AWS SSO SCIM Implementation

Developer Guide

AWS SSO SCIM Implementation: Developer Guide

Table of Contents

What is the AWS Single Sign-On SCIM implementation?

This reference guide helps software developers build custom integrations to provision (synchronize) users and groups into AWS Single Sign-On (AWS SSO) using the System for Cross-domain Identity Management (SCIM) v2.0 protocol. This guide will also be useful to IT administrators who need to understand or debug an existing SCIM implementation.

The AWS SSO SCIM implementation is based on SCIM RFCs 7642 (https://tools.ietf.org/html/rfc7642), 7643 (https://tools.ietf.org/html/rfc7643), and 7644 (https://tools.ietf.org/html/rfc7644), and the interoperability requirements laid out in the March 2020 draft of the FastFed Basic SCIM Profile 1.0 (https://openid.net/specs/fastfed-scim-1_0-02.html#rfc.section.4). Any differences between these documents and the current implementation in AWS SSO are described in the Supported API operations (p. 2) section of this guide.

The following sections contain examples of API requests and responses currently supported in the AWS SSO SCIM implementation, along with important notes and constraints to consider in your design.

Before you begin, we recommend that you first review Considerations for Using Automatic Provisioning in the *AWS Single Sign-On User Guide*. That topic instructs you how to use SCIM to enable automatic provisioning in AWS SSO. You will need to follow those instructions to retrieve your SCIM endpoint and access token.

Supported API operations

The following API operations are supported by the AWS SSO SCIM implementation:

Topics

CreateUser

You can create new users from a POST request using the AWS SSO SCIM implementation **/Users** endpoint. See the examples below.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- `ims`, `photos`, `x509Certificates`, `entitlements`, and `password` attributes
- `displayName` subattribute for manager
- `display` subattribute for `emails`, `addresses`, and `phoneNumbers`

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- The `givenName`, `familyName`, `userName`, and `displayName` fields are required.
- The `addresses` field can contain letters, accented characters, symbols, numbers, punctuation, space (normal and nonbreaking).
- We do not support multiple values in multi-value attributes (such as `emails`, `addresses`, `phoneNumbers`). Only single values are permitted.
- The `emails` attribute value must be marked as primary.
- The `groups` field cannot be specified with the `createUser` request.
- The `userName` field can contain letters, accented characters, symbols, numbers, punctuation.

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	Operation is not permitted based on the supplied authorization.
`ThrottlingException`	Too many requests were made that exceed the limits.
`ValidationException`	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.
`ConflictException`	User already exists.
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
POST https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Users
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>

{
  "id": "9067729b3d-94f1e0b3-c394-48d5-8ab1-2c122a167074",
  "externalId": "701984",
  "userName": "bjensen",
  "name": {
    "formatted": "Ms. Barbara J Jensen, III",
    "familyName": "Jensen",
    "givenName": "Barbara",
    "middleName": "Jane",
    "honorificPrefix": "Ms.",
    "honorificSuffix": "III"
  },
  "displayName": "Babs Jensen",
  "nickName": "Babs",
  "profileUrl": "https://login.example.com/bjensen",
  "emails": [
    {
      "value": "bjensen@example.com",
      "type": "work",
      "primary": true
    }
  ],
  "addresses": [
    {
      "type": "work",
      "streetAddress": "100 Universal City Plaza",
```

```
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "91608",
      "country": "USA",
      "formatted": "100 Universal City Plaza Hollywood, CA 91608 USA",
      "primary": true
    }
  ],
  "phoneNumbers": [
    {
      "value": "555-555-5555",
      "type": "work"
    }
  ],
  "userType": "Employee",
  "title": "Tour Guide",
  "preferredLanguage": "en-US",
  "locale": "en-US",
  "timezone": "America/Los_Angeles",
  "active":true,
  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "employeeNumber": "701984",
    "costCenter": "4130",
    "organization": "Universal Studios",
    "division": "Theme Park",
    "department": "Tour Operations",
    "manager": {
      "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734",
      "$ref": "../Users/9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
    }
  }
}
```

Example Response

```
HTTP/1.1 201
Date: Tue, 31 Mar 2020 02:36:15 GMT
Content-Type: application/json
x-amzn-RequestId: abbf9e53-9ecc-46d2-8efe-104a66ff128f
{
    "id": "9067729b3d-94f1e0b3-c394-48d5-8ab1-2c122a167074",
    "externalId": "701984",
    "meta": {
        "resourceType": "User",
        "created": "2020-03-31T02:36:15Z",
        "lastModified": "2020-03-31T02:36:15Z"
    },
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:User",
        "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
    ],
    "userName": "bjensen",
    "name": {
        "formatted": "Ms. Barbara J Jensen, III",
        "familyName": "Jensen",
        "givenName": "Barbara",
        "middleName": "Jane",
        "honorificPrefix": "Ms.",
        "honorificSuffix": "III"
    },
    "displayName": "Babs Jensen",
    "nickName": "Babs",
    "title": "Tour Guide",
    "userType": "Employee",
```

