SCIM
Internet-Draft
Intended status: Standards Track
Expires: 17 October 2026

D. Zollner
Okta

15 April 2026

SCIM 2.0 Interoperability Profile
draft-zollner-scim-interop-profile-01

Abstract

This document defines an implementation profile for the System for Cross-domain Identity Management (SCIM) 2.0. In the typical deployment model, an identity provider acting as a SCIM Client provisions and manages identities at multiple downstream service providers, while each service provider (commonly a multi-tenant application serving many customers) accepts provisioning connections from multiple different identity providers. This many-to-many integration model compounds the interoperability challenges arising from the wide range of optional features and implementation variations permitted by the base specification. This profile addresses those challenges by restricting the optional features and protocol variations permitted under the base specification and by establishing normative requirements for a common implementation baseline.

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 17 October 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 framework for automating identity provisioning across organizational boundaries. While the base specification’s flexibility enables broad applicability, it also permits a wide range of implementation choices — in areas such as PATCH syntax, pagination, filtering, case sensitivity, and error handling — that have proven to be common sources of interoperability failure in practice. Where implementations diverge on these points, the result is integrations that require bespoke configuration and maintenance for each pairing.

This document specifies a profile for SCIM 2.0 that addresses these challenges by establishing required capabilities and restricting protocol variations that have proven problematic. Implementations conforming to this profile can be expected to interoperate without integration-specific customization.

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.

Requirements in this document, expressed using the normative terms "MUST", "SHALL", and "REQUIRED" as defined in the Notational Conventions section above, are addressed to Service Providers, Clients, or both. A Service Provider is conformant with this profile if it satisfies all such requirements addressed to Service Providers and all such requirements addressed to both roles. A Client is conformant with this profile if it satisfies all such requirements addressed to Clients and all such requirements addressed to both roles.

Service Providers claiming conformance with this profile MUST include an interopProfileConformant attribute with a value of true in their ServiceProviderConfig response. This attribute is defined as follows:

* Name: interopProfileConformant

* Type: Boolean

* Multi-Valued: false

* Required: false

* Mutability: readOnly

* Returned: default

A Service Provider that does not conform to this profile MUST either omit this attribute or set its value to false.

5. Data Model Requirements

5.1. Discovery Endpoints

Service Providers MUST implement the following configuration discovery endpoints:

* ServiceProviderConfig ([RFC7643], Section 4)

* Schema ([RFC7643], Section 7)

* ResourceType ([RFC7643], Section 6)

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. All schemas referenced in resource type definitions returned by the /ResourceTypes endpoint MUST be available at the /Schemas endpoint.

The schema and resource type definitions for ServiceProviderConfig, Schema, and ResourceType MAY be omitted from the /Schemas and /ResourceTypes endpoints respectively, as these are configuration resources rather than provisioning targets.

For every ResourceType resource returned by the /ResourceTypes endpoint, Service Providers MUST populate the id attribute and its value MUST equal the value of the name attribute.

5.2. 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.

This profile requires adherence to the case sensitivity definitions specified in [RFC7643] for the following attributes:

Common Attributes ([RFC7643], Section 3.1): * id: caseExact is true. Service Providers *MUST* treat abc-123 and ABC-123 as distinct values. * externalId (optional): caseExact is true. Service Providers *MUST* treat ABC-123 and abc-123 as distinct values, if externalId is supported.

User Schema Attributes ([RFC7643], Section 4.1.1) -- applicable only if the User resource type is implemented: * userName: caseExact is false. Service Providers *MUST* treat JSmith and jsmith as equivalent.

5.3. 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, 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.

5.4. Canonical Values for Typed Attributes

For multi-valued complex attributes that include a type sub-attribute (e.g., emails, phoneNumbers, ims, addresses), Service Providers MUST declare the acceptable type values for each such attribute in that attribute’s canonicalValues property, as returned by /Schemas. Clients MUST only use type values listed in the canonicalValues property for that attribute.

