SCIM
Internet-Draft
Intended status: Standards Track
Expires: 3 September 2026

D. Zollner
Okta

2 March 2026

SCIM 2.0 Group Member Resource
draft-zollner-scim-group-members-00

Abstract

This document extends the System for Cross-domain Identity Management (SCIM) 2.0 standard by defining a new "GroupMember" top-level resource. Under the existing model defined in [RFC7643], group memberships are represented as values in a multi-valued attribute within a Group resource. This architecture lacks native support for server-side pagination, filtering, or sorting of individual members. In deployments managing large-scale groups (e.g., 100,000 to 1,000,000 members or more), retrieving a Group resource results in massive HTTP response payloads that can exceed 100MB in size. This can lead to service timeouts, memory exhaustion, and network instability, and has led to many major SCIM implementations choosing to not support returning the value of the "members" attribute for Group resources. This extension introduces a flattened resource model that enables group memberships to benefit from pagination and other SCIM protocol features, ensuring interoperability and performance at scale.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 September 2026.

Table of Contents

1. Discussion Venues

This note is to be removed before publishing as an RFC.

Source for this draft and an issue tracker can be found at https://github.com/Zollnerd/scim-group-membership.

2. Introduction

The System for Cross-domain Identity Management (SCIM) 2.0 protocol RFC7643 [RFC7644] is widely used for automating the provisioning of identities across disparate systems. While SCIM excels at managing individual User and Group resources, its design for representing relationships, specifically group memberships, encounters significant performance bottlenecks in large-scale enterprise environments.

Currently, the "members" attribute of a Group resource is a multi-valued attribute. Because SCIM only supports paginating resources, a client requesting a Group resource must receive the entire list of group members in a single HTTP response. For a group with one million members, an HTTP response can reach approximately 200MB in size. These large payloads create several critical failure points including memory pressure and network timeouts.

This document proposes the "GroupMember" resource type. By treating a membership as a first-class, top-level resource, Service Providers can leverage existing SCIM query parameters including filter, count, and multiple pagination methods, allowing them to implement a scaleable and reliable interface for managing groups of any size.

3. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

4. The GroupMember Resource

This section defines the GroupMember resource, which represents a single membership relationship between a SCIM Group and a member. By representing each membership as a distinct, top-level resource, Service Providers can manage group memberships individually, allowing for pagination, filtering, and other operations at scale.

4.1. Resource Properties

The GroupMember resource is defined by the following properties:

schemas
: A multi-valued attribute that contains the SCIM schema URNs for this resource. The URN for the GroupMember resource's core schema is urn:ietf:params:scim:schemas:core:2.0:GroupMember. This is a *REQUIRED* attribute.

id
: A unique identifier for the GroupMember resource, generated by the Service Provider. This is a *REQUIRED*, read-only attribute. Clients MUST treat this value as opaque.

group
: A complex attribute that provides a reference to the parent Group resource. This attribute contains the following sub-attributes:

  value
  : The id of the referenced Group resource. *REQUIRED*.

  $ref
  : The URI of the referenced Group resource. Read-only.

member
: A complex attribute that provides a reference to the member resource, which can be a User, another Group, or any other resource type that can be a member of a group. This attribute contains the following sub-attributes:

  value
  : The id of the referenced member resource. *REQUIRED*.

  $ref
  : The URI of the referenced member resource. Read-only.

  type
  : A string that specifies the resource type of the member, e.g., "User" or "Group". Read-only.

meta
: A complex attribute containing metadata about the resource. This includes the resourceType (which MUST be "GroupMember"), created, lastModified, and location attributes. This is a REQUIRED, read-only attribute.

4.2. JSON Representation

The following is an example of a GroupMember resource in JSON format. This example represents the membership of a User in a Group. ($ref values truncated for formatting purposes):

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:GroupMember"],
  "id": "gm12345",
  "group": {
    "value": "e9e30dba-f08f-4139-944c-2e6949b80b05",
    "$ref": "https://example.com/scim/v2/Groups/e9e3xxx"
  },
  "member": {
    "value": "2819c223-7f76-453a-919d-413861904646",
    "$ref": "https://example.com/scim/v2/Users/2819xxx",
    "type": "User"
  },
  "meta": {
    "resourceType": "GroupMember",
    "created": "2026-02-24T20:26:44Z",
    "lastModified": "2026-02-24T20:26:44Z",
    "location": "https://example.com/scim/v2/GroupMembers/gm12345"
  }
}
```

4.3. Resource Type Representation

The Service Provider’s ResourceType schema, available at the /ResourceTypes endpoint, MUST include an entry for "GroupMember".

Example ResourceType entry:

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:ResourceType"],
  "id": "GroupMember",
  "name": "GroupMember",
  "endpoint": "/GroupMembers",
  "description": "Resource representing a single group membership.",
  "schema": "urn:ietf:params:scim:schemas:core:2.0:GroupMember"
}
```

