Abstract

This document describes a collection resource for the RESTCONF protocol to provide enhanced filtering features for the retrieval of data nodes with the GET method.

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 August 3, 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 ............................................. 2
   1.1. Terminology ........................................ 3
       1.1.1. NETCONF .................................... 3
       1.1.2. HTTP ......................................... 4
       1.1.3. YANG ......................................... 5
       1.1.4. RESTCONF .................................... 5
       1.1.5. Terms ......................................... 5
       1.1.6. URI Template .................................. 6
       1.1.7. Tree Diagrams ................................ 6
       1.2. Collection Resource Type ......................... 6
       1.3. Collection Resource .............................. 6
       1.4. Query Parameters ................................. 7
           1.4.1. Query Parameter URIs ....................... 7
           1.4.2. The "limit" Query Parameter ............... 8
           1.4.3. The "offset" Query Parameter ............... 8
           1.4.4. The "sort" Query Parameter ................. 8
           1.4.5. The "where" Query Parameter ................. 8
       2. RESTCONF Collection module ....................... 9
       3. IANA Considerations ................................. 11
           3.1. YANG Module Registry .......................... 11
           3.2. application/yang Media Sub Types ............ 11
           3.3. NETCONF Capability URNs ...................... 12
       4. Security Considerations ............................ 12
       5. Acknowledgements .................................... 13
       6. Normative References ................................ 13
Appendix A. Change Log ..................................... 14
    A.1. restconf-03 to restconf-collection-00 ........... 14
Appendix B. Open Issues .................................... 14
Appendix C. RESTCONF Collection Examples .................. 14
    C.1. "limit" Parameter ................................ 14
    C.2. "offset" Parameter ................................ 15
    C.3. "sort" Parameter ................................ 16
    C.4. "where" Parameter ................................ 16
    C.5. "TopN" Use Case ................................ 16
Authors’ Addresses ........................................ 16

1. Introduction

There is a need for standard mechanisms to control the filtering,
sorting, and retrieval of data from RESTCONF devices. A server may
contain many instances of a particular YANG list. Retrieval of the
entire list at once can be extremely inefficient.
Pagination mechanisms are needed to allow a client to iterate through a large list, in a manner that is most efficient for the application.

This document describes a "collection" resource that can be used to control retrieval of data nodes from a RESTCONF server.

[FIXME: describe basic needs
- target resource picks the list
- ‘fields’ is a node-set XPath expression to pick the subtrees within the target resource to return
- ‘where’ is a boolean XPath expression to pick which list instances are selected for return
- ‘sort’ is ??? parameter to sort the selected list instances
- ‘limit’ is the max number of list instances returned
- ‘offset’ is the XPath position() of the list instance
  ??? pre or post access control ???
  ??? if post, then what if NACM changes while client retrieving]

Collection resources represent search results through the server data. Data that the client is not authorized to receive according to the access control parameters configured in [RFC6536] MUST NOT be returned in RESTCONF response messages.

1.1. Terminology

The keywords "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].

[FIXME: remove terms that are not used]

1.1.1. NETCONF

The following terms are defined in [RFC6241]:

- candidate configuration datastore
- client
- configuration data
- datastore
- configuration datastore
protocol operation
running configuration datastore
server
startup configuration datastore
state data
user

1.1.2. HTTP

The following terms are defined in [RFC3986]:
fragment
path
query

The following terms are defined in [RFC7230]:
header
message-body
Request-Line
request URI

The following terms are defined in [RFC7231]:
method
request
resource

The following terms are defined in [RFC7232]:
entity tag
1.1.3. YANG

The following terms are defined in [RFC6020]:

- container
- data node
- key leaf
- leaf
- leaf-list
- list
- presence container (or P-container)
- RPC operation (now called protocol operation)
- non-presence container (or NP-container)
- ordered-by system
- ordered-by user

1.1.4. RESTCONF

The following terms are defined in [I-D.ietf-netconf-restconf]:

- data resource
- target resource
- retrieval request

1.1.5. Terms

The following terms are used within this document:

- collection resource: a resource with the media type "application/yang.collection+xml" or "application/yang.collection+json". Contains a set of data resources.
1.1.6. URI Template

Throughout this document, the URI template [RFC6570] syntax 
"{+restconf}" is used to refer to the RESTCONF API entry point 
outside of an example. See the root resource discovery section 
defined in [I-D.ietf-netconf-restconf] for details.

All of the examples in this document assume "/restconf" as the 
discovered RESTCONF API root path.

1.1.7. Tree Diagrams

A simplified graphical representation of the data model is used in 
this document. The meaning of the symbols in these diagrams is as 
follows:

- Brackets "[" and "]" enclose list keys.
- Abbreviations before data node names: "rw" means configuration 
data (read-write) and "ro" state data (read-only).
- Symbols after data node names: "?" means an optional node, "!" 
means a presence container, and "*" denotes a list and leaf-list.
- Parentheses enclose choice and case nodes, and case nodes are also 
marked with a colon (":"
- Ellipsis ("...") stands for contents of subtrees that are not 
shown.

