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

D. Zollner
Okta

2 March 2026

SCIM 2.0 Interoperability Profile
draft-zollner-scim-interop-profile-00

Abstract

This document defines an implementation profile for the System for Cross-domain Identity Management (SCIM) 2.0. The goal of this profile is to increase interoperability between identity providers and service providers by reducing the number of optional features and providing clear guidance on implementing a common subset of the SCIM standard. It deprecates certain features that have proven to be problematic for interoperability or are considered insecure.

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.

Copyright Notice

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-interop-profile.

2. Introduction

The SCIM 2.0 standard RFC7643 [RFC7644] provides a powerful and flexible framework for automating user provisioning. However, its flexibility, with numerous optional features, attributes, and protocol variations, has led to significant interoperability challenges. Implementers are often faced with a wide array of choices, resulting in bespoke integrations that are costly to build and maintain.

This document specifies a profile for SCIM 2.0 to address these challenges. It provides a baseline of required features and deprecates others to ensure that implementations conforming to this profile can interoperate seamlessly.

The target audience for this profile includes developers of identity providers (IDPs) and service providers (SPs) who wish to build conformant and interoperable SCIM clients and servers.

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. Scope and Conformance

This profile applies to SCIM 2.0 Service Providers and Clients.

A Service Provider is conformant with this profile if it implements all the "MUST" and "REQUIRED" features defined herein.

A Client is conformant with this profile if it is capable of interacting with a conformant Service Provider.

Implementations claiming conformance to this profile should indicate so in their ServiceProviderConfig response.

5. Data Model Requirements

5.1. Supported Resource Types

Conformant Service Providers MUST implement the following resource types and their corresponding endpoints:

* User ([RFC7643], Section 4.1)

Service Providers MUST publish an accurate list of schemas and attributes via the /Schemas endpoint, matching exactly what is implemented and supported by the service.

Conformant Service Providers MAY implement the following resource types and their corresponding endpoints:

* Group ([RFC7643], Section 4.2)

The following configuration discovery-related resource types and their corresponding endpoints MUST be implemented:

* ServiceProviderConfig ([RFC7643], Section 4)

* Schema ([RFC7643], Section 7)

* ResourceType ([RFC7643], Section 6)

5.2. Attribute Requirements

This section will define the minimal set of attributes that MUST be supported for the User and Group resources to ensure a baseline level of interoperability.

5.2.1. User Attributes

To ensure a functional baseline for user provisioning, Service Providers *MUST* support the following attributes for the User resource:

* userName

* active

* displayName

* name.givenName

* name.familyName

The password attribute is deprecated and *MUST NOT* be implemented. Service Providers *SHOULD NOT* store user passwords and should rely on other authentication methods, such as federation via SAML or OpenID Connect, to authenticate users.

5.2.2. Group Attributes

Service Providers MUST support both the displayName and members attributes. All group resources MUST contain a value for displayName. Service Providers MUST allow groups to be created without any members.

5.3. Case Sensitivity

To ensure predictable and interoperable behavior, Service Providers *MUST* implement case sensitivity consistently across both filtering operations and uniqueness constraint enforcement. For any given attribute, the case sensitivity rules applied during a filter query (e.g., userName eq "User.A") *MUST* be identical to the rules used to detect a uniqueness conflict during a POST or PATCH operation.

Furthermore, this profile requires adherence to the case sensitivity definitions specified in [RFC7643] for the following common attributes: * userName: caseExact is false. Service Providers *MUST* treat JSmith and jsmith as equivalent. * externalId: caseExact is true. Service Providers *MUST* treat ABC-123 and abc-123 as distinct values.

5.4. Attribute and Schema Handling

To ensure strict conformance and prevent unintended data loss or corruption, Service Providers *MUST* adhere to a strict handling model for attributes and schema extensions. When a SCIM request (e.g., POST, PUT, PATCH) contains attributes or schema URIs that are not defined in the Service Provider’s ResourceType or Schema definitions, the request *MUST* be rejected.

The Service Provider *MUST* return an HTTP 400 Bad Request with a scimType error of invalidSyntax for such requests.

5.5. Canonical Values for Typed Attributes

For multi-valued attributes that include a type sub-attribute (e.g., emails, phoneNumbers, ims, photos), Clients *MUST* use the canonical type values defined in [RFC7643] (e.g., "work", "home", "other"). Service Providers *MUST* treat these canonical values as case-insensitive.

6. Protocol and Endpoint Requirements

6.1. Data Format and HTTP Headers

All data exchange MUST use the JSON format as defined in [RFC7643], and all requests and responses containing SCIM data MUST use the Content-Type header with the value application/scim+json as defined in [RFC7644], Section 8.1. The XML data format is explicitly out of scope for this profile and MUST NOT be used.

To aid in troubleshooting and client identification, SCIM clients MUST include a User-Agent header in all HTTP requests. The header's value should be meaningful, for example, identifying the name of the client software.

6.2. Filtering

The filter query parameter MUST be supported. Service Providers MUST support the eq and and operators. To promote interoperability, clients MUST NOT use operators other than eq and and.

The use of filters in the URI for any HTTP method other than GET (e.g., PATCH /Users?filter=...) is deprecated and *MUST NOT* be used.

