SCIM
Internet-Draft
Updates: 7643, 7644 (if approved)
Intended status: Standards Track
Expires: 16 January 2026

M. Peterson, Ed.
Entrust
D. Zollner
Independent
A. Sehgal
Amazon Web Services

15 July 2025

Cursor-based Pagination of SCIM Resources
draft-ietf-scim-cursor-pagination-11

Abstract

This document updates RFC7643 and RFC7644 by defining additional SCIM (System for Cross-Domain Identity Management) query parameters and result attributes to allow use of cursor-based pagination in SCIM service providers that are implemented with existing code bases, databases, or APIs where cursor-based pagination is already well established.

Discussion Venues

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

Discussion of this document takes place on the System for Cross-domain Identity Management Working Group mailing list (scim@ietf.org), which is archived at https://mailarchive.ietf.org/arch/browse/scim/.

Source for this draft and an issue tracker can be found at https://github.com/ietf-scim-wg/draft-ietf-scim-cursor-pagination.

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 16 January 2026.

Table of Contents

1. Introduction

The two common patterns for result pagination are index-based pagination and cursor-based pagination. Rather than attempt to compare and contrast the advantages and disadvantages of competing pagination patterns, this document simply recognizes that SCIM (System for Cross-Domain Identity Management) service providers are commonly implemented as an interoperability layer on top of already existing application codebases, databases, and/or APIs that already have a well established pagination pattern.

Translating from an underlying cursor-based pagination pattern to the index-based pagination defined in Section 3.4.2.4 of [RFC7644] ultimately requires the SCIM service provider to fully iterate the underlying cursor, store the results, and then serve indexed pages from the stored results. This task of "pagination translation" increases complexity and memory requirements for implementing a SCIM service provider, and may be an impediment to SCIM adoption for some applications and identity systems.

This document defines a simple addition to the SCIM protocol that allows SCIM service providers to reuse underlying cursors without expensive translation. Support for cursor-based pagination in SCIM encourages broader cross-application identity management interoperability by encouraging SCIM service provider implementations for applications and identity systems where cursor-based pagination is already well-established.

This document updates RFCs 7643 and 7644 because it adds attributes to existing structures from those documents, as described in this memo in Section 2. These changes are invoked when using the "cursor" parameter when making SCIM search requests using GET or POST methods.

1.1. 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.

1.2. Definitions

This document uses the terms defined in section 1.2 of [RFC7643]

2. Query Parameters and Response Attributes

The following table describes the URL pagination query parameters for requesting cursor-based pagination:

Parameter	Description
cursor	The string value of the nextCursor attribute from a previous result page. The cursor value MUST be empty or omitted for the first request of a cursor-paginated query. This value may only contain characters from the unreserved characters set defined in section 2.3 of [RFC3986].
count	Specifies the desired maximum number of query results per page, e.g., 10. A negative value SHALL be interpreted as "0". A value of "0" indicates that no resource results are to be returned except for "totalResults". When specified, the service provider MUST NOT return more although it MAY return fewer results. If unspecified, the maximum number of returned is set by the service provider.

Table 1: Query Parameters

The following table describes cursor-based pagination attributes returned in a paged query response:

Element	Description
nextCursor	A cursor value string that MAY be used in a subsequent request to obtain the next page of results. Service providers supporting cursor-based pagination MUST include nextCursor in all paged query responses except when returning the last page. nextCursor MUST be omitted from a response only to indicate that there are no more result pages.
previousCursor	A cursor value string that MAY be used in a subsequent request to obtain the previous page of results. Returning previousCursor is OPTIONAL. previousCursor MUST NOT be returned with the first page.

Table 2: Response Attributes

Cursor values are URL-safe strings that are opaque to the client. To retrieve another result page for a query, the client MUST query the same service provider endpoint with all query parameters and values being identical to the initial query with the exception of the cursor value which SHOULD be set to a nextCursor (or previousCursor) value that was returned by the service provider in a previous response.

For example, to retrieve the first 10 Users with userName starting with J, use an empty cursor and set the count to 10:

```
GET /Users?filter=userName%20sw%20J&cursor&count=10
Host: example.com
Accept: application/scim+json
Authorization: Bearer U8YJcYYRMjbGeepD
```

The SCIM service provider in response to the query above returns metadata regarding pagination similar to the following example (actual resources removed for brevity):

```
HTTP/1.1 200 OK
Content-Type: application/scim+json

{
   "totalResults":100,
   "itemsPerPage":10,
   "nextCursor":"VZUTiyhEQJ94IR",
   "schemas":["urn:ietf:params:scim:api:messages:2.0:ListResponse"],
   "Resources":[{
      ...
   }]
}
```

