



Developer Guide

IAM Identity Center SCIM Implementation



IAM Identity Center SCIM Implementation: Developer Guide

Copyright © 2024 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

Amazon's trademarks and trade dress may not be used in connection with any product or service that is not Amazon's, in any manner that is likely to cause confusion among customers, or in any manner that disparages or discredits Amazon. All other trademarks not owned by Amazon are the property of their respective owners, who may or may not be affiliated with, connected to, or sponsored by Amazon.

Table of Contents

What is the IAM Identity Center SCIM implementation?	1
Supported API operations	2
CreateGroup	2
Not supported	2
Constraints	3
Errors	3
Examples	4
CreateUser	5
Not supported	5
Constraints	5
Errors	5
Examples	6
DeleteGroup	9
Not supported	10
Constraints	10
Errors	10
Examples	11
DeleteUser	11
Not supported	11
Constraints	12
Errors	12
Examples	13
GetGroup	13
Not supported	13
Constraints	13
Errors	14
Examples	14
GetSchema	15
Not supported	15
Constraints	15
Errors	16
Example	16
GetUser	17
Not supported	17

Constraints	18
Errors	18
Examples	19
ListGroups	20
Not supported	20
Constraints	20
Errors	20
Examples	21
Filter examples	24
ListResourceTypes	27
Not supported	27
Constraints	27
Errors	28
Examples	28
ListSchemas	30
Not supported	30
Constraints	30
Errors	30
Example	31
ListUsers	48
Not supported	49
Constraints	49
Errors	49
Examples	50
Filter examples	54
PatchGroup	59
Not supported	59
Constraints	60
Errors	60
Examples	61
Member operations examples	61
PatchUser	65
Not supported	65
Constraints	65
Errors	65
Examples	66

PutUser	69
Not supported	69
Constraints	69
Errors	69
Examples	70
ServiceProviderConfig	74
Not supported	74
Constraints	74
Errors	74
Examples	75
Making API Requests	77
Limitations	78
Document History	82
AWS Glossary	84

What is the IAM Identity Center SCIM implementation?

This reference guide helps software developers build custom integrations to provision (synchronize) users and groups into AWS IAM Identity Center 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.

Note

IAM Identity Center uses the `sso` and `identitystore` API namespaces.

The IAM Identity Center 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 IAM Identity Center are described in the [Supported API operations](#) section of this guide.

The following sections contain examples of API requests and responses currently supported in the IAM Identity Center 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 *IAM Identity Center User Guide*. That topic instructs you how to use SCIM to enable automatic provisioning in IAM Identity Center. 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 IAM Identity Center SCIM implementation:

Topics

- [CreateGroup](#)
- [CreateUser](#)
- [DeleteGroup](#)
- [DeleteUser](#)
- [GetGroup](#)
- [GetSchema](#)
- [GetUser](#)
- [ListGroups](#)
- [ListResourceTypes](#)
- [ListSchemas](#)
- [ListUsers](#)
- [PatchGroup](#)
- [PatchUser](#)
- [PutUser](#)
- [ServiceProviderConfig](#)

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 IAM Identity Center SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The IAM Identity Center 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 IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
<code>ValidationException</code>	Request can't be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400
<code>UnauthorizedException</code>	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
<code>AccessDeniedException</code>	Operation isn't permitted based on the supplied authorization.	403
<code>ConflictException</code>	Group already exists.	409
<code>ThrottlingException</code>	Too many requests exceeded the limits.	429
<code>InternalServerErrorException</code>	Service failed to process the request.	500

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>

{
  "externalId": "701984",
  "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",
  "externalId": "701984",
  "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"
}
```

CreateUser

You can create new users from a POST request using the IAM Identity Center SCIM implementation /Users endpoint. See the Examples section.

Not supported

The IAM Identity Center 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 IAM Identity Center 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 IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
<code>ValidationException</code>	Request cannot be parsed, is syntactically incorrect, or	400

Error	Condition	HTTP Status Code
	violates schema. This error also occurs if the operation is unsupported.	
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ConflictException	User already exists.	409
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

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>

{
  "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"
  }
}
}
```

DeleteGroup

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

Not supported

The IAM Identity Center SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The IAM Identity Center SCIM implementation has the following constraints for this API operation.