5. membersMetadata Group Schema Extension

To prevent ambiguity and provide a clear path for clients, this specification also defines an extension schema for the Group resource. This extension introduces a new complex attribute, membersMetadata, which signals how group memberships are managed and provides metadata about those memberships.

When a Service Provider supports the /GroupMembers endpoint, it SHOULD include the membersMetadata attribute on Group resources to declare its membership management policy for that group. The schema URN for the membersMetadata schema extension is urn:ietf:params:scim:schemas:extension:groupMembers:2.0:Group

5.1. The membersMetadata Attribute

The membersMetadata attribute is a complex attribute with the following sub-attributes:

policy
: A REQUIRED string that specifies how membership for this group is represented. It MUST have one of the following values:

  inline
  : Indicates that this group's members are fully represented in the members attribute. Clients SHOULD NOT use the /GroupMembers endpoint for this group.

  external
  : Indicates that this group's members are managed exclusively via the /GroupMembers endpoint. The members attribute MUST be omitted from this Group resource representation.

  hybrid
  : Indicates that the Service Provider MAY return members in the members attribute, but the canonical method for managing memberships is via /GroupMembers. Clients SHOULD prefer using the /GroupMembers endpoint for reliability and scale.

ref
: A REQUIRED URI that a client can use to query for the group's members. It MUST be the URI of the /GroupMembers endpoint with a pre-populated filter for the current group's ID. Its format is [GroupMembers_Endpoint]?filter=group.value eq "[Group_ID]".

memberCount
: An OPTIONAL non-negative integer indicating the total number of members in the group.

allowedMemberTypes
An OPTIONAL multi-valued attribute containing a list of strings that specify the resource types (resourceType) of members allowed in this group.

5.2. Example Group Resources

5.2.1. Example of an "External" Policy

The following is a Group with a large number of members. The policy is "external", the members attribute is absent, and the client is directed to use the ref URI.

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
  "id": "e9e30dba-f08f-4139-944c-2e6949b80b05",
  "displayName": "All Employees",
  "urn:ietf:params:scim:schemas:extension:groupMembers:2.0: \
  Group:membersMetadata": {
    "policy": "external",
    "ref": "https://example.com/scim/v2/GroupMembers?filter= \
    group.value%20eq%20%22e9e30dba-f08f-4139-944c-2e6949b80b05%22",
    "memberCount": 150321
  },
  "meta": {
    "resourceType": "Group",
    "location": "https://example.com/scim/v2/Groups/ \
      e9e30dba-f08f-4139-944c-2e6949b80b05"
  }
}
```

5.2.2. Example of a "Hybrid" Policy

The following is a Group with a small number of members. The policy is "hybrid", indicating that while the members are included inline for convenience, clients should still prefer using the /GroupMembers endpoint for management operations.

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:Group"],
  "id": "a0b1c2d3-f08f-4139-944c-2e6949b80b05",
  "displayName": "Sales Team",
  "members": [
    {
      "value": "2819c223-7f76-453a-919d-413861904646",
      "display": "Babs Jensen"
    }
  ],
  "urn:ietf:params:scim:schemas:extension:groupMembers:2.0: \
      Group:membersMetadata": {
    "policy": "hybrid",
    "ref": "https://example.com/scim/v2/GroupMembers?filter= \
      group.value%20eq%20%22a0b1c2d3-f08f-4139-944c-2e6949b80b05%22",
    "memberCount": 1,
    "allowedMemberTypes": ["User", "Group"]
  },
  "meta": {
    "resourceType": "Group",
    "location": "https://example.com/scim/v2/Groups/ \
      a0b1c2d3-f08f-4139-944c-2e6949b80b05"
  }
}
```

6. Managing GroupMember Resources

This section describes how GroupMember resources are managed using the SCIM protocol. A GroupMember is a simple resource that represents a linkage between a group and a member. As such, a membership can only be created, retrieved, or deleted. Updating a membership serves little practical value, as changing the group or the member would fundamentally represent a new membership, not a modification of the existing one. Therefore, a Service Provider that supports this specification MUST only support the POST, GET, and DELETE methods for this resource type.