```
    "preferredLanguage": "en-US",
    "locale": "en-US",
    "timezone": "America/Los_Angeles",
    "active": true,
    "emails": [
        {
            "value": "bjensen@example.com",
            "type": "work",
            "primary": true
        }
    ],
    "addresses": [
        {
            "formatted": "100 Universal City Plaza Hollywood, CA 91608 USA",
            "streetAddress": "100 Universal City Plaza",
            "locality": "Hollywood",
            "region": "CA",
            "postalCode": "91608",
            "country": "USA",
            "type": "work",
            "primary": true
        }
    ],
    "phoneNumbers": [
        {
            "value": "555-555-5555",
            "type": "work"
        }
    ],
    "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
        "employeeNumber": "701984",
        "costCenter": "4130",
        "organization": "Universal Studios",
        "division": "Theme Park",
        "department": "Tour Operations",
        "manager": {
            "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
        }
    }
}
```


GetUser

Existing users can be retrieved by making a `GET` request to the AWS SSO SCIM implementation `/Users` endpoint with a user ID.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- None

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	The operation is not permitted based on the supplied authorization.
`ThrottlingException`	Too many requests were made that exceed the limits.
`ValidationException`	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.
`ResourceNotFoundException`	Specified user or endpoint does not exist.
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Users/9067729b3d-
ee533c18-538a-4cd3-a572-63fb863ed734
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 200
Date: Tue, 31 Mar 2020 02:36:15 GMT
Content-Type: application/json
x-amzn-RequestId: abbf9e53-9ecc-46d2-8efe-104a66ff128f

{
    "id": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734",
    "externalId": "1",
    "meta": {
        "resourceType": "User",
        "created": "2020-03-30T16:55:15Z",
        "lastModified": "2020-03-30T16:55:15Z"
    },
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:User"
    ],
    "userName": "johndoe",
    "name": {
        "familyName": "Doe",
        "givenName": "John"
    },
```

```
    "displayName": "John Doe",
    "active": false,
    "emails": [
        {
            "value": "johndoe@example.com",
            "type": "work",
            "primary": true
        }
    ]
}
```

ListUsers

This endpoint provides the ability to perform filter queries on an existing list of users through a `GET` request to `/Users` by inserting additional filters. Only a maximum of 50 results can be returned. See the **Constraints** section for more information.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- `startIndex`, `attributes`, and `excludedAttributes` (despite being listed in the SCIM protocol)

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- At this time, the `ListUsers` API is only capable of returning up to 50 results.
- Supported filter combinations: (`userName`), (`externalId`), (`id` and `manager`), (`manager` and `id`). Note that the use of `id` as an individual filter, though valid, should be avoided as a `getUser` endpoint is already available.
- Supported comparison operator in filters: `eq`
- Filter must be specified as follows: `<filterAttribute> eq "<filterValue>"`

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	Operation is not permitted based on the supplied authorization.
`ThrottlingException`	Too many requests were made that exceed the limits.
`ValidationException`	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.

Error	Condition
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Users
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 200
Date: Thu, 23 Jul 2020 00:15:28 GMT
Content-Type: application/json
x-amzn-RequestId: 88204ccc-30cd-4010-b3ac-b4a12cc31e8b

{
    "totalResults": 5,
    "itemsPerPage": 5,
    "startIndex": 1,
    "schemas": [
        "urn:ietf:params:scim:api:messages:2.0:ListResponse"
    ],
    "Resources": [
        {
            "id": "90677c608a-7afcdc23-0bd4-4fb7-b2ff-10ccffdff447",
            "externalId": "702135",
            "meta": {
                "resourceType": "User",
                "created": "2020-07-22T22:32:58Z",
                "lastModified": "2020-07-22T22:32:58Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:User",
                "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
            ],
            "userName": "mjack",
            "name": {
                "familyName": "Mark",
                "givenName": "Jackson",
                "honorificPrefix": "Mr.",
                "honorificSuffix": "I"
            },
            "displayName": "mjack",
            "nickName": "Mark",
            "active": false,
            "emails": [
                {
                    "value": "mjack@example.com",
                    "type": "work",
                    "primary": true
                }
            ],
            "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
                "manager": {
                    "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
```

```
            }
        }
    },
    {
        "id": "90677c608a-787142a0-3f27-4cd3-afb6-8aed7ce87094",
        "externalId": "705167",
        "meta": {
            "resourceType": "User",
            "created": "2020-07-22T22:34:55Z",
            "lastModified": "2020-07-22T22:34:55Z"
        },
        "schemas": [
            "urn:ietf:params:scim:schemas:core:2.0:User",
            "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
        ],
        "userName": "druss",
        "name": {
            "familyName": "Daniel",
            "givenName": "Russell",
            "honorificPrefix": "Mr.",
            "honorificSuffix": "I"
        },
        "displayName": "danrussell",
        "nickName": "Dan",
        "active": false,
        "emails": [
            {
                "value": "druss@example.com",
                "type": "work",
                "primary": true
            }
        ],
        "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
            "manager": {
                "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
            }
        }
    },
    {
        "id": "90677c608a-229f7eb1-c07d-4c21-a5fd-769bf2e8c5c9",
        "externalId": "2",
        "meta": {
            "resourceType": "User",
            "created": "2020-07-22T23:51:45Z",
            "lastModified": "2020-07-22T23:51:45Z"
        },
        "schemas": [
            "urn:ietf:params:scim:schemas:core:2.0:User"
        ],
        "userName": "tzhang",
        "name": {
            "familyName": "Terry",
            "givenName": "Zhang"
        },
        "displayName": "Terry Zhang",
        "active": false,
        "emails": [
            {
                "value": "tzhang@example.com",
                "type": "work",
                "primary": true
            }
        ]
    },
    {
        "id": "90677c608a-685d5bf3-efab-48c8-b3b1-648fc5c5d980",
```

