NETCONF Working Group
Internet-Draft
Updates: 8040 (if approved)
Intended status: Standards Track
Expires: 5 October 2025

K. Watsen
Watsen Networks
Q. Wu
Huawei Technologies
P. Andersson
Cisco Systems
O. Hagsand
SUNET
H. Li
Hewlett Packard Enterprise

3 April 2025

RESTCONF Extensions to Support List Pagination
draft-ietf-netconf-list-pagination-rc-08

Abstract

This document defines a mapping of the list pagination mechanism defined in [I-D.ietf-netconf-list-pagination] to RESTCONF [RFC8040].

This document updates RFC 8040, to declare "list" and "leaf-list" as valid resource targets for the RESTCONF GET operation, to define GET query parameters necessary for list pagination, and to define a media-type for XML-based lists.

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 5 October 2025.

Copyright Notice

Table of Contents

1. Introduction

This document defines a mapping of the list pagination mechanism defined in [I-D.ietf-netconf-list-pagination] to RESTCONF [RFC8040].

This document updates RFC 8040, as described in Section 2.

Declaring "list" and "leaf-list" as valid resource targets for the GET operation is necessary for list pagination.

1.1. Terminology

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. Conventions

Various examples in this document use "BASE64VALUE=" as a placeholder value for binary data that has been base64 encoded (per Section 9.8 of [RFC7950]). This placeholder value is used because real base64 encoded structures are often many lines long and hence distracting to the example being presented.

2. Updates to RFC 8040

2.1. Resource Targets

This document extends Section 3.5 of [RFC8040] to add "list" and "leaf-list" nodes (not just their entries) as valid data resources for the "GET" operation.

2.2. Media Type

This document extends Section 3.2 of [RFC8040] to add a new media type, "application/yang-data+xml-list", to encode "list" and "leaf-list" nodes in XML.

The "application/yang-data+xml-list" media-type defines a pseudo top-level element called "xml-list" that is used to wrap the response set, thus ensuring that a single top-level element is returned for the XML encoding", as required by Section 4.3 of [RFC8040].

For JSON, the existing "application/yang-data+json" media type is sufficient, as the JSON format has built-in support for encoding arrays.

The "application/yang-data+xml-list" media type is registered in Section 3.2.1.

2.3. Query Parameters

This document extends Section 4.8 of [RFC8040] to add new query parameters "limit", "offset", "cursor", "direction", "sort-by", "locale", "where", and "sublist-list".

These six query parameters correspond to those defined in Sections 3.1 and 3.2 in [I-D.ietf-netconf-list-pagination].

Name	Methods	Description
limit	GET, HEAD	Limits the number of entries returned. If not specified, the number of entries that may be returned is unbounded.
offset	GET, HEAD	Indicates the number of entries in the result set that should the skipped over when preparing the response. If not specified, then no entries in the result set are skipped.
cursor	GET, HEAD	Indicates where to start the working result set, the previous entries are skipped over. If not specified, then no entries in the result set are skipped over.
direction	GET, HEAD	Indicates the direction that the result set is to be traversed. If not specified, then the result set is traversed in the "forwards" direction.
sort-by	GET, HEAD	Indicates the node name that the result set should be sorted by. If not specified, then the result set’s default order is used, per YANG’s "ordered-by" statement.
locale	GET, HEAD	Specifies the locale the server should use when collating the result set. If not specified, the server chooses what locale is used for collation.
where	GET, HEAD	Specifies a filter expression that result set entries must match. If not specified, then no entries are filtered from the result set.
sublist-limit	GET, HEAD	Limits the number of entries returned returned for descendent lists and leaf-lists. If not specified, the number of entries that may be returned is unbounded.

For all of the query parameters, the query parameter is only allowed for the GET and HEAD methods on "list" and "leaf-list" data resources. A "400 Bad Request" status-line MUST be returned if used with any other method or resource type. The error-tag value "operation-not-supported" is used in this case.