- None

Errors

The following IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
ValidationException	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ResourceNotFoundException	Specified group does not exist.	404

Error	Condition	HTTP Status Code
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

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
```

DeleteUser

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

Not supported

The IAM Identity Center SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The IAM Identity Center SCIM implementation has the following constraints for this API operation.

- None

Errors

The following IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
ValidationException	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ResourceNotFoundException	Specified user does not exist.	404
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

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
```

GetGroup

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

Not supported

The IAM Identity Center 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](#).

Constraints

The IAM Identity Center SCIM implementation has the following constraints for this API operation.

- None

Errors

The following IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
ValidationException	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ResourceNotFoundException	Specified group does not exist.	404
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

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"
}
```

GetSchema

Information about supported SCIM schemas can be retrieved by making a request to the `/Schemas` endpoint. See the Examples section.

Not supported

The IAM Identity Center SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The IAM Identity Center SCIM implementation has the following constraints for this API operation.

- None

Errors

The following IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
ValidationException	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

Example

Following are example request and response for this API operation.

Example Request

```
GET /{tenant_id}/scim/v2/Schemas/urn:ietf:params:scim:schemas:core:2.0:User
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/2 200 OK
Date: Fri, 12 May 2023 17:00:33 GMT
Content-Type: application/json
Content-Length: 2
x-amzn-Requestid: 80cc7268-02b8-4e37-a787-da7b9b7a1952
{
  "id" : "urn:ietf:params:scim:schemas:core:2.0:User",
  "name" : "User",
  "description" : "User Account",
  "attributes" : [
    {
      "name" : "userName",
      "type" : "string",
      "multiValued" : false,
      "description" : "Unique identifier for the User, typically used by the user
to directly authenticate to the service provider. Each User MUST include a non-empty
userName value. This identifier MUST be unique across the service provider's entire
set of Users. REQUIRED.",
      "required" : true,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "server"
    }
  ]
}
```

GetUser

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

Not supported

The IAM Identity Center SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The IAM Identity Center SCIM implementation has the following constraints for this API operation.

- None

Errors

The following IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
ValidationException	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	The operation is not permitted based on the supplied authorization.	403
ResourceNotFoundException	Specified user or endpoint does not exist.	404
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

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
    }
  ]
}
```



```
}
```

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 IAM Identity Center 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 IAM Identity Center 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`), (`externalId`), (`id` and `member`), and (`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 IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
<code>ValidationException</code>	Request cannot be parsed, is syntactically incorrect, or violates schema. This error	400

Error	Condition	HTTP Status Code
	also occurs if the operation is unsupported.	
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ResourceNotFoundException	When filter querying with a nonexisting member.	404
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

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",
      "externalId": "702135",
      "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
- externalId
- 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": []
    }
  ]
}
```

externalId

Example Request

```
GET https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/Groups?filter=externalId
  eq "705167"
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-ef9cb2da-d480-422b-9901-451b1bf9e607",
      "externalId": "702135",
      "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": []
    }
  ]
}
```

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": []
    }
  ]
}
```

ListResourceTypes

Information about supported resource types can be retrieved by making a request to the `/ResourceTypes` endpoint.

Not supported

The IAM Identity Center SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The IAM Identity Center SCIM implementation has the following constraints for this API operation.

- None.

Errors

The following IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
ValidationException	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

Examples

Following are example requests and responses for this API operation.

Example Request

```
GET /{tenant_id}/scim/v2/ResourceTypes HTTP/2
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/2 200 OK
Date: Fri, 12 May 2023 16:59:31 GMT
Content-Type: application/json
Content-Length: 2
X-Amzn-Requestid: 5b6de8b5-36a3-4179-8afa-71e918d85516
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 2,
  "itemsPerPage": 2,
  "startIndex": 1,
  "Resources": [
    {
      "schemas": ["urn:ietf:params:scim:schemas:core:2.0:ResourceType"],
      "id": "User",
      "name": "User",
      "endpoint": "/Users",
      "description": "User Account",
      "schema": "urn:ietf:params:scim:schemas:core:2.0:User",
      "schemaExtensions": [
        {
          "schema":
            "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
          "required": true
        }
      ],
      "meta": {
        "resourceType": "ResourceType"
        "\"location\": \"https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/ResourceTypes/User\""
      }
    },
    {
      "schemas": ["urn:ietf:params:scim:schemas:core:2.0:ResourceType"],
      "id": "Group",
      "name": "Group",
      "endpoint": "/Groups",
      "description": "Group",
      "schema": "urn:ietf:params:scim:schemas:core:2.0:Group",
      "meta": {
        "resourceType": "ResourceType"
      }
    }
  ]
}
```

```
    "location": "https://scim.us-east-1.amazonaws.com/{tenant_id}/scim/v2/ResourceTypes/Group"
  }
}
```

ListSchemas

Information about supported SCIM schemas can be retrieved by making a request to the `/Schema` endpoint. See the Examples section.