```
            "externalId": "701985",
            "meta": {
                "resourceType": "User",
                "created": "2020-07-22T22:17:47Z",
                "lastModified": "2020-07-22T22:17:47Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:User",
                "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
            ],
            "userName": "jdoe",
            "name": {
                "familyName": "John",
                "givenName": "Doe",
                "honorificPrefix": "Mr.",
                "honorificSuffix": "III"
            },
            "displayName": "jdoe",
            "nickName": "Johnny",
            "active": false,
            "emails": [
                {
                    "value": "johndoe@example.com",
                    "type": "work",
                    "primary": true
                }
            ],
            "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
                "manager": {
                    "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
                }
            }
        },
        {
            "id": "90677c608a-9683e752-a6fd-4935-b6b8-3fe26a202f21",
            "externalId": "702138",
            "meta": {
                "resourceType": "User",
                "created": "2020-07-22T22:38:11Z",
                "lastModified": "2020-07-22T22:38:11Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:User",
                "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
            ],
            "userName": "hmack",
            "name": {
                "familyName": "Henry",
                "givenName": "Mackenzie",
                "honorificPrefix": "Mr.",
                "honorificSuffix": "I"
            },
            "displayName": "hmack",
            "nickName": "Henry",
            "active": false,
            "emails": [
                {
                    "value": "hmack@example.com",
                    "type": "work",
                    "primary": true
                }
            ],
            "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
                "manager": {
                    "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863jd956"
                }
```

```
            }
        }
    ]
}
```

Filter examples

The following four different filter combinations are supported.

- `externalId`
- `userName`
- `id` and `manager`
- `manager` and `id`

The filters can be applied in the formats as shown.

Single filter

```
filter=<filterAttribute> eq "<filterValue>"
```

Two filters

```
filter=<filterAttribute1> eq "<filterValue1>" and <filterAttribute2> eq "<filterValue2>"
```

externalId

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Users?filter=externalId eq
 "705167"
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 200
Date: Wed, 22 Jul 2020 22:57:01 GMT
Content-Type: application/json
x-amzn-RequestId: c482800a-f6ba-4979-91d0-72d3a7b496cb

{
    "totalResults": 1,
    "itemsPerPage": 1,
    "startIndex": 1,
    "schemas": [
        "urn:ietf:params:scim:api:messages:2.0:ListResponse"
    ],
    "Resources": [
        {
            "id": "90677c608a-787142a0-3f27-4cd3-afb6-8aed7ce87094",
            "externalId": "705167",
            "meta": {
                "resourceType": "User",
                "created": "2020-07-22T22:34:55Z",
                "lastModified": "2020-07-22T22:34:55Z"
            },
            "schemas": [
```

```
                "urn:ietf:params:scim:schemas:core:2.0:User",
                "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
            ],
            "userName": "druss",
            "name": {
                "familyName": "Daniel",
                "givenName": "Russell",
                "honorificPrefix": "Mr.",
                "honorificSuffix": "I"
            },
            "displayName": "danrussell",
            "nickName": "Dan",
            "active": false,
            "emails": [
                {
                    "value": "druss@example.com",
                    "type": "work",
                    "primary": true
                }
            ],
            "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
                "manager": {
                    "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
                }
            }
        }
    ]
}
```

userName

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Users?filter=userName eq
 "jdoe"
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 200
Date: Wed, 22 Jul 2020 22:53:33 GMT
Content-Type: application/json
x-amzn-RequestId: a8764ca2-899f-4362-871d-3f255671ca1f

{
    "totalResults": 1,
    "itemsPerPage": 1,
    "startIndex": 1,
    "schemas": [
        "urn:ietf:params:scim:api:messages:2.0:ListResponse"
    ],
    "Resources": [
        {
            "id": "90677c608a-685d5bf3-efab-48c8-b3b1-648fc5c5d980",
            "externalId": "701985",
            "meta": {
                "resourceType": "User",
                "created": "2020-07-22T22:17:47Z",
                "lastModified": "2020-07-22T22:17:47Z"
            },
            "schemas": [
```

```
                "urn:ietf:params:scim:schemas:core:2.0:User",
                "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
            ],
            "userName": "jdoe",
            "name": {
                "familyName": "John",
                "givenName": "Doe",
                "honorificPrefix": "Mr.",
                "honorificSuffix": "III"
            },
            "displayName": "jdoe",
            "nickName": "Johnny",
            "active": false,
            "emails": [
                {
                    "value": "johndoe@example.com",
                    "type": "work",
                    "primary": true
                }
            ],
            "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
                "manager": {
                    "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
                }
            }
        }
    ]
}
```

User id and manager

Both `id` and `manager` can be used together, and their order can be interchanged.

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Users?filter=id eq
 "90677c608a-7afcdc23-0bd4-4fb7-b2ff-10ccffdff447" and manager eq "9067729b3d-
ee533c18-538a-4cd3-a572-63fb863ed734"
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 200
Date: Wed, 22 Jul 2020 22:42:29 GMT
Content-Type: application/json
x-amzn-RequestId: 23178777-466c-44fb-b5b4-7efc12a766aa
{
    "totalResults": 1,
    "itemsPerPage": 1,
    "startIndex": 1,
    "schemas": [
        "urn:ietf:params:scim:api:messages:2.0:ListResponse"
    ],
    "Resources": [
        {
            "id": "90677c608a-7afcdc23-0bd4-4fb7-b2ff-10ccffdff447",
            "externalId": "702135",
            "meta": {
                "resourceType": "User",
                "created": "2020-07-22T22:32:58Z",
                "lastModified": "2020-07-22T22:32:58Z"
```

```
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:User",
                "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
            ],
            "userName": "mjack",
            "name": {
                "familyName": "Mark",
                "givenName": "Jackson",
                "honorificPrefix": "Mr.",
                "honorificSuffix": "I"
            },
            "displayName": "mjack",
            "nickName": "Mark",
            "active": false,
            "emails": [
                {
                    "value": "mjack@example.com",
                    "type": "work",
                    "primary": true
                }
            ],
            "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
                "manager": {
                    "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
                }
            }
        }
    ]
}
```

DeleteUser

A user can be deleted by making a `DELETE` request to the `/Users` endpoint with an existing user ID.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- None

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.