Service Providers MAY also support management of group members through the existing members attribute of the Group resource as defined in [RFC7643] for the purpose of backwards compatibility with existing clients. However, when adding or removing members from a group that also has GroupMember resources, Service Providers MUST ensure that the state remains consistent across both representations. For example, deleting a GroupMember resource MUST result in the corresponding member being removed from the members array on the Group resource, if that attribute is supported by the Service Provider.

6.1. Creating GroupMember Resources (POST)

To add a new member to a group, the client sends a POST request to the GroupMembers endpoint. The request body MUST contain a GroupMember resource, specifying the group.value and member.value.

* Request: POST /scim/v2/GroupMembers

* Response: 201 Created with the full GroupMember resource in the body, including its newly generated id and meta attributes.

A Service Provider MUST ensure that both the group and the member referenced by their ids exist before creating the GroupMember resource. If either the group or the member does not exist, the Service Provider SHOULD return a 400 Bad Request error with a scimType of invalidValue.

If the membership already exists, the Service Provider MUST return a 409 Conflict error.

Example Request Body:

POST /GroupMembers

```
{
  "schemas": ["urn:ietf:params:scim:schemas:core:2.0:GroupMember"],
  "group": {
    "value": "e9e30dba-f08f-4139-944c-2e6949b80b05"
  },
  "member": {
    "value": "2819c223-7f76-453a-919d-413861904646"
  }
}
```

6.2. Retrieving GroupMember Resources (GET)

GroupMember resources can be retrieved by sending a GET request to the GroupMembers endpoint. Clients can retrieve an individual resource by its id or a list of resources.

* To get a specific membership: GET /scim/v2/GroupMembers/{id}

* To get all memberships: GET /scim/v2/GroupMembers

6.2.1. Pagination

Service Providers MUST support pagination of GroupMember resources to allow clients to retrieve large sets of memberships in manageable chunks.

Index-based Pagination: The startIndex and count query parameters are the primary method for pagination, as defined in [RFC7644].

* Example: GET /scim/v2/GroupMembers?startIndex=1&count=1000

Cursor as defined in [RFC9865] for improved performance with very large data sets.

* Example: GET /scim/v2/GroupMembers?count=1000&cursor=aW5kZXg9MTAx

The response for a paginated request is a ListResponse containing the GroupMember resources for the current page.

6.2.2. Filtering

Service Providers MUST support filtering on the group.value and member.value attributes. This enables clients to perform critical queries, such as "find all members of a specific group" or "find all groups a specific user is a member of."

To find all members of a group:
GET /scim/v2/GroupMembers?filter=group.value eq e9e30dba-f08f-4139-944c-2e6949b80b05"

To find all groups for a member:
GET /scim/v2/GroupMembers?filter=member.value eq "2819c223-7f76-453a-919d-413861904646"

6.3. Deleting GroupMember Resources (DELETE)

To remove a member from a group, the client sends a DELETE request to the URI of the specific GroupMember resource.

* Request: DELETE /scim/v2/GroupMembers/{id}

* Response: 204 No Content on successful deletion.

6.4. Bulk Operations

Clients can create and delete multiple GroupMember resources in a single request using the /Bulk endpoint as defined in [RFC7644]. This is highly efficient for synchronizing memberships for a group with many changes.

The following is an example of a Bulk request that adds two new members and removes one existing member from a group.

Example Bulk Request:

```
{
"schemas": ["urn:ietf:params:scim:api:messages:2.0:BulkRequest"],
"failOnErrors": 1,
"Operations": [
{
  "method": "POST",
  "path": "/GroupMembers",
  "bulkId": "add-user-1",
  "data": {
    "schemas": ["urn:ietf:params:scim:schemas:core:2.0:GroupMember"],
    "group": { "value": "e9e30dba-f08f-4139-944c-2e6949b80b05" },
    "member": { "value": "aed9876f-e83c-4359-99a3-37e082236081" }
  }
},
{
  "method": "POST",
  "path": "/GroupMembers",
  "bulkId": "add-user-2",
  "data": {
    "schemas": ["urn:ietf:params:scim:schemas:core:2.0:GroupMember"],
    "group": { "value": "e9e30dba-f08f-4139-944c-2e6949b80b05" },
    "member": { "value": "bce5231a-6d36-4b89-a249-1b913e16338b" }
  }
},
{
  "method": "DELETE",
  "path": "/GroupMembers/gm12345",
  "bulkId": "delete-user-3"
}
]
}
```

7. Service Provider Considerations

This section describes the requirements for Service Providers that implement the GroupMember resource.