Not supported

The IAM Identity Center SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The IAM Identity Center SCIM implementation has the following constraints for this API operation.

- None

Errors

The following IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
ValidationException	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400

Error	Condition	HTTP Status Code
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

Example

Following are example request and response for this API operation.

Example Request

```
GET /{tenant_id}/scim/v2/Schemas
User-Agent: Mozilla/5.0
Authorization: Bearer <bearer_token>
```

Example Response

```
HTTP/2 200 OK
Date: Fri, 12 May 2023 17:00:33 GMT
Content-Type: application/json
Content-Length: 2
x-amzn-Requestid: 80cc7268-02b8-4e37-a787-da7b9b7a1952
{
  "schemas": [
    "urn:ietf:params:scim:api:messages:2.0:ListResponse"
  ],
  "totalResults": 3,
  "itemsPerPage": 3,
```

```
"startIndex": 1,
"Resources": [
{
  "id" : "urn:ietf:params:scim:schemas:core:2.0:User",
  "name" : "User",
  "description" : "User Schema",
  "attributes" : [
    {
      "name" : "userName",
      "type" : "string",
      "multiValued" : false,
      "description" : "Unique identifier for the User, typically used by the user
to directly authenticate to the service provider. Each User MUST include a non-empty
userName value. This identifier MUST be unique across the service provider's entire
set of Users. REQUIRED.",
      "required" : true,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "server"
    },
    {
      "name" : "name",
      "type" : "complex",
      "multiValued" : false,
      "description" : "The components of the user's real name. Providers MAY return
just the full name as a single string in the formatted sub-attribute, or they MAY
return just the individual component attributes using the other sub-attributes, or
they MAY return both. If both variants are returned, they SHOULD be describing the
same name, with the formatted name indicating how the component attributes should be
combined.",
      "required" : false,
      "subAttributes" : [
        {
          "name" : "formatted",
          "type" : "string",
          "multiValued" : false,
          "description" : "The full name, including all middle names, titles, and
suffixes as appropriate, formatted for display (e.g., 'Ms. Barbara J Jensen, III').",
          "required" : false,
          "caseExact" : false,
          "mutability" : "readWrite",
          "returned" : "default",
          "uniqueness" : "none"
        }
      ]
    }
  ]
}
```

```
    },
    {
      "name" : "familyName",
      "type" : "string",
      "multiValued" : false,
      "description" : "The family name of the User, or last name in most Western languages (e.g., 'Jensen' given the full name 'Ms. Barbara J Jensen, III').",
      "required" : true,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "givenName",
      "type" : "string",
      "multiValued" : false,
      "description" : "The given name of the User, or first name in most Western languages (e.g., 'Barbara' given the full name 'Ms. Barbara J Jensen, III').",
      "required" : true,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "middleName",
      "type" : "string",
      "multiValued" : false,
      "description" : "The middle name(s) of the User (e.g., 'Jane' given the full name 'Ms. Barbara J Jensen, III').",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "honorificPrefix",
      "type" : "string",
      "multiValued" : false,
      "description" : "The honorific prefix(es) of the User, or title in most Western languages (e.g., 'Ms.' given the full name 'Ms. Barbara J Jensen, III').",
      "required" : false,
```

```
        "caseExact" : false,
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
    },
    {
        "name" : "honorificSuffix",
        "type" : "string",
        "multiValued" : false,
        "description" : "The honorific suffix(es) of the User, or suffix in most
Western languages (e.g., 'III' given the full name 'Ms. Barbara J Jensen, III').",
        "required" : false,
        "caseExact" : false,
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
    }
],
"mutability" : "readWrite",
"returned" : "default",
"uniqueness" : "none"
},
{
    "name" : "displayName",
    "type" : "string",
    "multiValued" : false,
    "description" : "The name of the User, suitable for display to end-users. The
name SHOULD be the full name of the User being described, if known.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
},
{
    "name" : "nickName",
    "type" : "string",
    "multiValued" : false,
    "description" : "The casual way to address the user in real life, e.g., 'Bob'
or 'Bobby' instead of 'Robert'. This attribute SHOULD NOT be used to represent a
User's username (e.g., 'bjensen' or 'mpepperidge').",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
```

```
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "profileUrl",