Error	Condition
`AccessDeniedException`	Operation is not permitted based on the supplied authorization.
`ThrottlingException`	Too many requests were made that exceed the limits.
`ResourceNotFoundException`	Specified user does not exist.
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
DELETE https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Users/9067729b3d-
ee533c18-538a-4cd3-a572-63fb863ed734
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 204
Date: Tue, 31 Mar 2020 02:36:15 GMT
Content-Type: application/json
x-amzn-RequestId: abbf9e53-9ecc-46d2-8efe-104a66ff128f
```

PutUser

An existing user can be overwritten by making a `PUT` request to the `/Users` endpoint with the user ID.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- `ims`, `photos`, `x509Certificates`, `entitlements`, and `password` attributes
- `displayName` subattribute for manager
- `display` subattribute for `emails`, `addresses`, and `phoneNumbers`

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- `givenName`, `familyName`, `userName`, and `displayName` are required.
- The `addresses` field can contain letters, accented characters, symbols, numbers, punctuation, space (normal and nonbreaking).
- AWS SSO does not support multiple values in multi-value attributes (such as `emails`, `addresses`, `phoneNumbers`). Only single values are permitted.

- The `emails` attribute value must be marked as primary.
- `groups` cannot be specified with `createUser` request.
- The `userName` field can contain letters, accented characters, symbols, numbers, punctuation, space (normal and nonbreaking).

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	Operation is not permitted based on the supplied authorization.
`ThrottlingException`	Too many requests were made that exceed the limits.
`ValidationException`	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.
`ConflictException`	User already exists.
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
PUT https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Users/9067729b3d-94f1e0b3-
c394-48d5-8ab1-2c122a167074
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>

{
  "id": "9067729b3d-94f1e0b3-c394-48d5-8ab1-2c122a167074",
  "externalId": "701984",
  "userName": "bjensen",
  "name": {
    "formatted": "Ms. Barbara J Jensen, III",
    "familyName": "Jensen",
    "givenName": "Barbara",
    "middleName": "Jane",
    "honorificPrefix": "Ms.",
    "honorificSuffix": "III"
  },
  "displayName": "Babs Jensen",
  "nickName": "BabJ",
  "profileUrl": "https://login.example.com/bjensen",
  "emails": [
    {
      "value": "bjensen@example.com",
```

```
      "type": "work",
      "primary": true
    }
  ],
  "addresses": [
    {
      "type": "work",
      "streetAddress": "100 Universal City Plaza",
      "locality": "Hollywood",
      "region": "CA",
      "postalCode": "91608",
      "country": "USA",
      "formatted": "100 Universal City Plaza Hollywood, CA 91608 USA",
      "primary": true
    }
  ],
  "phoneNumbers": [
    {
      "value": "555-555-5555",
      "type": "work"
    }
  ],
  "userType": "Employee",
  "title": "Tour Guide",
  "preferredLanguage": "en-US",
  "locale": "en-US",
  "timezone": "America/Los_Angeles",
  "active":true,
  "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
    "employeeNumber": "701984",
    "costCenter": "4130",
    "organization": "Universal Studios",
    "division": "Theme Park",
    "department": "Tour Operations",
    "manager": {
      "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734",
      "$ref": "../Users/9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
    }
  }
}
```

Example Response

```
HTTP/1.1 201
Date: Tue, 31 Mar 2020 02:36:15 GMT
Content-Type: application/json
x-amzn-RequestId: abbf9e53-9ecc-46d2-8efe-104a66ff128f

{
    "id": "9067729b3d-94f1e0b3-c394-48d5-8ab1-2c122a167074",
    "externalId": "701984",
    "meta": {
        "resourceType": "User",
        "created": "2020-03-31T02:36:15Z",
        "lastModified": "2020-03-31T02:36:15Z"
    },
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:User",
        "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
    ],
    "userName": "bjensen",
    "name": {
        "formatted": "Ms. Barbara J Jensen, III",
        "familyName": "Jensen",
```

```
        "givenName": "Barbara",
        "middleName": "Jane",
        "honorificPrefix": "Ms.",
        "honorificSuffix": "III"
    },
    "displayName": "Babs Jensen",
    "nickName": "BabJ",
    "title": "Tour Guide",
    "userType": "Employee",
    "preferredLanguage": "en-US",
    "locale": "en-US",
    "timezone": "America/Los_Angeles",
    "active": true,
    "emails": [
        {
            "value": "bjensen@example.com",
            "type": "work",
            "primary": true
        }
    ],
    "addresses": [
        {
            "formatted": "100 Universal City Plaza Hollywood, CA 91608 USA",
            "streetAddress": "100 Universal City Plaza",
            "locality": "Hollywood",
            "region": "CA",
            "postalCode": "91608",
            "country": "USA",
            "type": "work",
            "primary": true
        }
    ],
    "phoneNumbers": [
        {
            "value": "555-555-5555",
            "type": "work"
        }
    ],
    "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
        "employeeNumber": "701984",
        "costCenter": "4130",
        "organization": "Universal Studios",
        "division": "Theme Park",
        "department": "Tour Operations",
        "manager": {
            "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
        }
    }
}
```

PatchUser

The `/Users` endpoint allows a `PATCH` request to be made for partial changes to an existing user. In the body of the request, the target attribute and its new value must be specified as shown in the `Examples` section.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- Multiple `PATCH` operations on `userName` or `active` attribute

- `ims`, `photos`, `x509Certificates`, `entitlements`, and `password` field
- `displayName` subattribute for manager
- `display` subattribute for `emails`, `addresses`, and `phoneNumbers`

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- The supported operations are `add`, `replace`, and `remove`.
- The operation must be specified.
- The path is required for a `remove` operation.
- A value is required for `add` and `replace`.
- Modification is only allowed for the `userName`, `active`, `externalId`, `displayName`, `nickName`, `profileUrl`, `title`, `userType`, `preferredLanguage`, `locale`, `timezone`, `name`, `enterprise`, `emails`, `addresses`, and `phoneNumbers` attributes.
- Only the `eq` operator is supported in filters.
- The `remove` patch operation is not supported for `userName` or active attributes.
- We do not support having multi-valued attributes (such as `emails`, `addresses`, `phoneNumbers`. Only one value is permitted for each of those attributes.

