
Stream: Internet Engineering Task Force (IETF)
RFC: [9865](#)
Updates: [7643](#), [7644](#)
Category: Standards Track
Published: October 2025
ISSN: 2070-1721
Authors: M. Peterson, Ed. D. Zollner A. Sehgal
Entrust Independent Amazon Web Services

RFC 9865

Cursor-Based Pagination of System of Cross-domain Identity Management (SCIM) Resources

Abstract

This document updates RFCs 7643 and 7644 by defining additional System for Cross-Domain Identity Management (SCIM) query parameters and result attributes to allow use of cursor-based pagination in SCIM service providers that are implemented with existing codebases, databases, or APIs where cursor-based pagination is already well established.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <https://www.rfc-editor.org/info/rfc9865>.

Copyright Notice

Copyright (c) 2025 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 (<https://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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction	2
1.1. Notational Conventions	3
1.2. Definitions	3
2. Query Parameters and Response Attributes	3
2.1. Pagination Errors	6
2.2. Sorting	6
2.3. Implementing Cursors as the Only Pagination Method	6
2.4. Implementing Both Cursors and Index Pagination	7
3. Querying Resources Using HTTP POST	7
4. Service Provider Configuration	8
5. Security Considerations	10
5.1. Threat Model and Security Environment	10
5.2. Confidentiality	11
5.3. Availability	11
5.4. Other Security References	12
6. IANA Considerations	12
7. References	12
7.1. Normative References	12
7.2. Informative References	13
Acknowledgments	13
Contributors	13
Authors' Addresses	13

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 System for Cross-Domain

Identity Management (SCIM) 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 [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 character set defined in Section 2.3 of [RFC3986] .

Parameter	Description
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 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 (Resources omitted 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 of 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": "YkU30F86Pz0rGv",
  "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 defined in Table 9 ("SCIM Detail Error Keyword Values") of [Section 3.12](#) of [\[RFC7644\]](#)

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 be equal to the count value 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 that 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 a 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 providers 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. Service 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 [Section 3.12](#) of [\[RFC7644\]](#).
- Implement rate limiting to control the volume and cadence of cursor requests. This approach should adhere to established standards for rate limiting; details 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, as 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

IANA has amended the "System for Cross-domain Identity Management (SCIM) Schema URIs" registry group established by [\[RFC7643\]](#) as described below.

IANA has updated the "SCIM Schema URIs for Data Resources" registry as follows:

- For the `urn:ietf:params:scim:api:messages:2.0>ListResponse`, [Section 2](#) of this document has been added to the References column.
- For the `urn:ietf:params:scim:api:messages:2.0:SearchRequest`, [Section 2](#) of this document has been added to the References column.

IANA has updated the "SCIM Server-Related Schema URIs" registry as follows:

- For the `urn:ietf:params:scim:schemas:core:2.0:ServiceProviderConfig`, [Section 4](#) of this document has been added to the References column.

7. References

7.1. 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/info/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/info/rfc3986>>.
- [RFC6585]** Nottingham, M. and R. Fielding, "Additional HTTP Status Codes", RFC 6585, DOI 10.17487/RFC6585, April 2012, <<https://www.rfc-editor.org/info/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/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>>.

7.2. 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/info/rfc9110>>.

Acknowledgments

The authors would also like to acknowledge the following individuals who provided valuable feedback while reviewing the document: Aaron Parecki, David Brossard, Dean H. Saxe, and Pamela Dingle.

Contributors

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

Authors' Addresses

Matt Peterson (EDITOR)

Entrust

Email: matt.peterson@entrust.com

Danny Zollner

Independent

Email: danny@zollnerd.com**Anjali Sehgal**

Amazon Web Services

Email: anjalig@amazon.com