Per the conformance defined in Section 3.1 of [I-D.ietf-netconf-list-pagination], all of these parameters MUST be supported for all "config true" lists and leaf-lists, and SHOULD be supported for "config false" lists and leaf-lists. A server MAY disable the support for some or all "config false" lists, as described in Section 3.3 of [I-D.ietf-netconf-list-pagination].

2.3.1. The "limit" Query Parameter

The "limit" query parameter corresponds to the "limit" parameter defined in Section 3.1.7 of [I-D.ietf-netconf-list-pagination].

If the limit value is invalid, i.e. not an unsigned 32 bit integer greater than or equal to 1 or the string "unbounded", then a "400 Bad Request" status-line MUST be returned with the error-type value "application" and error-tag value "invalid-value".

2.3.2. The "offset" Query Parameter

The "offset" query parameter corresponds to the "offset" parameter defined in Section 3.1.5 of [I-D.ietf-netconf-list-pagination].

If the offset value is invalid, a "400 Bad Request" status-line MUST be returned with the error-type value "application" and error-tag value "invalid-value".

If the offset value exceeds the number of entries in the working result set, then a "416 Range Not Satisfiable" status-line MUST be returned with the error-type value "application", error-tag value "invalid-value", and SHOULD also include the "offset-out-of-range" identity as error-app-tag value.

2.3.3. The "cursor" Query Parameter

The "cursor" query parameter corresponds to the "cursor" parameter defined in Section 3.1.6 of [I-D.ietf-netconf-list-pagination].

If the cursor value is unknown, i.e. the key does not exist, a "404 Not Found" status-line MUST be returned with the error-type value "application" and error-tag value "invalid-value", and SHOULD also include the "cursor-not-found" identity as error-app-tag value.

If the "cursor" query parameter is not supported on the target node, then a a "501 Not Implemented" status-line MUST be returned with error-type value "application" and error-tag value "operation-not-supported".

2.3.4. The "direction" Query Parameter

The "direction" query parameter corresponds to the "direction" parameter defined in Section 3.1.4 of [I-D.ietf-netconf-list-pagination].

If the direction value is invalid, then a "400 Bad Request" status-line MUST be returned with the error-type value "application" and error-tag value "invalid-value".

2.3.5. The "sort-by" Query Parameter

The "sort-by" query parameter corresponds to the "sort-by" parameter defined in Section 3.1.2 of [I-D.ietf-netconf-list-pagination].

If the specified node identifier is invalid, then a "400 Bad Request" status-line MUST be returned with the error-type value "application" and error-tag value "invalid-value".

2.3.6. The "locale" Query Parameter

The "locale" query parameter corresponds to the "locale" parameter defined in Section 3.1.3 of [I-D.ietf-netconf-list-pagination].

If the specified node identifier is invalid, i.e. the locale is unknown to the server, then a "501 Not Implemented" status-line MUST be returned with the error-type value "application" and error-tag value "invalid-value", and SHOULD also include the "locale-unavailable" identity in as the error-app-tag value.

If "locale" is supplied but not "sort-by", a "400 Bad Request" status-line MUST be return with the error-type "application" and error-tag value "invalid-value".

2.3.7. The "where" Query Parameter

The "where" query parameter corresponds to the "where" parameter defined in Section 3.1.1 of [I-D.ietf-netconf-list-pagination].

Prefixes in the XPath expression MUST be YANG module names.

If the specified XPath expression is invalid, then a "400 Bad Request" status-line MUST be returned with the error-type value "application" and error-tag value "invalid-value".

2.3.8. The "sublist-limit" Query Parameter

The "sublist-limit" query parameter corresponds to the "sublist-limit" parameter defined in Section 3.2.1 of [I-D.ietf-netconf-list-pagination].

If the sublist-limit value is invalid, then a "400 Bad Request" status-line MUST be returned with the error-type value "application" and error-tag value "invalid-value".