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	Operation is not permitted based on the supplied authorization.
`ThrottlingException`	Too many requests were made that exceed the limits.
`ValidationException`	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.
`ConflictException`	User already exists.
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
PATCH https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Users/9067729b3d-94f1e0b3-c394-48d5-8ab1-2c122a167074
```

```
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>

{
    "schemas": [
        "urn:ietf:params:scim:api:messages:2.0:PatchOp"
    ],
    "Operations": [
        {
            "op": "replace",
            "path": "active",
            "value": "false"
        }
    ]
}
```

Example Response

```
HTTP/1.1 200
Date: Tue, 31 Mar 2020 02:36:15 GMT
Content-Type: application/json
x-amzn-RequestId: abbf9e53-9ecc-46d2-8efe-104a66ff128f
{
    "id": "9067729b3d-94f1e0b3-c394-48d5-8ab1-2c122a167074",
    "externalId": "701984",
    "meta": {
        "resourceType": "User",
        "created": "2020-03-31T02:36:15Z",
        "lastModified": "2020-04-03T06:02:47Z"
    },
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:User",
        "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
    ],
    "userName": "bjensen",
    "name": {
        "formatted": "Ms. Barbara J Jensen, III",
        "familyName": "Jensen",
        "givenName": "Barbara",
        "middleName": "Jane",
        "honorificPrefix": "Ms.",
        "honorificSuffix": "III"
    },
    "displayName": "Babs Jensen",
    "nickName": "Bas",
    "title": "Tour Guide",
    "userType": "Employee",
    "preferredLanguage": "en-US",
    "locale": "en-US",
    "timezone": "America/Los_Angeles",
    "active": false,
    "emails": [
        {
            "value": "bjensen@example.com",
            "type": "work",
            "primary": true
        }
    ],
    "addresses": [
        {
            "formatted": "100 Universal City Plaza Hollywood, CA 91608 USA",
            "streetAddress": "100 Universal City Plaza",
            "locality": "Hollywood",
            "region": "CA",
```

```
            "postalCode": "91608",
            "country": "USA",
            "type": "work",
            "primary": true
        }
    ],
    "phoneNumbers": [
        {
            "value": "555-555-5555",
            "type": "work"
        }
    ],
    "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User": {
        "employeeNumber": "701984",
        "costCenter": "4130",
        "organization": "Universal Studios",
        "division": "Theme Park",
        "department": "Tour Operations",
        "manager": {
            "value": "9067729b3d-ee533c18-538a-4cd3-a572-63fb863ed734"
        }
    }
}
```

CreateGroup

Groups can be created through a `POST` request to the `/Groups` endpoint with the body containing the information of the group.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- `displayName` is required.
- A maximum of 100 members can be added in a single request.

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	Operation is not permitted based on the supplied authorization.

Error	Condition
`ThrottlingException`	Too many requests were made that exceed the limits.
`ValidationException`	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.
`ConflictException`	Group already exists.
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
POST https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>

{
    "displayName": "Group Bar",
    "members": [
        {
            "value": "9067729b3d-94f1e0b3-c394-48d5-8ab1-2c122a167074",
            "$ref": "../Users/9067729b3d-94f1e0b3-c394-48d5-8ab1-2c122a167074",
            "type": "User"
        }
    ]
}
```

Example Response

```
HTTP/1.1 201
Date: Mon, 06 Apr 2020 16:48:19 GMT
Content-Type: application/json
x-amzn-RequestId: abbf9e53-9ecc-46d2-8efe-104a66ff128f

{
    "id": "9067729b3d-a2cfc8a5-f4ab-4443-9d7d-b32a9013c554",
    "meta": {
        "resourceType": "Group",
        "created": "2020-04-06T16:48:19Z",
        "lastModified": "2020-04-06T16:48:19Z"
    },
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:Group"
    ],
    "displayName": "Group Bar"
}
```

GetGroup

Information about an existing group can be retrieved by making a request to the `/Groups` endpoint with the group ID.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- `GetGroup` and `ListGroups` return an empty member list. To see group info for a certain member, call `ListGroups` with a member filter. For more information, see ListGroups (p. 24).

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- None

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	Operation is not permitted based on the supplied authorization.
`ThrottlingException`	Too many requests were made that exceed the limits.
`ResourceNotFoundException`	Specified group does not exist.
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups/9067729b3d-a2cfc8a5-
f4ab-4443-9d7d-b32a9013c554
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 200
```