    "type" : "reference",
    "referenceTypes" : ["external"],
    "multiValued" : false,
    "description" : "A fully qualified URL pointing to a page representing the
User's online profile.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "title",
    "type" : "string",
    "multiValued" : false,
    "description" : "The user's title, such as \"Vice President.\",",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "userType",
    "type" : "string",
    "multiValued" : false,
    "description" : "Used to identify the relationship between the organization and
the user. Typical values used might be 'Contractor', 'Employee', 'Intern', 'Temp',
'External', and 'Unknown', but any value may be used.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "preferredLanguage",
    "type" : "string",
    "multiValued" : false,
```



```
    "description" : "Indicates the User's preferred written or spoken language.
Generally used for selecting a localized user interface; e.g., 'en_US' specifies the
language English and country US.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "locale",
    "type" : "string",
    "multiValued" : false,
    "description" : "Used to indicate the User's default location for purposes of
localizing items such as currency, date time format, or numerical representations.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "timezone",
    "type" : "string",
    "multiValued" : false,
    "description" : "The User's time zone in the 'Olson' time zone database format,
e.g., 'America/Los_Angeles'.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "active",
    "type" : "boolean",
    "multiValued" : false,
    "description" : "A Boolean value indicating the User's administrative status.",
    "required" : false,
    "mutability" : "readWrite",
    "returned" : "default"
  },
  {
    "name" : "emails",
```

```
    "type" : "complex",
    "multiValued" : true,
    "description" : "Email addresses for the user. The value SHOULD be
canonicalized by the service provider, e.g., 'bjensen@example.com' instead of
'bjensen@EXAMPLE.COM'. Canonical type values of 'work', 'home', and 'other'.",
    "required" : false,
    "subAttributes" : [
      {
        "name" : "value",
        "type" : "string",
        "multiValued" : false,
        "description" : "Email addresses for the user. The value SHOULD be
canonicalized by the service provider, e.g., 'bjensen@example.com' instead of
'bjensen@EXAMPLE.COM'. Canonical type values of 'work', 'home', and 'other'.",
        "required" : false,
        "caseExact" : false,
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
      },
      {
        "name" : "type",
        "type" : "string",
        "multiValued" : false,
        "description" : "A label indicating the attribute's function, e.g., 'work'
or 'home'.",
        "required" : false,
        "caseExact" : false,
        "canonicalValues" : [
          "work",
          "home",
          "other"
        ],
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
      },
      {
        "name" : "primary",
        "type" : "boolean",
        "multiValued" : false,
        "description" : "A Boolean value indicating the 'primary' or preferred
attribute value for this attribute, e.g., the preferred mailing address or primary
email address. The primary attribute value 'true' MUST appear no more than once.",
```

```
        "required" : false,
        "mutability" : "readWrite",
        "returned" : "default"
    }
],
"mutability" : "readWrite",
"returned" : "default",
"uniqueness" : "none"
},
{
    "name" : "phoneNumbers",
    "type" : "complex",
    "multiValued" : true,
    "description" : "Phone numbers for the User. The value SHOULD be canonicalized
by the service provider according to the format specified in RFC 3966, e.g., 'tel:
+1-201-555-0123'. Canonical type values of 'work', 'home', 'mobile', 'fax', 'pager',
and 'other'.",
    "required" : false,
    "subAttributes" : [
        {
            "name" : "value",
            "type" : "string",
            "multiValued" : false,
            "description" : "Phone number of the User.",
            "required" : false,
            "caseExact" : false,
            "mutability" : "readWrite",
            "returned" : "default",
            "uniqueness" : "none"
        },
        {
            "name" : "type",
            "type" : "string",
            "multiValued" : false,
            "description" : "A label indicating the attribute's function, e.g., 'work',
'home', 'mobile'.",
            "required" : false,
            "caseExact" : false,
            "canonicalValues" : [
                "work",
                "home",
                "mobile",
                "fax",
                "pager",
```

```
        "other"
      ],
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "primary",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value indicating the 'primary' or preferred