Given the example above, to request the next page or results, use the same query parameters and values except set the cursor to the value of nextCursor (VZUTiyhEQJ94IR):

```
GET /Users?filter=username%20sw%20J&cursor=VZUTiyhEQJ94IR&count=10
Host: example.com
Accept: application/scim+json
Authorization: Bearer U8YJcYYRMjbGeepD
```

The service provider responds with:

```
HTTP/1.1 200 OK
Content-Type: application/scim+json

{
   "totalResults": 100,
   "itemsPerPage": 10,
   "previousCursor: "ze7L30kMiiLX6x",
   "nextCursor": "YkU3OF86Pz0rGv",
   "schemas": ["urn:ietf:params:scim:api:messages:2.0:ListResponse"],
   "Resources":[{
      ...
   }]
}
```

In the example above, the response includes the optional previousCursor indicating that the service provider supports forward and reverse traversal of result pages.

As described in Section 3.4.1 of [RFC7644] service providers should return an accurate value for totalResults which is the total number of resources for all pages. Service providers implementing cursor pagination that are unable to estimate totalResults MAY choose to omit the totalResults attribute.

2.1. Pagination errors

If a service provider encounters invalid pagination query parameters (invalid cursor value, count value, etc), or other error conditions, the service provider SHOULD return the appropriate HTTP response status code and detailed JSON error response as defined in Section 3.12 of [RFC7644].

For HTTP status code 400 (Bad Request) responses, the following detail error types are defined. These error types extend the list of error types defined in section 3.12 of [RFC7644], Table 9: SCIM Detail Error Keyword Values.

scimType	Description	Applicability
invalidCursor	Cursor value is invalid. Cursor value SHOULD be empty to request the first page and set to the nextCursor or previousCursor value for subsequent queries.	GET (Section 3.4.2 of [RFC7644])
expiredCursor	Cursor has expired. Do not wait longer than service provider's cursorTimeout to request additional pages.	GET (Section 3.4.2 of [RFC7644])
invalidCount	Count value is invalid. Count value must be between 0 and service provider's maxPageSize and must value identical count of the initial query.	GET (Section 3.4.2 of [RFC7644])

Table 3: Pagination Errors

2.2. Sorting

If sorting is implemented as described Section 3.4.2.3 of [RFC7644], then cursor-paged results should be sorted.

2.3. Implementing Cursors as the Only Pagination Method

A service provider MAY require cursor-based pagination to retrieve all results for a query by including a nextCursor value in the response even when the query does not include the cursor parameter.

For example:

```
GET /Users
Host: example.com
Accept: application/scim+json
```

The service provider may respond to the above query with a page containing defaultPageSize results and a nextCursor value as shown in the below example (Resources omitted for brevity):

```
HTTP/1.1 200 OK
Content-Type: application/scim+json

{
   "totalResults": 5000,
   "itemsPerPage": 100,
   "nextCursor": "HPq72Pax3JUaNa",
   "schemas": ["urn:ietf:params:scim:api:messages:2.0:ListResponse"],
   "Resources": [{
      ...
   }]
}
```

2.4. Implementing Both Cursors and Index Pagination

When a service provider supports both index-based and cursor-based pagination, clients can use the 'startIndex' or 'cursor' query parameters to request a specific method. Additionally, service providers supporting both pagination methods MUST choose a default pagination method to use when responding to requests that have not specified a pagination query parameter.

Implementers of SCIM service providers that previously supported only index-based pagination and are adding support for cursor-based pagination should use index as the default pagination method to avoid incompatibility with clients that expect index-based pagination behaviors when no pagination query parameters are specified.

SCIM clients can query the service provider configuration (Section 4) endpoint to determine if index-based, cursor-based or both types of pagination are supported and which of these is the default.

3. Querying Resources using HTTP POST

Section 3.4.3 of [RFC7644] defines how clients may execute queries without passing parameters on the URL by using the POST verb combined with the /.search path extension execute. When posting to /.search, the client would pass the parameters defined in Section 2 in the body of the POST request. For example:

```
POST /User/.search
Host: example.com
Accept: application/scim+json
Authorization: Bearer U8YJcYYRMjbGeepD

{
   "schemas": ["urn:ietf:params:scim:api:messages:2.0:SearchRequest"],
   "attributes": ["displayName", "userName"],
   "filter": "displayName sw \"smith\"",
   "cursor": "",
   "count": 10
}
```

Which would return a result containing a nextCursor value which may be used by the client in a subsequent call to return the next page of resources:

```
HTTP/1.1 200 OK
Content-Type: application/scim+json

{
   "totalResults": 100,
   "itemsPerPage": 10,
   "nextCursor": "VZUTiyhEQJ94IR",
   "schemas": ["urn:ietf:params:scim:api:messages:2.0:ListResponse"],
   "Resources": [{
      ...
   }]
}
```