```
Date: Mon, 06 Apr 2020 17:16:53 GMT
Content-Type: application/json
x-amzn-RequestId: abbf9e53-9ecc-46d2-8efe-104a66ff128f
{
    "id": "9067729b3d-a2cfc8a5-f4ab-4443-9d7d-b32a9013c554",
    "meta": {
        "resourceType": "Group",
        "created": "2020-04-06T16:48:19Z",
        "lastModified": "2020-04-06T16:48:19Z"
    },
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:Group"
    ],
    "displayName": "Group Bar"
}
```

ListGroups

You can use the `/Groups` endpoint to filter queries on a list of existing groups by making a `GET` request with additional filter information. Only a maximum of 50 results can be returned. See the **Constraints** section for a list of available filters.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- `GetGroup` and `ListGroups` return an empty member list. To see group info for a certain member, call `ListGroups` with a member filter. (See the examples that follow.)

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- At this time, the `ListGroups` API is only capable of returning up to 50 results.
- Supported filter combinations: (`displayName`), (`id` and `member`), (`member` and `id`). Note that the use of id as an individual filter, while valid, should be avoided as there is already a getGroup endpoint available.
- Supported comparison operator in filters: `eq`
- Filter must be specified as: `<filterAttribute> eq "<filterValue>"`

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	Operation is not permitted based on the supplied authorization.

Error	Condition
`ThrottlingException`	Too many requests were made that exceed the limits.
`ResourceNotFound`	When filter querying with a nonexisting member.
`ValidationException`	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 200
 Date: Thu, 23 Jul 2020 00:37:15 GMT
 Content-Type: application/json
 x-amzn-RequestId: e01400a1-0f10-4e90-ba58-ea1766a009d7

 {
    "totalResults": 6,
    "itemsPerPage": 6,
    "startIndex": 1,
    "schemas": [
        "urn:ietf:params:scim:api:messages:2.0:ListResponse"
    ],
    "Resources": [
        {
            "id": "90677c608a-ef9cb2da-d480-422b-9901-451b1bf9e607",
            "meta": {
                "resourceType": "Group",
                "created": "2020-07-22T23:10:21Z",
                "lastModified": "2020-07-22T23:10:21Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:Group"
            ],
            "displayName": "Group Foo",
            "members": []
        },
        {
            "id": "90677c608a-95aca21b-4bb7-4161-94cb-d885e2920414",
            "meta": {
                "resourceType": "Group",
                "created": "2020-07-23T00:16:49Z",
                "lastModified": "2020-07-23T00:16:49Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:Group"
```

```
            ],
            "displayName": "Group Beta",
            "members": []
        },
        {
            "id": "90677c608a-00dbcb72-e0b2-49a0-86a2-c259369fc6a7",
            "meta": {
                "resourceType": "Group",
                "created": "2020-07-23T00:18:08Z",
                "lastModified": "2020-07-23T00:18:08Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:Group"
            ],
            "displayName": "Group Omega",
            "members": []
        },
        {
            "id": "90677c608a-10d47528-1e68-4730-910e-c8a102121f47",
            "meta": {
                "resourceType": "Group",
                "created": "2020-07-22T22:58:48Z",
                "lastModified": "2020-07-22T22:58:48Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:Group"
            ],
            "displayName": "Group Bar",
            "members": []
        },
        {
            "id": "90677c608a-6ba7b52f-67e5-4849-b64c-15464fe7893b",
            "meta": {
                "resourceType": "Group",
                "created": "2020-07-23T00:14:19Z",
                "lastModified": "2020-07-23T00:14:19Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:Group"
            ],
            "displayName": "Group Delta",
            "members": []
        },
        {
            "id": "90677c608a-a9f17294-7931-41a5-9c00-6e7ace3c2c11",
            "meta": {
                "resourceType": "Group",
                "created": "2020-07-23T00:20:08Z",
                "lastModified": "2020-07-23T00:20:08Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:Group"
            ],
            "displayName": "Group Gamma",
            "members": []
        }
    ]
}
```


Filter examples

For the ListGroup endpoint we support three different combinations of filters as follows:

- `displayName`

- `id` and `member`
- `member` and `id`

The filters can be applied in the formats as shown.

Single filter

```
filter=<filterAttribute> eq "<filterValue>"
```

Two filters

```
filter=<filterAttribute1> eq "<filterValue1>" and <filterAttribute2> eq "<filterValue2>"
```

See the following examples.

displayName

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups?filter=displayName eq
 "Group Bar"
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 200
Date: Wed, 22 Jul 2020 23:06:38 GMT
Content-Type: application/json
x-amzn-RequestId: 45995b44-02cd-419f-87f4-ff8fa323448d

{
    "totalResults": 1,
    "itemsPerPage": 1,
    "startIndex": 1,
    "schemas": [
        "urn:ietf:params:scim:api:messages:2.0:ListResponse"
    ],
    "Resources": [
        {
            "id": "90677c608a-10d47528-1e68-4730-910e-c8a102121f47",
            "meta": {
                "resourceType": "Group",
                "created": "2020-07-22T22:58:48Z",
                "lastModified": "2020-07-22T22:58:48Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:Group"
            ],
            "displayName": "Group Bar",
            "members": []
        }
    ]
}
```

Group id and members

Both group `id` and `members` are interchangeable in order.

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups?filter=id
 eq "90677c608a-a9f17294-7931-41a5-9c00-6e7ace3c2c11" and members eq
 "90677c608a-787142a0-3f27-4cd3-afb6-8aed7ce87094"
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 200
Date: Wed, 22 Jul 2020 23:06:38 GMT
Content-Type: application/json
x-amzn-RequestId: 65d18c02-fc7c-4f2b-9410-ed417acf4fb2

{
    "totalResults": 1,
    "itemsPerPage": 1,
    "startIndex": 1,
    "schemas": [
        "urn:ietf:params:scim:api:messages:2.0:ListResponse"
    ],
    "Resources": [
        {
            "id": "90677c608a-a9f17294-7931-41a5-9c00-6e7ace3c2c11",
            "meta": {
                "resourceType": "Group",
                "created": "2020-07-23T00:20:08Z",
                "lastModified": "2020-07-23T00:20:08Z"
            },
            "schemas": [
                "urn:ietf:params:scim:schemas:core:2.0:Group"
            ],
            "displayName": "Group Gamma",
            "members": []
        }
    ]
}
```