attribute value for this attribute, e.g., the preferred phone number or primary phone
number. The primary attribute value 'true' MUST appear no more than once.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    }
  ],
  "mutability" : "readWrite",
  "returned" : "default"
},
{
  "name" : "addresses",
  "type" : "complex",
  "multiValued" : true,
  "description" : "A physical mailing address for this User. Canonical type
values of 'work', 'home', and 'other'. This attribute is a complex type with the
following sub-attributes.",
  "required" : false,
  "subAttributes" : [
    {
      "name" : "formatted",
      "type" : "string",
      "multiValued" : false,
      "description" : "The full mailing address, formatted for display or use
with a mailing label. This attribute MAY contain newlines.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "streetAddress",
```

```
    "type" : "string",
    "multiValued" : false,
    "description" : "The full street address component, which may include house
number, street name, P.O. box, and multi-line extended street address information.
This attribute MAY contain newlines.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "locality",
    "type" : "string",
    "multiValued" : false,
    "description" : "The city or locality component.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "region",
    "type" : "string",
    "multiValued" : false,
    "description" : "The state or region component.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "postalCode",
    "type" : "string",
    "multiValued" : false,
    "description" : "The zip code or postal code component.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
}
```

```
{
  "name" : "country",
  "type" : "string",
  "multiValued" : false,
  "description" : "The country name component.",
  "required" : false,
  "caseExact" : false,
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
},
{
  "name" : "type",
  "type" : "string",
  "multiValued" : false,
  "description" : "A label indicating the attribute's function, e.g., 'work'
or 'home'.",
  "required" : false,
  "caseExact" : false,
  "canonicalValues" : [
    "work",
    "home",
    "other"
  ],
  "mutability" : "readWrite",
  "returned" : "default",
  "uniqueness" : "none"
}
],
"mutability" : "readWrite",
"returned" : "default",
"uniqueness" : "none"
},
{
  "name" : "groups",
  "type" : "complex",
  "multiValued" : true,
  "description" : "A list of groups to which the user belongs, either through
direct membership, through nested groups, or dynamically calculated.",
  "required" : false,
  "subAttributes" : [
    {
      "name" : "value",
      "type" : "string",
```

```
    "multiValued" : false,
    "description" : "The identifier of the User's group.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readOnly",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "$ref",
    "type" : "reference",
    "referenceTypes" : [
      "User",
      "Group"
    ],
    "multiValued" : false,
    "description" : "The URI of the corresponding 'Group' resource to which the
user belongs.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readOnly",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "type",
    "type" : "string",
    "multiValued" : false,
    "description" : "A label indicating the attribute's function, e.g.,
'direct' or 'indirect'.",
    "required" : false,
    "caseExact" : false,
    "canonicalValues" : [
      "direct",
      "indirect"
    ],
    "mutability" : "readOnly",
    "returned" : "default",
    "uniqueness" : "none"
  }
],
"mutability" : "readOnly",
"returned" : "default"
},
```

```
{
  "name" : "roles",
  "type" : "complex",
  "multiValued" : true,
  "description" : "A list of roles for the User that collectively represent who
the User is, e.g., 'Student', 'Faculty'.",
  "required" : false,
  "subAttributes" : [
    {
      "name" : "value",
      "type" : "string",
      "multiValued" : false,
      "description" : "The value of a role.",
      "required" : false,
      "caseExact" : false,
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "type",
      "type" : "string",
      "multiValued" : false,
      "description" : "A label indicating the attribute's function.",
      "required" : false,
      "caseExact" : false,
      "canonicalValues" : [],
      "mutability" : "readWrite",
      "returned" : "default",
      "uniqueness" : "none"
    },
    {
      "name" : "primary",
      "type" : "boolean",
      "multiValued" : false,
      "description" : "A Boolean value indicating the 'primary' or preferred
attribute value for this attribute. The primary attribute value 'true' MUST appear no
more than once.",
      "required" : false,
      "mutability" : "readWrite",
      "returned" : "default"
    }
  ],
  "mutability" : "readWrite",
```



```
        "returned" : "default"
    }
],
"meta" : {
    "resourceType" : "Schema",
    "location" :
        "/v2/Schemas/urn:ietf:params:scim:schemas:core:2.0:User"
}
},
{