4. Service Provider Configuration

The /ServiceProviderConfig resource defined in Section 4 of [RFC7644] facilitates discovery of SCIM service provider features. A SCIM service provider implementing cursor-based pagination SHOULD include the following additional attribute in JSON document returned by the /ServiceProviderConfig endpoint:

pagination A complex type that indicates pagination configuration options. OPTIONAL. The following sub-attributes are defined:

cursor A Boolean value specifying support of cursor-based pagination. REQUIRED.

index A Boolean value specifying support of index-based pagination. REQUIRED.

defaultPaginationMethod A string value specifying the type of

pagination that the service provider defaults to when the client has not specified which method it wishes to use. Possible values are "cursor" and "index". OPTIONAL.

defaultPageSize Positive integer value specifying the default number of results returned in a page when a count is not specified in the query. OPTIONAL.

maxPageSize Positive integer specifying the maximum number of results returned in a page regardless of what is specified for the count in a query. The maximum number of results returned may be further restricted by other criteria. OPTIONAL.

cursorTimeout Positive integer specifying the minimum number of seconds that a cursor is valid between page requests. Clients waiting too long between cursor pagination requests may receive an invalid cursor error response. No value being specified may mean that there is no cursor timeout or that the cursor timeout is not a static duration. OPTIONAL.

Service providers may choose not to advertise Service Provider Configuration information regarding default pagination method, page size or cursor validity. Clients MUST NOT interpret the lack of published Service Provider Configuration values to mean that no defaults or limits on page sizes or cursor lifetimes exist, or that there is no default pagination method. Service providers may choose not to publish values for the pagination sub-attributes for many reasons. Examples include:

* Service providers containing multiple resource types may have different values set for each resource type.

* Default and maximum page size may be determined by factors besides or in addition to the number of resources returned, such as the size of each resource on the page.

Before using cursor-based pagination, a SCIM client MAY fetch the Service Provider Configuration document from the SCIM service provider and verify that cursor-based pagination is supported.

For example:

```
GET /ServiceProviderConfig
Host: example.com
Accept: application/scim+json
```

A service provider supporting both cursor-based pagination and index-based pagination would return a document similar to the following (full ServiceProviderConfig schema defined in Section 5 of [RFC7643] has been omitted for brevity):

```
HTTP/1.1 200 OK
Content-Type: application/scim+json

{
   "schemas": [
      "urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig"],
      ...

   "pagination": {
      "cursor": true,
      "index": true,
      "defaultPaginationMethod": "cursor",
      "defaultPageSize": 100,
      "maxPageSize": 250,
      "cursorTimeout": 3600
   },

   ...
}
```

5. Security Considerations

This section elaborates on the security considerations associated with the implementation of cursor pagination in SCIM. This document is under the same security and privacy considerations of those described in [RFC7644]. It is imperative that implementers additionally consider the following security aspects to safeguard against both deliberate attacks and inadvertent misuse that may compromise the system’s security posture.

5.1. Threat Model and Security Environment

The threat landscape is characterized by two primary types of actors:

1. Unauthenticated and Authenticated Malicious Actors: These individuals or entities represent a malevolent threat. Their objectives include unauthorized access to data, alteration, or deletion through cursor-enabled queries. They may also seek to deplete service provider resources deliberately, aiming to cause a denial-of-service state, thereby reducing service availability.

2. Authenticated Benign Users: This category includes legitimate users who, due to confusion or a lack of understanding, inadvertently engage in actions that consume service provider resources excessively. Such actions, while not ill-intended, can lead to unintended denial of service by overwhelming the service provider's capacity.

5.2. Confidentiality

To ensure that confidential data remains appropriately secured:

* Implementers MUST ensure that pagination through results sets is strictly confined to the data that the actor's current identity has been authorized to access. This holds true even in cases where the actor has obtained a cursor pertaining to a result set that was generated by a different actor.

* Authorization checks MUST be continuously applied as an actor navigates through the result set associated with a cursor. Under no circumstances should possession of a cursor be interpreted as granting any supplementary access privileges to the actor.

* When possible, service providers SHOULD invalidate all cursors corresponding to an actor immediately following a change in permissions. This ensures that any queries executed post-permission change, utilizing old cursors, will be denied. As an alternative approach, service provider may opt to retain the existing cursors but must ensure that any metadata tied to the result set, such as record counts, is updated to reflect the new permissions accurately.

* In alignment with Section 2, cursor values are URL-Safe strings that are opaque to clients. Server providers should obfuscate cursors values to prevent clients from interpreting cursors or forging new cursors. Service providers should be able to easily detect forged cursor values and immediately return an invalidCursor as described in Section 2.1

* The service provider MUST handle error scenarios without exposing sensitive data. For instance, if an actor attempts to access a page of results outside their authorized scope, or if a request is made for a non-existent page, the service provider should respond with identical error messages, so as not to disclose any details of the underlying data or the nature of the authorization failure. It is acceptable, however, for the service provider to log different messages to a log accessible by administrators or other authorized personnel.