DeleteGroup

The `DELETE` request is also available for the `/Groups` endpoint to delete existing groups using the value of the `id`.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- None

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	Operation is not permitted based on the supplied authorization.
`ThrottlingException`	Too many requests were made that exceed the limits.
`ResourceNotFoundException`	Specified group does not exist.
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
DELETE https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups/9067729b3d-f987ac4d-a175-44f0-a528-6d23c5d2ec4d
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 204
Date: Mon, 06 Apr 2020 22:21:24 GMT
Content-Type: application/json
x-amzn-RequestId: abbf9e53-9ecc-46d2-8efe-104a66ff128
```

PatchGroup

Existing groups can be updated by calling upon the PATCH operation to replace specific attribute values. For more information, see the **Examples** section.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- Only `displayName`, `members`, and `externalId` attributes are allowed in the request.
- A maximum of 100 membership changes are allowed in a single request.

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	Operation is not permitted based on the supplied authorization.
`ThrottlingException`	Too many requests were made that exceed the limits.
`ResourceNotFoundException`	Specified group does not exist.
`ValidationException`	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.
`InternalServerException`	Service failed to process the request.

Examples

Following are example requests and responses for this API operation.

Example Request

```
PATCH https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups/9067729b3d-f987ac4d-
a175-44f0-a528-6d23c5d2ec4d
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>

{
    "schemas": ["urn:ietf:params:scim:api:messages:2.0:PatchOp"],
    "Operations": [{
        "op": "replace",
        "value": {
            "id": "9067729b3d-07124c20-d960-4a47-93ff-460d76461f81",
            "displayName": "Group Foo New"
        }
    }]
}
```

Example Response

```
HTTP/1.1 204
Date: Tue, 07 Apr 2020 23:59:09 GMT
Content-Type: application/json
x-amzn-RequestId: dad0c91c-1ea8-4b36-9fdb-4f099b59c1c9
```

Member operations examples

The following three different member operations are supported using the patch operation for a group.

- Add members to a group
- Replace members in a group (full member list replacement)
- Remove members from a group

These member operations can be applied using the examples below.

Add members to a group

In the value field, provide a list of objects containing the value of the user **id**. Multiple members can be added at a time. For example, if you have a user with the **id** `906722b2be-ee23ed58-6e4e-4b2f-a94a-3ace8456a36c` that you want to add to the group with an **id** of `9067729b3d-f987ac4d-a175-44f0-a528-6d23c5d2ec4d`, use the following call:

Example Request

```
PATCH  https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups/9067729b3d-f987ac4d-
a175-44f0-a528-6d23c5d2ec4d
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>

{
   "schemas":[
      "urn:ietf:params:scim:api:messages:2.0:PatchOp"
   ],
   "Operations":[
      {
         "op":"add",
         "path":"members",
         "value":[
            {
               "value":"906722b2be-ee23ed58-6e4e-4b2f-a94a-3ace8456a36c"
            }
         ]
      }
   ]
}
```

Example Response

```
HTTP/1.1  204
Date: Mon, 21 Sep 2020 16:39:26 GMT
Content-Type: application/json
x-amzn-RequestId: 1e9abe4c-b6e1-4d3b-bb86-73ca6187e08b
```

Replace members in a group

In the value field, provide a list of objects containing the value of the user id. Please note that the new list of members provided in the API call will replace all existing members in the group.

For example, if you want to replace all of the members in the group with an **id** of `9067729b3d-f987ac4d-a175-44f0-a528-6d23c5d2ec4d` with a list of two members, use the following call:

Example Request

```
PATCH  https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups/9067729b3d-f987ac4d-a175-44f0-a528-6d23c5d2ec4d
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>

{
   "schemas":[
      "urn:ietf:params:scim:api:messages:2.0:PatchOp"
   ],
   "Operations":[
      {
         "op":"replace",
         "path":"members",
         "value":[
            {
               "value":"906722b2be-61c204e7-56d0-4dad-882d-f41911b31ccb"
            },
            {
               "value":"906722b2be-da1f7ef3-3e37-473e-95be-df2efaa2590d"
            }
         ]
      }
   ]
}
```

Example Response

```
HTTP/1.1  204
Date: Mon, 21 Sep 2020 16:43:52 GMT
Content-Type: application/json
x-amzn-RequestId: 3155a146-f729-4765-af66-fe01c862e688
```

Remove members from a group

In the value field, provide a list of objects containing the value of the user id. Multiple members can be removed at a time. If the value field contains an empty list or is not provided, all of the path's members will be removed.

For example, if you want to remove two users, one with the **id** `906722b2be-61c204e7-56d0-4dad-882d-f41911b31ccb` and another with the **id** `906722b2be-da1f7ef3-3e37-473e-95be-df2efaa2590d`, use the following call:

Example Request

```
PATCH https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups/9067729b3d-f987ac4d-a175-44f0-a528-6d23c5d2ec4d
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>

{
   "schemas":[
      "urn:ietf:params:scim:api:messages:2.0:PatchOp"
   ],
   "Operations":[
      {
         "op":"remove",
         "path":"members",
         "value":[
            {
```

```
            "value":"906722b2be-61c204e7-56d0-4dad-882d-f41911b31ccb"
         },
         {
            "value":"906722b2be-da1f7ef3-3e37-473e-95be-df2efaa2590d"
         }
      ]
   }
]
}
```

Example Response