Service Providers MUST support filtering on the following User attributes:

* 'userName'
* 'emails.value'
* 'emails.type'
* 'externalId'

Service Providers MUST support filtering on the following Group attributes:

* 'displayName'
* 'members.value'
* 'externalId'

6.3. Pagination

To ensure the reliable handling of large data sets, service providers MUST implement at least one of the following pagination methods for all list operations:

* Index-based pagination as defined in [RFC7644], Section 3.4.2.4.

* Cursor-based pagination, as defined in [RFC9865].

If the count parameter is omitted from a request, Service Providers SHOULD return at least 100 results by default. Service Providers MUST support a client-requested count value of at least 250. Service Providers SHOULD enforce a server-specified maximum number of results per page and MUST return fewer results than requested when the client specifies a count value that exceeds that limit.

6.4. Updating Resources

Service Providers MUST support the PATCH operation ([RFC7644], Section 3.5.2) for resource updates. Clients MUST use the PATCH operation for updates and SHALL NOT use the PUT operation.

This profile defines a restricted subset of the SCIM 2.0 PATCH method to ensure predictable behavior and high interoperability.

6.4.1. General PATCH Constraints

6.4.1.1. Mandatory Use of the ’path’ Attribute

Every operation object within the Operations array *MUST* contain a path attribute. Clients *MUST NOT* issue "path-less" PATCH operations where the target attribute is implied by the keys within the value object.

Service Providers *MUST* reject PATCH requests containing operations that lack a path attribute with an HTTP 400 Bad Request and a scimType error of invalidSyntax.

6.4.1.2. Multi-Attribute Updates

When a Client needs to update multiple attributes in a single HTTP request, it *MUST* provide a separate operation entry within the Operations array for each unique attribute path.

6.4.2. Attribute-Specific Requirements

6.4.2.1. Singular Attributes (Simple and Complex)

Simple Attribute Operation Equivalence
: For singular simple attributes (e.g., userName, active), Service Providers *MUST* treat add and replace as functionally equivalent.

Complex Sub-attribute Targeting
: When only intending to modify the value of a specific sub-attribute of a complex attribute, Clients *SHOULD* target that sub-attribute using dot-notation (e.g., path: "name.givenName").

Complex Attribute Operations
: The following rules apply to each PATCH operation type:

  Replace
  : If a Client targets a singular complex attribute in its entirety (e.g., path: "name") using the replace operation, the value *MUST* be a JSON object containing all sub-attributes the Client intends to persist.

  Add
  : If a Client targets a singular complex attribute in its entirety using the add operation, the value *MUST* be a JSON object. The Service Provider *MUST* perform a partial merge.

6.4.2.2. Multi-valued Attributes (Simple and Complex)

Collection-Level Operations
: The following rules apply to each PATCH operation type:

  Replace
  : If a Client targets a multi-valued attribute path without a filter using the replace operation, the value *MUST* be an array.

  Add
  : If a Client targets a multi-valued attribute path without a filter using the add operation, the value *MUST* be an array.

Filtering Constraints
: When using a filter to target an element within a multi-valued complex attribute:

  Sub-attribute Requirement
  : The path *MUST* target a specific sub-attribute.

Prohibition
Targeting the element object itself is *PROHIBITED*.

6.4.3. Error Handling

Service Providers *MAY* return an HTTP 400 Bad Request for any PATCH operation that violates these constraints. Specific scimType values should be used as follows: * invalidSyntax: Missing path or use of prohibited operators. * invalidFilter: Filter matches more than one element in a multi-valued attribute. * invalidPath: Filter fails to target a specific sub-attribute.

6.5. Resource Lifecycle

Service Providers *MUST NOT* treat a PATCH request that sets the active attribute to false as a deletion of the resource. A disabled user is considered inactive and should be excluded from authentication and normal access, but the user object itself *MUST* be retained by the Service Provider. Deletion of a resource can only be performed via an HTTP DELETE request.

6.6. Concurrency and Versioning

SCIM clients *MUST NOT* include HTTP headers related to conditional requests or entity tags (ETags), such as If-Match, If-None-Match, If-Modified-Since, or If-Unmodified-Since. Service Providers are not expected to support these headers and *MAY* ignore them or reject the request. This profile relies on the principle of "last write wins" for simplicity.

6.7. Bulk Operations

Support for the /Bulk endpoint ([RFC7644], Section 3.7) is RECOMMENDED butOPTIONAL.

6.8. Error Handling

6.8.1. Uniqueness Conflicts

When a POST or PATCH request attempts to create or modify a resource in a way that violates a uniqueness constraint (e.g., for attributes like userName or externalId), the Service Provider *MUST* return an HTTP 409 Conflict response. The response body *MUST* also include a SCIM error detail with the scimType set to uniqueness, as defined in [RFC7644], Section 3.12.

7. Security Considerations

7.1. Transport Security

All communication between a SCIM client and service provider MUST be secured using Transport Layer Security (TLS) version 1.2 [RFC5246] or a later version.

7.2. Authentication

The use of HTTP Basic Authentication over TLS is NOT RECOMMENDED.

8. IANA Considerations

This document has no IANA actions.

9. Acknowledgements

(TODO: Add acknowledgements)

10. 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>.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, DOI 10.17487/RFC5246, August 2008, <https://www.rfc-editor.org/info/rfc5246>.

[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