5.3. Availability

The concern for availability primarily stems from the potential for Denial of Service (DoS) attacks. If the service provider elects to retain substantial data or metadata for each cursor, numerous initial queries that allocate cursors could strain and eventually exhaust service provider resources. Such an attack could be orchestrated by an attacker with malicious intent or could occur unintentionally as a result of client testing or bugs.

To mitigate risks, the following strategies are recommended for service providers:

* Clients should authenticate to retrieve large result sets. Anonymous queries yielding numerous results, may return an HTTP status code 400 (Bad Request) with the error type "tooMany," as outlined in [RFC7644] section 3.12.

* Implement rate limiting to control the volume and cadence of cursor requests. This approach should adhere to established standards for rate limiting, details of which can be found in [RFC6585].

* Allow administrator of the service provider to set a ceiling on the number of cursors permissible at any given time or to specify a maxPageSize value. Guidance on configuring such values should be documented in the implementation administrator/installation guide.

* Cursor invalidation mechanisms (including mechanisms triggered by permissions changes) must be designed to be resource-efficient to prevent them from being exploited for DoS attacks.

5.4. Other Security References

Using URIs to describe and locate resources has its own set of security considerations discussed in Section 7 of [RFC3986]. Implementations should also refer to [BCP195] and [RFC9110] for additional security considerations that are relevant for underlying TLS and HTTP protocols.

6. IANA Considerations

This specification requests IANA to amends the SCIM Server-Related Schema URIs registry established by [RFC7643].

For the urn:ietf:params:scim:api:messages:2.0:ListResponse, add Section 2 of this document to the References column.

For the urn:ietf:params:scim:api:messages:2.0:SearchRequest, add Section 2 of this document to the References column.

For the urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig, add Section 4 of this document to the References column.

7. Change Log

RFC Editor: Please remove this section in the release version of the document.

-08

* Fix several typos and wording consistencies
* Add reference to RFC7644 in Security Considerations
* Adjust indenting and wording to clarify the definition of the pagination attribute in serviceProviderConfig
* Reference RFC section 2.3 (not section 2.2) for unreserved characters
* Reference section RFC 7644 3.4.3 (not section 3.4.2.4) for POST query
* Added updates 7644, 7643
* Changed IANA considerations to add sections of this document to References column of SCIM Schema URIs for Data Resources impacted by this document

-07

* Minor grammar change
* Add informative reference to BCP195 and RFC9110

-05

* Various updates in response to WG/IETF Last Call feedback

-04

* Added IANA Considerations section
* Added Security Considerations section

* Added Backwards Compatibility Considerations section

-03

* Minor grammatical/typo fixes, rename + changes to maxPageSize SCP definition

-02

* Typos/semantics, acknowledgements, expansion of cursorTimeout SCP definition

-01

* Updated after Httpdir review.

-00

* Adopted by SCIM WG.

8. Acknowledgments and Contributions

The authors would like to acknowledge the contribution of Paul Lanzi (IDenovate) in leading the writing of security considerations section.

The authors would also like to acknowledge the following individuals who provided valuable feedback while reviewing the document:

* Aaron Parecki - Okta

* David Brossard - Axiomatics

* Dean H. Saxe - Independent

* Pamela Dingle - Microsoft

9. References

9.1. 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/rfc/rfc2119>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/rfc/rfc3986>.

[RFC6585] Nottingham, M. and R. Fielding, "Additional HTTP Status Codes", RFC 6585, DOI 10.17487/RFC6585, April 2012, <https://www.rfc-editor.org/rfc/rfc6585>.

[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/rfc/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/rfc/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/rfc/rfc8174>.

9.2. Informative References

[BCP195] Best Current Practice 195, <https://www.rfc-editor.org/info/bcp195>.
At the time of writing, this BCP comprises the following:

Moriarty, K. and S. Farrell, "Deprecating TLS 1.0 and TLS 1.1", BCP 195, RFC 8996, DOI 10.17487/RFC8996, March 2021, <https://www.rfc-editor.org/info/rfc8996>.

Sheffer, Y., Saint-Andre, P., and T. Fossati, "Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)", BCP 195, RFC 9325, DOI 10.17487/RFC9325, November 2022, <https://www.rfc-editor.org/info/rfc9325>.

[RFC9110] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP Semantics", STD 97, RFC 9110, DOI 10.17487/RFC9110, June 2022, <https://www.rfc-editor.org/rfc/rfc9110>.

Authors' Addresses

Matt Peterson (editor)
Entrust

Email: matt.peterson@entrust.com

Danny Zollner
Independent
Email: danny@zollnerd.com

Anjali Sehgal
Amazon Web Services
Email: anjalisg@amazon.com