6. Protocol and Endpoint Requirements

6.1. Endpoint Structure

Service Providers MUST offer a unique endpoint for each implemented resource type (e.g., /Users, /Groups). Resources of different types MUST NOT share an endpoint. This requirement applies only to resource-type endpoints; the root endpoint (e.g., https://example.com/scim/v2/) is not subject to this requirement and MAY return resources of multiple types if querying at the root level is supported.

Service Providers MUST use /{endpoint}/{id} as the canonical URI for addressing any resource and MUST NOT address resources via other attribute values in the URI path (e.g., /{userName}). Clients MUST use /{id} when retrieving a known resource, and MUST use the filter query parameter to locate a resource by any other attribute value.

6.2. Data Format and HTTP Headers

All data exchange MUST use the JSON format as defined in [RFC7643]. All requests and responses containing SCIM data MUST include a Content-Type header with the value application/scim+json, as defined in [RFC7644], Section 8.1.

To aid in troubleshooting and client identification, 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.

6.3. 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=...) *MUST NOT* be used.

When a filter expression is applied to a resource-type endpoint (e.g., /Users, /Groups) and references an attribute not defined in any of that resource type’s schemas, Service Providers MUST return an HTTP 400 error with a scimType of invalidFilter, as defined in [RFC7644], Section 3.12.

6.4. 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 100. 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.

6.5. 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.

6.5.1. General PATCH Constraints

6.5.1.1. 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.

6.5.1.2. 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.

6.5.2. Attribute-Specific Requirements

6.5.2.1. Singular Attributes (Simple and Complex)

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

2. *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").

3. *Complex Attribute Operations:*

    * *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. The Service Provider *MUST* replace the entire complex attribute with the provided object.

    * *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, updating only the provided sub-attributes and preserving existing values for omitted sub-attributes.

6.5.2.2. Multi-valued Attributes (Simple and Complex)

This section applies to both multi-valued simple attributes (e.g., the schemas attribute) and multi-valued complex attributes (e.g., emails, addresses).

1. *Collection-Level Operations:*

    * *Replace:* If a Client targets a multi-valued attribute path without a filter (e.g., path: "emails" or path: "roles") using the replace operation, the value *MUST* be an array of objects/values. The Service Provider *MUST* replace the entire collection with the provided array.

    * *Add:* If a Client targets a multi-valued attribute path without a filter using the add operation, the value *MUST* be an array of objects/values. The Service Provider *MUST* append the provided values to the existing collection.

2. *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 of the matched element (e.g., path: "emails[type eq \"work\"].value"). Targeting the element object itself (e.g., path: "emails[type eq \"work\"]") is *PROHIBITED*.

6.5.3. 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.

6.6. Resource Lifecycle

Service Providers *MUST NOT* treat a PATCH request as a deletion of the resource. A resource modified via PATCH *MUST* be retained by the Service Provider. Deletion of a resource can only be performed via an HTTP DELETE request.

Service Providers MUST return 404 Not Found in response to any operation targeting a deleted resource and MUST omit deleted resources from all query results.

After a resource is successfully deleted, its unique identifiers (such as userName) MUST NOT be considered in uniqueness conflict calculations and MUST be available for reassignment to a new resource.

Service Providers MUST NOT respond to an HTTP DELETE request by modifying resource attributes rather than deleting the resource (e.g., setting a User’s active attribute to false). A successful DELETE request MUST result in the resource being deleted.

6.7. Concurrency and Versioning

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.

6.8. Bulk Operations

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

6.9. Error Handling

6.9.1. 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 emails), 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 Client and Service Provider MUST be secured using Transport Layer Security (TLS) [RFC8446]. Implementations MUST support TLS 1.3 and MAY support TLS 1.2.

7.2. Authentication

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

8. IANA Considerations

Prior to being published as an RFC, this document requests that the IANA SCIM Server-Related Schema URIs registry entry for urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig be updated to include or reference the interopProfileConformant attribute defined in Section 4 of this document.

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>.

[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>.

[RFC8446] Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.3", RFC 8446, DOI 10.17487/RFC8446, August 2018, <https://www.rfc-editor.org/info/rfc8446>.

[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