    "id" : "urn:ietf:params:scim:schemas:core:2.0:Group",
    "name" : "Group",
    "description" : "Group Schema",
    "attributes" : [
        {
            "name" : "displayName",
            "type" : "string",
            "multiValued" : false,
            "description" : "A human-readable name for the Group. REQUIRED.",
            "required" : true,
            "caseExact" : false,
            "mutability" : "readWrite",
            "returned" : "default",
            "uniqueness" : "none"
        },
        {
            "name" : "members",
            "type" : "complex",
            "multiValued" : true,
            "description" : "A list of members of the Group.",
            "required" : false,
            "subAttributes" : [
                {
                    "name" : "value",
                    "type" : "string",
                    "multiValued" : false,
                    "description" : "Identifier of the member of this Group.",
                    "required" : false,
                    "caseExact" : false,
                    "mutability" : "immutable",
                    "returned" : "default",
                    "uniqueness" : "none"
                }
            ],
        }
    ]
}
```

```
    "name" : "$ref",
    "type" : "reference",
    "referenceTypes" : [
      "User",
      "Group"
    ],
    "multiValued" : false,
    "description" : "The URI corresponding to a SCIM resource that is a member
of this Group.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "immutable",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "type",
    "type" : "string",
    "multiValued" : false,
    "description" : "A label indicating the type of resource, e.g., 'User' or
'Group'.",
    "required" : false,
    "caseExact" : false,
    "canonicalValues" : [
      "User",
      "Group"
    ],
    "mutability" : "immutable",
    "returned" : "default",
    "uniqueness" : "none"
  }
],
"mutability" : "readWrite",
"returned" : "default"
}
],
"meta" : {
  "resourceType" : "Schema",
  "location" :
"/v2/Schemas/urn:ietf:params:scim:schemas:core:2.0:Group"
}
},
{
  "id" : "urn:ietf:params:scim:schemas:extension:enterprise:2.0:User",
```

```
"name" : "EnterpriseUser",
"description" : "Enterprise User Schema",
"attributes" : [
  {
    "name" : "employeeNumber",
    "type" : "string",
    "multiValued" : false,
    "description" : "Numeric or alphanumeric identifier assigned to a person,
typically based on order of hire or association with an organization.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "costCenter",
    "type" : "string",
    "multiValued" : false,
    "description" : "Identifies the name of a cost center.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "organization",
    "type" : "string",
    "multiValued" : false,
    "description" : "Identifies the name of an organization.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "division",
    "type" : "string",
    "multiValued" : false,
    "description" : "Identifies the name of a division.",
    "required" : false,
    "caseExact" : false,
```

```
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "department",
    "type" : "string",
    "multiValued" : false,
    "description" : "Identifies the name of a department.",
    "required" : false,
    "caseExact" : false,
    "mutability" : "readWrite",
    "returned" : "default",
    "uniqueness" : "none"
  },
  {
    "name" : "manager",
    "type" : "complex",
    "multiValued" : false,
    "description" : "The User's manager. A complex type that optionally allows
service providers to represent organizational hierarchy by referencing the 'id'
attribute of another User.",
    "required" : false,
    "subAttributes" : [
      {
        "name" : "value",
        "type" : "string",
        "multiValued" : false,
        "description" : "The id of the SCIM resource representing the User's
manager. REQUIRED.",
        "required" : false,
        "caseExact" : false,
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
      },
      {
        "name" : "$ref",
        "type" : "reference",
        "referenceTypes" : [
          "User"
        ],
        "multiValued" : false,
```

```

        "description" : "The URI of the SCIM resource representing the User's
manager. REQUIRED.",
        "required" : false,
        "caseExact" : false,
        "mutability" : "readWrite",
        "returned" : "default",
        "uniqueness" : "none"
    },
    {
        "name" : "displayName",
        "type" : "string",
        "multiValued" : false,
        "description" : "The displayName of the User's manager. OPTIONAL and READ-
ONLY.",
        "required" : false,
        "caseExact" : false,
        "mutability" : "readOnly",
        "returned" : "default",
        "uniqueness" : "none"
    }
],
"mutability" : "readWrite",
"returned" : "default"
}
],
"meta" : {
    "resourceType" : "Schema",
    "location" :
"/v2/Schemas/urn:ietf:params:scim:schemas:extension:enterprise:2.0:User"
}
}
]
}