1.2. Collection Resource Type

The following resource type are defined in this document:

<table>
<thead>
<tr>
<th>Resource</th>
<th>Media Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Collection</td>
<td>application/yang.collection</td>
</tr>
</tbody>
</table>

RESTCONF Media Types

1.3. Collection Resource

A collection resource contains a set of data resources. It is used 
to represent a all instances or a subset of all instances in a YANG 
list or leaf-list.
A collection resource can be retrieved with the GET method, optionally with the query parameters "limit" (Section 1.4.2) and "offset" (Section 1.4.3).

The "ietf-restconf-collection" YANG module contains the "application/yang.collection" restconf-media-type extension which specifies the syntax and semantics of a "collection" media type.

### 1.4. Query Parameters

Each RESTCONF operation allows zero or more query parameters to be present in the request URI. The following query parameters are defined for RESTCONF collection resources.

<table>
<thead>
<tr>
<th>Name</th>
<th>Methods</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>limit</td>
<td>GET</td>
<td>Number of entries to return for collection resources</td>
</tr>
<tr>
<td>offset</td>
<td>GET</td>
<td>Starting point for collection resources</td>
</tr>
<tr>
<td>sort</td>
<td>GET</td>
<td>Sorting criteria for collection resources</td>
</tr>
<tr>
<td>where</td>
<td>GET</td>
<td>Boolean filter to select data instances for a collection resource</td>
</tr>
</tbody>
</table>

**RESTCONF Query Parameters**

Query parameters can be given in any order. Each parameter can appear at most once in a request URI. A default value may apply if the parameter is missing.

Refer to Appendix C for examples of query parameter usage.

If vendors define additional query parameters, they SHOULD use a prefix (such as the enterprise or organization name) for query parameter names in order to avoid collisions with other parameters.

### 1.4.1. Query Parameter URIs

A new set of RESTCONF Capability URNs are defined to identify the specific query parameters supported by the server.
1.4.2. The "limit" Query Parameter

The "limit" parameter is used to restrict the number of data resources to return in response to GET requests on collection resources.

The value of the "limit" parameter is either an integer greater than or equal to 1, or the string "unbounded". The string "unbounded" is the default value.

If the server includes the "page" query parameter URI in the "capability" leaf-list in the "ietf-restconf-monitoring" module defined in [I-D.ietf-netconf-restconf], then the "limit" query parameter MUST be supported.

1.4.3. The "offset" Query Parameter

The "offset" parameter is used to specify the first data resource to return in response to GET requests on collection resources. Resources instances are numbered with consecutive integers from 1 to the number of resource instances.

The value of the "offset" parameter is an integer greater than or equal to 1. The default value is 1.

If the server includes the "page" query parameter URI in the "capability" leaf-list in "ietf-restconf-monitoring" module defined in [I-D.ietf-netconf-restconf], then the "offset" query parameter MUST be supported.

1.4.4. The "sort" Query Parameter

[FIXME]

1.4.5. The "where" Query Parameter

[FIXME]
2. RESTCONF Collection module

The "ietf-restconf-collection" module defines conceptual definitions within groupings, which are not meant to be implemented as datastore contents by a server.

The "ietf-restconf" module from [I-D.ietf-netconf-restconf] is used by this module for the ‘restconf-media-type’ extension definition.

RFC Ed.: update the date below with the date of RFC publication and remove this note.

<CODE BEGINS> file "ietf-restconf-collection@2015-01-30.yang"
module ietf-restconf-collection {
  prefix "rcoll";

  import ietf-restconf {
    prefix rc;
    revision-date 2015-01-30;
  }

  organization
    "IETF NETCONF (Network Configuration) Working Group";

  contact
    "WG Web:  <http://tools.ietf.org/wg/netconf/>
    WG List:  <mailto:netconf@ietf.org>
    WG Chair: Mehmet Ersue
      <mailto:mehmet.ersue@nsn.com>
    WG Chair: Mahesh Jethanandani
      <mailto:mjethanandani@gmail.com>
    Editor:  Andy Bierman
      <mailto:andy@yumaworks.com>
    Editor:  Martin Bjorklund
      <mailto:mbj@tail-f.com>
    Editor:  Kent Watsen
      <mailto:kwatsen@juniper.net>"

  description
    "This module contains conceptual YANG specifications for the RESTCONF Collection resource type."
Note that the YANG definitions within this module do not represent configuration data of any kind. The YANG grouping statements provide a normative syntax for XML and JSON message encoding purposes.