7.1. Discovering Support for the GroupMember Resource

Service Providers that support the GroupMember resource MUST declare this support in their ResourceType and Schema metadata.

7.1.1. Schema Endpoint

The Service Provider’s Schema definition, available at the /Schemas endpoint, MUST include the full schema definitions for urn:ietf:params:scim:schemas:core:2.0:GroupMember as defined in Section 2.3 of this document.

7.1.2. Impact on the Group Resource

As noted in Section 3, a Service Provider MAY continue to support the members attribute on the Group resource for backwards compatibility. When doing so, the Service Provider MUST maintain transactional integrity and consistency between the state of the members attribute and the state of the corresponding GroupMember resources.

For example, if a DELETE request to a /GroupMembers/{id} URI is successful, the corresponding member MUST also be removed from the members array of the parent Group resource. Conversely, if a member is removed from a Group via a PATCH request to the /Groups/{id} URI, the corresponding GroupMember resource MUST be deleted.

Service Providers that support both mechanisms SHOULD clearly document their consistency model. It is RECOMMENDED that for groups with a very large number of members, Service Providers implement the members attribute as write-only by setting the ’returned’ schema property to ’never’.

8. Schema Representation

8.1. GroupMember Core Schema

The following is the formal SCIM schema definition for the GroupMember resource.

```
{
  "id": "urn:ietf:params:scim:schemas:core:2.0:GroupMember",
  "name": "GroupMember",
  "description": "SCIM resource representing a single
      group membership.",
  "attributes": [
    {
      "name": "group",
      "type": "complex",
```

```
        "mutability": "immutable",
        "required": true,
        "uniqueness": "none",
        "description": "The group in which the member is a member.",
        "subAttributes": [
          {
            "name": "value",
            "type": "string",
            "mutability": "immutable",
            "required": true,
            "uniqueness": "none",
            "description": "The id of the group."
          },
          {
            "name": "$ref",
            "type": "reference",
            "referenceTypes": ["Group"],
            "mutability": "readOnly",
            "required": false,
            "uniqueness": "none",
            "description": "The URI of the group."
          }
        ]
      },
      {
        "name": "member",
        "type": "complex",
        "mutability": "immutable",
        "required": true,
        "uniqueness": "none",
        "description": "The member of the group.",
        "subAttributes": [
          {
            "name": "value",
            "type": "string",
            "mutability": "immutable",
            "required": true,
            "uniqueness": "none",
            "description": "The id of the member."
          },
          {
            "name": "$ref",
            "type": "reference",
            "referenceTypes": ["User", "Group"],
            "mutability": "readOnly",
            "required": false,
            "uniqueness": "none",
            "description": "The URI of the member."
```

```
          },
          {
            "name": "type",
            "type": "string",
            "mutability": "readOnly",
            "required": false,
            "uniqueness": "none",
            "description": "The type of the member,
              e.g., 'User' or 'Group'."
          }
        ]
      }
    ]
  }
```

8.2. membersMetadata Schema Extension

This specification defines a schema extension for the SCIM Group resource to support the discoverability of membership management policies.

