

Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 10, 2015

M. Ansari, Ed.
Cisco
P. Hunt
Oracle
March 9, 2015

SCIM Soft Delete
draft-ansari-scim-soft-delete-00

Abstract

The System for Cross-Domain Identity Management (SCIM) specification is an HTTP based protocol that makes managing identities in multi-domain scenarios easier to support through a standardized HTTP service.

Among other operations, SCIM defines delete operation where upon successful completion of the call, the SCIM endpoint is supposed to delete the requested object and the object should not be available for future SCIM calls and not used in uniqueness criteria requirements.

While this model is sufficient for a number of SCIM implementations, there are cases this simple definition of delete may not meet product or business requirements. For example a service provider may require a user object to continue to exist as other objects/data is linked with it or for billing purposes, etc. For example a cloud file storage mechanism may require to show basic information about who created a given file or modified one even if the user is de-provisioned from the system.

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 <http://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 September 10, 2015.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction and Overview	2
1.1. Notational Conventions	3
1.2. Definitions	4
2. Soft Delete	4
2.1. Soft Delete Schema Extension	4
2.2. ServiceProviderConfig Extension	5
2.3. SCIM Create	5
2.4. SCIM Retrieve	5
2.5. SCIM Modify	5
2.6. SCIM Delete	5
2.7. SCIM Bulk	6
2.8. SCIM Undelete	6
2.9. SCIM Hard Delete	6
3. Security Considerations	6
4. IANA Considerations	6
5. References	7
5.1. Normative References	7
5.2. Informative References	7
Appendix A. Contributors	7
Appendix B. Acknowledgments	7
Appendix C. Change Log	7
Authors' Addresses	7

1. Introduction and Overview

The System for Cross-Domain Identity Management (SCIM) specification is an HTTP based protocol that makes managing identities in multi-domain scenarios easier to support through a standardized HTTP service.

For some services, even after a resource has been "deleted" from the identity system, there are many artifacts that remain in the application/services layer that were created/touched by the deleted user. Such objects need to remain connected to the deleted object and provide basic information. For example if a user created a document, post, event, even after the user is removed, other users of the system may still interact with these objects and need to see who created the object even if the user is no longer part of the system. Another use case is to protect against accidental loss of references in case of a mistaken "delete" of a resource. Once a resource is removed from the identity system, all that resource's references to any data type is lost given the id of the resource will not be recycled. Recreating the same resource is not going to revive its id and essentially creates a new instance with a new id.

While SCIM delete operation as defined in Section 3.6 of [I-D.ietf-scim-api] allows a service provider to keep a resource after deletion, it does not provide any additional guidance or specification on how the "deleted" resource will be managed beyond the initial delete in cases the service provider does not want to permanently remove a resource.

This specification defines a set of extensions to SCIM API [I-D.ietf-scim-api] to allow additional operations on resources that have been soft deleted:

- o Query for resources that have been soft deleted
- o Hard delete a soft deleted resource
- o Undelete a soft deleted resource
- o Extension to the ServiceProviderConfig [I-D.ietf-scim-core-schema] to allow discovery of softDelete extensions

1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119]. These keywords are capitalized when used to unambiguously specify requirements of the protocol or application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

For purposes of readability examples are not URL encoded. Implementers MUST percent encode URLs as described in Section 2.1 of [RFC3986].

Throughout this documents all figures MAY contain spaces and extra line-wrapping for readability and space limitations. Similarly, some URI's contained within examples, have been shortened for space and readability reasons.

1.2. Definitions

Soft Deleted Resources A SCIM resource that has been deleted using standard SCIM DELETE operation

2. Soft Delete

A SCIM endpoint supporting Soft Delete extensions MUST implement SCIM DELETE operation in such a way that the resource being deleted is not permanently deleted and stored in an alternate "soft deleted" state. Other standard SCIM operations will continue to function as if soft deleted resources do not exist in the system. READ, MODIFY, PATCH, BULK requests with the soft deleted resource id MUST result in a HTTP NOT FOUND (404). A create request for a user resource with a userName that has been soft deleted, MUST NOT fail with an HTTP status 409 due to the userName conflict with the soft deleted record.

2.1. Soft Delete Schema Extension

Any SCIM resource type that supports soft delete extensions MUST extend the schema of the resource type by adding the extension defined in this section.