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

Redistribution and use in source and binary forms, with or without modification, is permitted pursuant to, and subject to the license terms contained in, the Simplified BSD License set forth in Section 4.c of the IETF Trust’s Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info).

This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.

// RFC Ed.: replace XXXX with actual RFC number and remove this note.

// RFC Ed.: remove this note
// Note: extracted from
// draft-ietf-netconf-restconf-collection-00.txt

// RFC Ed.: update the date below with the date of RFC publication
// and remove this note.
revision 2015-01-30 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: RESTCONF Collection Resource.";
}

rc:restconf-media-type "application/yang.collection" {
  uses collection;
}

grouping collection {
  description
    "Conceptual container representing the application/yang.collection resource type.";

  container collection {
    description
      "Container representing the application/yang.collection resource type.";
  }
}
3. IANA Considerations

3.1. YANG Module Registry

This document registers three URIs in the IETF XML registry [RFC3688]. Following the format in RFC 3688, the following registration is requested to be made.

Registrant Contact: The NETMOD WG of the IETF.
XML: N/A, the requested URI is an XML namespace.

This document registers three YANG modules in the YANG Module Names registry [RFC6020].

name:     ietf-restconf-collection
prefix:   rcoll
// RFC Ed.: replace XXXX with RFC number and remove this note
reference: RFC XXXX

3.2. application/yang Media Sub Types

The parent MIME media type for RESTCONF resources is application/yang, which is defined in [RFC6020]. This document defines the following sub-types for this media type.
- collection

Type name: application

Subtype name: yang.xxx

Required parameters: TBD

Optional parameters: TBD

Encoding considerations: TBD

Security considerations: TBD

Interoperability considerations: TBD

// RFC Ed.: replace XXXX with RFC number and remove this note
Published specification: RFC XXXX

3.3. NETCONF Capability URNs

This document registers two capability identifiers in "RESTCONF Protocol Capability URNs" registry

Index
   Capability Identifier
   -----------------------
   :page
      urn:ietf:params:restconf:capability:page:1.0

   :page-xpath
      urn:ietf:params:restconf:capability:page-xpath:1.0

4. Security Considerations

This section provides security considerations for the resources defined by the RESTCONF protocol. Security considerations for HTTPS are defined in [RFC2818]. Security considerations for the content manipulated by RESTCONF can be found in the documents defining data models.

All security considerations that apply to resources defined in [I-D.ietf-netconf-restconf] also apply to the collection resource.
5. Acknowledgements

The authors would like to thank for following for lively discussions on list and in the halls (ordered by last name): Rex Fernando

6. Normative References

[I-D.ietf-netconf-restconf]


Appendix A. Change Log

-- RFC Ed.: remove this section before publication.

The RESTCONF issue tracker can be found here: https://github.com/netconf-wg/restconf/issues

A.1. restconf-03 to restconf-collection-00

- Moved collection resource from RESTCONF to a new document

Appendix B. Open Issues

-- RFC Ed.: remove this section before publication.

The RESTCONF Collection issues are tracked on github.com:

https://github.com/netconf-wg/restconf/issues

Appendix C. RESTCONF Collection Examples

The examples within this document use the "example-jukebox" YANG module defined in [I-D.ietf-netconf-restconf].

C.1. "limit" Parameter

In this example, the client requests the first two "album" resources for a given artist:

Request from client:

GET /restconf/data/example-jukebox:jukebox/
   library/artist=Foo%20Fighters/album/?limit=2 HTTP/1.1
Host: example.com
Content-Type: application/yang.collection+xml

Response from server:

HTTP/1.1 200 OK
Date: Mon, 23 Apr 2012 17:01:00 GMT
Server: example-server
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/yang.collection+xml
C.2. "offset" Parameter

In this example, the client requests the next two albums, i.e., two albums starting from two.

Request from client:

GET /restconf/data/example-jukebox:jukebox/
  library/artist=Foo%20Fighters/album/?limit=2&offset=2 HTTP/1.1
Host: example.com
Content-Type: application/yang.collection+json

Response from server:
HTTP/1.1 200 OK
Date: Mon, 23 Apr 2012 17:02:00 GMT
Server: example-server
Cache-Control: no-cache
Pragma: no-cache
Content-Type: application/yang.collection+json

{
    "collection": {
        "example-jukebox:album" : [
            {
                "year" : 1999,
                "name" : "There is Nothing Left to Loose",

            },
            {
                "year" : 2002,
                "name" : "One by One",

            }
        ]
    }
}

C.3. "sort" Parameter

[FIXME]

C.4. "where" Parameter

[FIXME]

C.5. "TopN" Use Case

[FIXME: use-case using all parameters for topN for some list]

Authors’ Addresses

Andy Bierman
YumaWorks

Email: andy@yumaworks.com

Martin Bjorklund
Tail-f Systems

Email: mbj@tail-f.com