Schema URN:
urn:ietf:params:scim:schemas:extension:groupMembers:2.0:Group

8.2.1. The membersMetadata Attribute Definition

The extension introduces a single complex attribute to the Group resource: membersMetadata. This attribute is defined as follows:

```
{
  "id": "urn:ietf:params:scim:schemas:extension:groupMembers:2.0:Group",
  "name": "GroupMembersMetadata",
  "description": "A schema extension for Group resources to provide
    metadata about how members are managed.",
  "attributes": [
    {
      "name": "membersMetadata",
      "type": "complex",
      "mutability": "readOnly",
      "required": false,
      "description": "Provides metadata about the management of
        this group's members.",
      "subAttributes": [
        {
          "name": "policy",
          "type": "string",
          "mutability": "readOnly",
          "required": true,
```

```
          "canonicalValues": [
            "inline",
            "external",
            "hybrid"
          ],
          "description": "Specifies the policy for how membership
            of this group is represented."
        },
        {
          "name": "ref",
          "type": "reference",
          "referenceTypes": ["uri"],
          "mutability": "readOnly",
          "required": true,
          "description": "A URI that a client can use to query for
            the group's members."
        },
        {
          "name": "memberCount",
          "type": "integer",
          "mutability": "readOnly",
          "required": false,
          "description": "An integer indicating the total number
            of members in the group."
        },
        {
          "name": "allowedMemberTypes",
          "type": "string",
          "mutability": "readOnly",
          "required": false,
          "multiValued": true,
          "description": "A list of strings that specify the resource
            types of members allowed in this group."
        }
      ]
    }
  ]
}
```

8.2.2. Usage

When a Service Provider uses this extension, it MUST add the schema URN urn:ietf:params:scim:schemas:extension:group:2.0:membersMetadata to the schemas attribute of the Group resource. The membersMetadata attribute and its sub-attributes are read-only, as they are metadata reported by the Service Provider to the client.

Example schemas attribute in a Group resource:

```
{
  "schemas": [
    "urn:ietf:params:scim:schemas:core:2.0:Group",
    "urn:ietf:params:scim:schemas:extension:groupMembers:2.0: \
        Group:membersMetadata"
  ],
  "id": "e9e30dba-f08f-4139-944c-2e6949b80b05",
  "displayName": "All Employees",
  "urn:ietf:params:scim:schemas:extension:groupMembers:2.0: \
        Group:membersMetadata": {
    "policy": "external",
    "ref": "https://example.com/scim/v2/GroupMembers?filter= \
      group.value%20eq%20%22e9e30dba-f08f-4139-944c-2e6949b80b05%22",
    "memberCount": 150321
  }
}
```

8.3. Security Considerations

The security considerations for the GroupMember resource are substantially the same as those for the User and Group resources defined in Section 8 of the SCIM Protocol document [RFC7644]. All requests MUST be made over a secure channel such as Transport Layer Security (TLS).

Authentication and authorization for managing GroupMember resources are the responsibility of the Service Provider. Implementers should consider the following:

* Access controls for GroupMember resources may be inherited from the parent Group. For example, a client that has permission to view a Group and its members should also have permission to GET the corresponding GroupMember resources.

* A client authorized to add or remove members from a Group (e.g., via a PATCH to the Group resource) should have equivalent permissions to POST and DELETE GroupMember resources for that same group.

* When a client attempts to retrieve one or more GroupMember resources, whether through a direct GET to a resource URI (/GroupMembers/{id}) or through a list request to the endpoint (/GroupMembers), the Service Provider’s authorization decision MUST be based on the client’s permission to read the parent Group. A client that has permission to read a Group resource MUST also be granted permission to retrieve any GroupMember resource where the group.value attribute matches the id of that Group. This policy mirrors the existing SCIM behavior where access to a Group

resource implies access to its member list. A Service Provider MUST NOT require an additional authorization check on the member resource itself as a condition for retrieving a GroupMember resource.

8.4. IANA Considerations

This document requests that IANA register a new URN in the "SCIM Schemas" registry.

URI: urn:ietf:params:scim:schemas:core:2.0:GroupMember *Specification:* This document *Description:* Defines the schema for a resource representing a single group membership.

9. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7643] Hunt, P., Ed., Grizzle, K., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Core Schema", RFC 7643, DOI 10.17487/RFC7643, September 2015, <https://www.rfc-editor.org/info/rfc7643>.

[RFC7644] Hunt, P., Ed., Grizzle, K., Ansari, M., Wahlstroem, E., and C. Mortimore, "System for Cross-domain Identity Management: Protocol", RFC 7644, DOI 10.17487/RFC7644, September 2015, <https://www.rfc-editor.org/info/rfc7644>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC9865] Peterson, M., Ed., Zollner, D., and A. Sehgal, "Cursor-Based Pagination of System of Cross-domain Identity Management (SCIM) Resources", RFC 9865, DOI 10.17487/RFC9865, October 2025, <https://www.rfc-editor.org/info/rfc9865>.

Author’s Address

Danny Zollner
Okta
Email: danny.zollner@okta.com