```
HTTP/1.1  204
Date: Mon, 21 Sep 2020 17:03:29 GMT
Content-Type: application/json
x-amzn-RequestId: 3f89b66d-77e1-4128-a0c3-a890e72f5d4c
```

ServiceProviderConfig

You can use the `/ServiceProviderConfig` endpoint for `GET` requests to view additional information about the AWS SSO SCIM implementation. The `/ServiceProviderConfig` endpoint is read only.

Not supported

The AWS SSO SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The AWS SSO SCIM implementation has the following constraints for this API operation.

- None

Errors

The following AWS SSO SCIM implementation errors are common for this API operation.

Error	Condition
`UnauthorizedException`	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.
`AccessDeniedException`	Operation is not permitted based on the supplied authorization,
`ThrottlingException`	Too many requests were made that exceed the limits,
`ValidationException`	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported,

Error	Condition
`ConflictException`	User already exists,
`InternalServerException`	Service failed to process the request,

Examples

Following are example requests and responses for this API operation.

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/ServiceProviderConfig
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/1.1 200
Date: Thu, 13 Aug 2020 21:11:26 GMT
Content-Type: application/json
x-amzn-RequestId: d0f671f1-3217-4b0f-b310-82a0b6861967
{
    "schemas": [
        "urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig"
    ],
    "documentationUri": "https://docs.aws.amazon.com/singlesignon/ latest/userguide/manage-
your-identity-source-idp.html",
    "authenticationSchemes": [
        {
            "type": "oauthbearertoken",
            "name": "OAuth Bearer Token",
            "description": "Authentication scheme using the OAuth Bearer Token Standard",
            "specUri": "https://www.rfc-editor.org/info/rfc6750",
            "documentationUri": "https://docs.aws.amazon.com/singlesignon/ latest/
userguide/provision-automatically.html",
            "primary": true
        }
    ],
    "patch": {
        "supported": true
    },
    "bulk": {
        "supported": false,
        "maxOperations": 1,
        "maxPayloadSize": 1048576
    },
    "filter": {
        "supported": true,
        "maxResults": 50
    },
    "changePassword": {
        "supported": false
    },
    "sort": {
        "supported": false
    },
    "etag": {
        "supported": false
    }
```

```
}
```

Making API Requests

AWS SSO SCIM implementation supports the bearer HTTP authentication scheme. An access token (also known as a bearer token) must be passed in the HTTP Authorization header of each request to your SCIM endpoint. See Considerations for Using Automatic Provisioning in the *AWS Single Sign-On User Guide* for instructions on generating and retrieving your access token.

Other authentication schemes described in the SCIM specifications are not supported at this time.

Limitations from SCIM specification

The AWS SSO SCIM implementation supports only a subset of the SCIM specifications. This section lists the limitations that the AWS SSO SCIM implementation has in comparison to the SCIM specifications. These include the following:

- **Filter limitations** – Only `eq` with `and` is supported. We currently do not support any other filters.
- **Endpoint limitations** – Some SCIM protocol endpoints are not supported, such as `/Me`, `/Bulk`, and `/.Search`. AWS SSO supports `/ServiceProviderConfig`. However, `/Schemas` and `/ResourceTypes` are currently not supported.
- **Attribute limitations** – AWS SSO currently does not support multi-valued attributes in general for users. Examples include multiple emails, addresses, and phone numbers.

In addition, AWS SSO also has some attributes that are not supported. The following tables describe which attributes are currently supported.

User attributes – Single valued

Attribute	Subattributes (if applicable)	Supported
`userName`		Yes
`name`	`formatted`	Yes
	`familyName`	Yes
	`givenName`	Yes
	`middleName`	Yes
	`honorificPrefix`	Yes
	`honorificSuffix`	Yes
`displayName`		Yes
`nickName`		Yes
`profileUrl`		Yes
`title`		Yes
`userType`		Yes
`preferredLanguage`		Yes
`locale`		Yes
`timezone`		Yes
`active`		Yes
`password`		No

User attributes – Multi-valued

Attribute	Subattributes (if applicable)	Supported
emails		Partial (single value only)
	display	No
	type	Yes
	values	Yes
	primary	Yes
phoneNumbers		Partial (single value only)
	display	No
	type	Yes
	values	Yes
ims		No
photos		No
addresses		Yes (single value only)
	formatted	Yes
	streetAddress	Yes
	locality	Yes
	region	Yes
	postalCode	Yes
	Country	Yes
groups		Yes
entitlements		No
roles		Yes
x509Certificates		No

Group resource schema attributes – Single value

Attribute	Supported
displayName	Yes

Group resource schema attributes – Multi-value

Attribute	Supported
members	Yes, but cannot be read in a response

Enterprise user schema extension attributes – Single value

Attribute	Subattributes (if applicable)	Supported
`employeeNumber`		Yes
`costCenter`		Yes
`organization`		Yes
`division`		Yes
`department`		Yes
`manager`	`value`	Yes
	`$ref`	Yes
	`displayName`	No

Document History

The following table describes the important changes to the documentation in this release of the *AWS SSO SCIM Implementation Developer Guide*.

- **Latest documentation update:** March 18, 2022

Change	Description	Date Changed
Updates to GetGroup and ListGroups	Added information about what's not supported for GetGroup. Fixed a typo in ListGroups.	March 18, 2022
Updates to CreateGroup	Added a constraint for the maximum members allowed in a single request.	March 1, 2022
Updates to PatchUser	Added `nickName` to the bullet covering which modifications are allowed.	February 1, 2022
Updates to ListUsers and ListGroups	Added information about 50 maximum results returned.	April 9, 2021
Updates to PatchGroup and added new topic	Added member operations examples to PatchGroup and added new Making API Requests topic.	September 28, 2020
New guide	This is the first release of the *AWS SSO SCIM Implementation Developer Guide*.	August 27, 2020

AWS glossary

For the latest AWS terminology, see the AWS glossary in the *AWS General Reference*.