3. IANA Considerations

3.1. The "RESTCONF Capability URNs" Registry

This document registers six capabilities in the RESTCONF Capability URNs [RFC8040] maintained at https://www.iana.org/assignments/restconf-capability-urns/restconf-capability-urns.xhtml. Following the instructions defined in Section 11.4 of [RFC8040], the below registrations are requested:

All the registrations are to use this document (RFC XXXX) for the "Reference" value.

Index	Capability Identifier
:limit	urn:ietf:params:restconf:capability:limit:1.0
:offset	urn:ietf:params:restconf:capability:offset:1.0
:cursor	urn:ietf:params:restconf:capability:cursor:1.0
:direction	urn:ietf:params:restconf:capability:direction:1.0
:sort-by	urn:ietf:params:restconf:capability:sort-by:1.0
:locale	urn:ietf:params:restconf:capability:locale:1.0
:where	urn:ietf:params:restconf:capability:where:1.0
:sublist-limit	urn:ietf:params:restconf:capability:sublist-limit:1.0

3.2. The "Media Types" Registry

This document registers one media type in the "application" subregistry of the Media Types registry [RFC6838] [RFC4855] maintained at https://www.iana.org/assignments/media-types/media-types.xhtml#application. Following the format defined in [RFC4855], the below registration is requested:

3.2.1. Media Type "application/yang-data+xml-list"

Type name: application

Subtype name: yang-data+xml-list

Required parameters: None

Optional parameters: None

Encoding considerations: 8-bit
Each conceptual YANG data node is encoded according to the XML Encoding Rules and Canonical Format for the specific YANG data node type defined in [RFC7950].

Security considerations: Security considerations related to the generation and consumption of RESTCONF messages are discussed in Section 12 of RFC 8040. Additional security considerations are specific to the semantics of particular YANG data models. Each YANG module is expected to specify security considerations for the YANG data defined in that module.

Interoperability considerations: RFC XXXX specifies the format of conforming messages and the interpretation thereof.

Published specification: RFC XXXX

Applications that use this media type: Instance document data parsers used within a protocol or automation tool that utilize the YANG Patch data structure.

Fragment identifier considerations: Fragment identifiers for this type are not defined. All YANG data nodes are accessible as resources using the path in the request URI.

Additional information:

Deprecated alias names for this type: N/A
Magic number(s): N/A
File extension(s): None
Macintosh file type code(s): "TEXT"

Person & email address to contact for further information:
See the Authors' Addresses section of RFC XXXX.

Intended usage: COMMON

Restrictions on usage: N/A

Author: See the Authors' Addresses section of RFC XXXX.

Change controller: Internet Engineering Task Force
(mailto:iesg@ietf.org).

Provisional registration? (standards tree only): no

4. Security Considerations

This document introduces protocol operations for paging through data already provided by the RESTCONF protocol, and hence does not introduce any new security considerations.

This document does not define a YANG module and hence there are no data modeling considerations beyond those discussed in [I-D.ietf-netconf-list-pagination].

5. References

5.1. 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>.

[RFC8040] Bierman, A., Bjorklund, M., and K. Watsen, "RESTCONF Protocol", RFC 8040, DOI 10.17487/RFC8040, January 2017, <https://www.rfc-editor.org/info/rfc8040>.

[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>.

[I-D.ietf-netconf-list-pagination] Watsen, K., Wu, Q., Andersson, P., Hagsand, O., and H. Li, "List Pagination for YANG-driven Protocols", Work in Progress, Internet-Draft, draft-ietf-netconf-list-pagination-07, 3 April 2025, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-list-pagination-07>.

5.2. Informative References

[RFC4855] Casner, S., "Media Type Registration of RTP Payload Formats", RFC 4855, DOI 10.17487/RFC4855, February 2007, <https://www.rfc-editor.org/info/rfc4855>.

[RFC6838] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, DOI 10.17487/RFC6838, January 2013, <https://www.rfc-editor.org/info/rfc6838>.

[RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <https://www.rfc-editor.org/info/rfc7950>.

[I-D.ietf-netconf-restconf-collection] Bierman, A., Bj旦rklund, M., and K. Watsen, "RESTCONF Collection Resource", Work in Progress, Internet-Draft, draft-ietf-netconf-restconf-collection-00, 30 January 2015, <https://datatracker.ietf.org/doc/html/draft-ietf-netconf-restconf-collection-00>.

[I-D.zheng-netconf-fragmentation] Zheng, G. and Z. Wang, "A NETCONF Extension for Data Fragmentation", Work in Progress, Internet-Draft, draft-zheng-netconf-fragmentation-02, 26 June 2018, <https://datatracker.ietf.org/doc/html/draft-zheng-netconf-fragmentation-02>.

Appendix A. Example YANG Module

The examples within this document use the "example-social" YANG module defined in Appendix A.1 of [I-D.ietf-netconf-list-pagination].

Appendix B. Example Data Set

The Example Data Set used by the examples is defined in Appendix A.2 of [I-D.ietf-netconf-list-pagination].

Appendix C. Example Queries

C.1. List pagination with all query parameters

This example mimics that Appendix A.3.9 of [I-D.ietf-netconf-list-pagination]. This example is presented twice, once using XML and again using JSON.