The following Singular Attributes are defined:

isSoftDeleted

A Boolean attribute set to "true" for any resource that is soft deleted. No value or "false" means the resource is not soft deleted. This attribute has mutability of "readOnly"

softDeleted

A DateTime attribute set to the time the resource was softDeleted. This attribute has mutability of "readOnly". This attribute should be deleted once a resource is not in soft delete state.

2.2. ServiceProviderConfig Extension

SCIM endpoints that support Soft Delete extensions MUST advertise this support in the ServiceProviderConfig endpoint as defined:

softDelete

A complex type that specifies Soft Delete configuration options. REQUIRED.

supported Boolean value specifying whether the operation is supported. REQUIRED.

2.3. SCIM Create

SCIM Create SHOULD NOT ignore namespace conflicts arising from soft deleted objects. For example if there exists a user resource with userName value of "user1" that have been soft deleted, a create request for userName with the same value should not fail because of userName conflict.

2.4. SCIM Retrieve

SCIM retrieve operations MUST NOT match soft deleted objects unless the request includes a filter with the value of "isSoftDeleted=true". This is the only case where a soft deleted resource can be returned as a result of a retrieve operation.

2.5. SCIM Modify

SCIM modify operation on resources that have been soft deleted MUST result in a HTTP NOT FOUND 404.

2.6. SCIM Delete

SCIM Delete operations on normal resources MUST NOT remove the resource, but put it in the soft deleted state by modifying the resource and setting isSoftDeleted attribute on the resource to "true" and setting the softDeleted timestamp value to the time of the delete operation.

Delete operations on soft deleted resource MUST result in an HTTP NOT FOUND 404 error.

[ToDo]Define reference semantics as resources are soft deleted

2.7. SCIM Bulk

SCIM Bulk operations should follow the semantics defined in this section for regular SCIM operations.error.

2.8. SCIM Undelete

To allow soft deleted resources to be restored to regular state, a SCIM modify operations can be performed with a query parameter of "isSoftDeleted=true" on the resource. The SCIM Endpoint MUST change the state of the resource to reflect the change from soft deleted state back to normal by removing the softDeleted attribute from the resource and setting the isSoftDeleted attribute value to "false".

Furthermore if the process of undeleting the resource results in a namespace conflict, the operation MUST fail and return an HTTP Status 409, with "scimType" error code of "uniqueness".

The SCIM client can optionally provide new attribute values as part of the modify request to resolve the conflicts. For example to undelete a user resource where the userName has been recycled, a modify with a new userName value can be sent to the SCIM endpoint to undelete the user resource by setting the value of userName to the new value to avoid the conflict case.

2.9. SCIM Hard Delete

A soft deleted resource can be permanently deleted by sending a SCIM Delete request with a query parameter of "isSoftDeleted=true". This SCIM endpoint SHOULD permanently remove the resource.

3. Security Considerations

Soft deleted users MUST NOT be allowed to authenticate to the service provider or access any resources. Furthermore soft deleted resources SHOULD NOT be used in authorization decision and act as if those resources do not exist.

[TO BE COMPLETED]

4. IANA Considerations

[TO BE COMPLETED]

5. References

5.1. Normative References

- [I-D.ietf-scim-api]
Hunt, P., Grizzle, K., Ansari, M., Wahlstroem, E., and C. Mortimore, "System for Cross-Domain Identity Management: Protocol", draft-ietf-scim-api-16 (work in progress), March 2015.
- [I-D.ietf-scim-core-schema]
Hunt, P., Grizzle, K., Wahlstroem, E., and C. Mortimore, "System for Cross-Domain Identity Management: Core Schema", draft-ietf-scim-core-schema-17 (work in progress), March 2015.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.
- [RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

5.2. Informative References

- [I-D.ietf-scim-use-cases]
Hunt, P., Khasnabish, B., Nadalin, A., Li, K., and Z. Zeltsan, "SCIM Use Cases", draft-ietf-scim-use-cases-04 (work in progress), March 2015.

Appendix A. Contributors

Appendix B. Acknowledgments

The editor would like to thank the participants in the the SCIM working group for their support of this specification.

Appendix C. Change Log

Draft 00 - MA - First Draft

Authors' Addresses

Morteza Ansari (editor)
Cisco Corporation

Email: morteza.ansari@cisco.com

Phil Hunt
Oracle Corporation

Email: phil.hunt@yahoo.com