```

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 IAM Identity Center 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 IAM Identity Center 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 IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
<code>ValidationException</code>	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400
<code>UnauthorizedException</code>	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401

Error	Condition	HTTP Status Code
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

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"
        }
    }
}
]
```

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 IAM Identity Center SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The IAM Identity Center 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 IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
<code>ValidationException</code>	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400
<code>UnauthorizedException</code>	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
<code>AccessDeniedException</code>	Operation is not permitted based on the supplied authorization.	403
<code>ResourceNotFoundException</code>	Specified group does not exist.	404
<code>ThrottlingException</code>	Too many requests exceeded the limits.	429
<code>InternalServerErrorException</code>	Service failed to process the request.	500

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
```

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 IAM Identity Center 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 IAM Identity Center 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 IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
ValidationException	Request cannot be parsed, is syntactically incorrect, or violates schema. This error also occurs if the operation is unsupported.	400
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ResourceNotFoundException	Specified user does not exist.	404
ConflictException	User already exists.	409
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

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"
  }
}
}
```

PutUser

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

Not supported

The IAM Identity Center 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 IAM Identity Center 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).
- IAM Identity Center 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 IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
<code>ValidationException</code>	Request cannot be parsed, is syntactically incorrect, or	400

Error	Condition	HTTP Status Code
	violates schema. This error also occurs if the operation is unsupported.	
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ConflictException	User already exists.	409
ThrottlingException	Too many requests exceeded the limits.	429
InternalServerErrorException	Service failed to process the request.	500

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"
  }
}
}
```

ServiceProviderConfig

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

Not supported

The IAM Identity Center SCIM implementation does not support the following aspects of this API operation.

- None

Constraints

The IAM Identity Center SCIM implementation has the following constraints for this API operation.

- None

Errors

The following IAM Identity Center SCIM implementation errors are common for this API operation.

Error	Condition	HTTP Status Code
UnauthorizedException	Authorization header is invalid or missing. This error also occurs if the tenant ID is incorrect.	401
AccessDeniedException	Operation is not permitted based on the supplied authorization.	403
ThrottlingException	Too many requests were made that exceed the limits.	429

Error	Condition	HTTP Status Code
InternalServerErrorException	Service failed to process the request.	500

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

IAM Identity Center 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 *IAM Identity Center 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 IAM Identity Center SCIM implementation supports only a subset of the SCIM specifications. This section lists the limitations that the IAM Identity Center 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. IAM Identity Center supports /ServiceProviderConfig. However, /Schemas and /ResourceTypes are currently not supported.
- **Attribute limitations** – IAM Identity Center currently does not support multi-valued attributes in general for users. Examples include multiple emails, addresses, and phone numbers.

In addition, IAM Identity Center 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

Attribute	Subattributes (if applicable)	Supported
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

Attribute	Subattributes (if applicable)	Supported
ims		No
photos		No
addresses		Yes (single value only)
	formatted	Yes
	streetAddress	Yes
	locality	Yes
	region	Yes
	postalCode	Yes
	Country	Yes
groups		No
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 *IAM Identity Center SCIM Implementation Developer Guide*.

- **Latest documentation update:** July 26, 2022

Change	Description	Date Changed
Added ListResourceTypes and ListSchemas	Added the following API operations ListResourceTypes and ListSchemas .	February 7, 2024
AWS Single Sign-On (AWS SSO) renamed to AWS IAM Identity Center	AWS introduces AWS IAM Identity Center. IAM Identity Center expands the capabilities of AWS Identity and Access Management (IAM) to help you centrally manage account and access to cloud applications for your workforce users. For more information, see IAM Identity Center rename .	July 26, 2022
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

Change	Description	Date Changed
New guide	This is the first release of the <i>IAM Identity Center SCIM Implementation Developer Guide</i> .	August 27, 2020

AWS Glossary

For the latest AWS terminology, see the [AWS glossary](#) in the *AWS Glossary Reference*.