XML:

```
   =============== NOTE: '\' line wrapping per RFC 8792 ================

   GET /restconf/ds/ietf-datastores:operational/example-social:members/\
   member?where=//stats//joined[starts-with(timestamp,'2020')]&sort-by=\
   timestamp&direction=backwards&offset=2&limit=2&sublist-limit=1 HTTP/\
   1.1
   Host: example.com
   Accept: application/yang-data+xml-list

   Response from the RESTCONF server:

   =============== NOTE: '\' line wrapping per RFC 8792 ================

   HTTP/1.1 200 OK
   Date: Thu, 26 Jan 2017 20:56:30 GMT
   Server: example-server
   Last-Modified: Thu, 26 Jan 2017 20:55:30 GMT
   Content-Type: application/yang-data+xml-list

   <xml-list>
     <member
       xmlns="https://example.com/ns/example-social"
       xmlns:lp="urn:ietf:params:xml:ns:yang:ietf-list-pagination"
       lp:remaining="1" lp:locale="en_US">
       <member-id>eric</member-id>
       <email-address>eric@example.com</email-address>
       <password>$0$1543</password>
       <avatar>BASE64VALUE=</avatar>
       <tagline>Go to bed with dreams; wake up with purpose.</tagline>
       <following>alice</following>
       <posts>
         <post>
           <timestamp>2020-09-17T18:02:04Z</timestamp>
           <title>Son, brother, husband, father</title>
           <body>What's your story?</body>
         </post>
       </posts>
       <favorites>
         <bits lp:remaining="2" lp:locale="en_US">two</bits>
       </favorites>
       <stats>
         <joined>2020-09-17T19:38:32Z</joined>
         <membership-level>pro</membership-level>
         <last-activity>2020-09-17T18:02:04Z</last-activity>
       </stats>
     </member>
     <member
       xmlns="https://example.com/ns/example-social"
```

```
      xmlns:lp="urn:ietf:params:xml:ns:yang:ietf-list-pagination"
      lp:remaining="1" lp:locale="en_US">
      <member-id>bob</member-id>
      <email-address>bob@example.com</email-address>
      <password>$0$1543</password>
      <avatar>BASE64VALUE=</avatar>
      <tagline>Here and now, like never before.</tagline>
      <posts>
        <post lp:remaining="2" lp:locale="en_US">
          <timestamp>2020-08-14T03:32:25Z</timestamp>
          <body>Just got in.</body>
        </post>
      </posts>
      <favorites>
        <decimal64-numbers lp:remaining="1" lp:locale="en_US">3.14159<\
/decimal64-numbers>
      </favorites>
      <stats>
        <joined>2020-08-14T03:30:00Z</joined>
        <membership-level>standard</membership-level>
        <last-activity>2020-08-14T03:34:30Z</last-activity>
      </stats>
    </member>
  </xml-list>

  JSON:

  =============== NOTE: '\' line wrapping per RFC 8792 ================

  GET /restconf/ds/ietf-datastores:running/example-social:members/memb\
  er?where=//stats//joined[starts-with(timestamp,'2020')]&sort-by=time\
  stamp&direction=backwards&offset=2&limit=2&sublist-limit=1 HTTP/1.1
  Host: example.com
  Accept: application/yang-data+json

  Response from the RESTCONF server:

  HTTP/1.1 200 OK
  Date: Thu, 26 Jan 2017 20:56:30 GMT
  Server: example-server
  Last-Modified: Thu, 26 Jan 2017 20:55:30 GMT
  Content-Type: application/yang-data+json

  {
    "example-social:member": [
      {
        "@": {
          "ietf-list-pagination:remaining": 1,
```

```
          "ietf-list-pagination:locale": "en_US"
        },
        "member-id": "eric",
        "email-address": "eric@example.com",
        "password": "$0$1543",
        "avatar": "BASE64VALUE=",
        "tagline": "Go to bed with dreams; wake up with purpose.",
        "following": ["alice"],
        "posts": {
          "post": [
            {
              "@": {
                "ietf-list-pagination:remaining": 2,
                "ietf-list-pagination:locale": "en_US"
              },
              "timestamp": "2020-09-17T18:02:04Z",
              "title": "Son, brother, husband, father",
              "body": "What's your story?"
            }
          ]
        },
        "favorites": {
          "bits": ["two"],
          "@example-social:bits": [
            {
              "ietf-list-pagination:remaining": 2,
              "ietf-list-pagination:locale": "en_US"
            }
          ]
        },
        "stats": {
          "joined": "2020-09-17T19:38:32Z",
          "membership-level": "pro",
          "last-activity": "2020-09-17T18:02:04Z"
        }
      },
      {
        "member-id": "bob",
        "email-address": "bob@example.com",
        "password": "$0$1543",
        "avatar": "BASE64VALUE=",
        "tagline": "Here and now, like never before.",
        "posts": {
          "post": [
            {
              "@": {
                "ietf-list-pagination:remaining": 2,
                "ietf-list-pagination:locale": "en_US"
```

```
              },
              "timestamp": "2020-08-14T03:32:25Z",
              "body": "Just got in."
            }
          ]
        },
        "favorites": {
          "decimal64-numbers": ["3.14159"],
          "@example-social:decimal64-numbers": [
            {
              "ietf-list-pagination:remaining": 1,
              "ietf-list-pagination:locale": "en_US"
            }
          ]
        },
        "stats": {
          "joined": "2020-08-14T03:30:00Z",
          "membership-level": "standard",
          "last-activity": "2020-08-14T03:34:30Z"
        }
      }
    ]
  }
```

Acknowledgements

This work has benefited from the discussions of RESTCONF resource collection over the years, in particular, [I-D.ietf-netconf-restconf-collection] which provides enhanced filtering features for the retrieval of data nodes with the GET method and [I-D.zheng-netconf-fragmentation] which document large size data handling challenge. The authors would like to thank the following for lively discussions on list (ordered by first name): Andy Bierman, Martin Bj旦rklund, Robert Varga, and Rob Wilton.

Authors' Addresses

Kent Watsen
Watsen Networks
Email: kent+ietf@watsen.net

Qin Wu
Huawei Technologies
Email: bill.wu@huawei.com

Per Andersson
Cisco Systems
Email: per.ietf@ionio.se

Olof Hagsand
SUNET
Email: olof@hagsand.se

Hongwei Li
Hewlett Packard Enterprise
Email: flycoolman@gmail.com