2/Users/7fce0092-d52e-4f76-b727-3955bd72c939" } }
```

Set SCIM information for a provisioned enterprise user

✓ Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

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.

Warning: Setting `active: false` will suspend a user and obfuscate the user handle and user email. Since the implementation is a generic SCIM implementation and does not differentiate yet between different IdP providers, for Okta, the user GDPR data will not be purged and the credentials will not be removed.

Parameters for "Set SCIM information for a provisioned enterprise user"

Headers

`accept` string

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

Path parameters

`scim_user_id` string Required

The unique identifier of the SCIM user.

Body parameters

`schemas` array of strings Required

The URIs that are used to indicate the namespaces of the SCIM schemas. Supported values are: `urn:ietf:params:scim:schemas:core:2.0:User`

`externalId` string Required

A unique identifier for the resource as defined by the provisioning client.

`active` boolean Required

Whether the user active in the IdP.

`userName` string Required

The username for the user.

`name` object

▶ Properties of `name`

`displayName` string Required

A human-readable name for the user.

`emails` array of objects Required

The emails for the user.

▶ Properties of `emails`

`roles` array of objects

The roles assigned to the user.

▶ Properties of `roles`

HTTP response status codes for "Set SCIM information for a provisioned enterprise user"

Status code	Description
`200`	User was updated
`400`	Bad request

`401`	Authorization failure
`403`	Permission denied
`404`	Resource not found
`409`	Duplicate record detected
`429`	Too many requests
`500`	Internal server error

Code samples for "Set SCIM information for a provisioned enterprise user"

PUT `/scim/v2/Users/{scim_user_id}`

cURL JavaScript

```
 curl -L \ -X PUT \ -H "Accept: application/scim+json" \ -H "Authorization: Bearer <YOUR-TOKEN>" \ -H "X-GitHub-Api-Version: 2022-11-
28" \ http(s)://HOSTNAME/api/v3/scim/v2/Users/SCIM_USER_ID \ -d '{"schemas":
["urn:ietf:params:scim:schemas:core:2.0:User"],"externalId":"E012345","active":true,"userName":"E012345","name":{"formatted":"Ms.
Mona Lisa Octocat","familyName":"Octocat","givenName":"Mona","middleName":"Lisa"},"displayName":"Mona Lisa","emails":
[{"value":"mlisa@example.com","type":"work","primary":true}],"roles":[{"value":"User","primary":false}]}'
```

User was updated

Example response Response schema

```
Status: 200
```

```
 { "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:User" ], "id": "7fce0092-d52e-4f76-b727-3955bd72c939", "externalId":
"E012345", "active": true, "userName": "E012345", "name": { "formatted": "Ms. Mona Lisa Octocat", "familyName": "Octocat",
"givenName": "Mona", "middleName": "Lisa" }, "displayName": "Mona Lisa", "emails": [ { "value": "mlisa@example.com", "type":
"work", "primary": true } ], "roles": [ { "value": "User", "primary": false } ], "meta": { "resourceType": "User", "created":
"2012-03-27T19:59:26.000Z", "lastModified": "2018-03-27T19:59:26.000Z", "location":
"https://api.github.localhost/scim/v2/Users/7fce0092-d52e-4f76-b727-3955bd72c939" } }
```

Update an attribute for a SCIM enterprise user

✓ Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Update 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: Setting `active: false` will suspend a user and obfuscate the user handle and user email. Since the implementation is a generic SCIM implementation and does not differentiate yet between different IdP providers, for Okta, the user GDPR data will not be purged and the credentials will not be removed.

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

Parameters for "Update an attribute for a SCIM enterprise user"

Headers

`accept` string

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

Path parameters

`scim_user_id` string Required

The unique identifier of the SCIM user.

Body parameters

`Operations` array of objects Required

patch operations list

▶ Properties of `Operations`

`schemas` array of strings Required

undefinedSupported values are: `urn:ietf:params:scim:api:messages:2.0:PatchOp`

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

Status code	Description
`200`	Success, user was updated
`400`	Bad request
`401`	Authorization failure
`403`	Permission denied
`404`	Resource not found
`409`	Duplicate record detected
`429`	Too many requests
`500`	Internal server error

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

Multi Valued Property ⬍

PATCH /scim/v2/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" \ http(s)://HOSTNAME/api/v3/scim/v2/Users/SCIM_USER_ID \ -d '{"schemas":
["urn:ietf:params:scim:api:messages:2.0:PatchOp"],"Operations":[{"op":"replace","path":"emails[type eq
'\''work'\''].value","value":"updatedEmail@microsoft.com"},{"op":"replace","path":"name.familyName","value":"updatedFamilyName"}]}'
```

Success, user was updated

Example response Response schema

```
Status: 200
```

```
{ "schemas": [ "urn:ietf:params:scim:schemas:core:2.0:User" ], "id": "7fce0092-d52e-4f76-b727-3955bd72c939", "externalId":
"E012345", "active": true, "userName": "E012345", "name": { "formatted": "Ms. Mona Lisa Octocat", "familyName": "Octocat",
"givenName": "Mona", "middleName": "Lisa" }, "displayName": "Mona Lisa", "emails": [ { "value": "mlisa@example.com", "type":
"work", "primary": true } ], "roles": [ { "value": "User", "primary": false } ], "meta": { "resourceType": "User", "created":
"2012-03-27T19:59:26.000Z", "lastModified": "2018-03-27T19:59:26.000Z", "location":
"https://api.github.localhost/scim/v2/Users/7fce0092-d52e-4f76-b727-3955bd72c939" } }
```

Delete a SCIM user from an enterprise

✓ Works with GitHub Apps

Note: The SCIM API endpoints for enterprise accounts are currently in *private* beta and are subject to change.

Permanently suspends a SCIM user from an enterprise, removes all data for the user, obfuscates the login, email, and display name of the user, removes all external-identity SCIM attributes, and deletes the emails, avatar, PATs, SSH keys, OAuth authorizations credentials, GPG keys, and SAML mappings for the user. You will not be able to undo this action.

Parameters for "Delete a SCIM user from an enterprise"

Headers

`accept` string

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

Path parameters

`scim_user_id` string Required

The unique identifier of the SCIM user.

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

Status code	Description
`204`	User was deleted, no content

`400`	Bad request
`401`	Authorization failure
`403`	Permission denied
`404`	Resource not found
`429`	Too many requests
`500`	Internal server error

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

DELETE `/scim/v2/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" \ http(s)://HOSTNAME/api/v3/scim/v2/Users/SCIM_USER_ID
```

User was deleted, no content

```
Status: 204
```

Legal

Terms Privacy Status Pricing Expert services Blog