

NETMOD Working Group
Internet-Draft
Intended status: Standards Track
Expires: December 15, 2018

L. Lhotka
CZ.NIC
June 13, 2018

RESTCONF with Transactions
draft-lhotka-netconf-restconf-transactions-00

Abstract

This document extends the RESTCONF protocols with transaction capabilities that allow for safe concurrent access of multiple clients.

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 December 15, 2018.

Copyright Notice

Copyright (c) 2018 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 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
2.	Terminology	3
2.1.	YANG	3
2.2.	HTTP	3
2.3.	RESTCONF	3
2.4.	NMDA	3
2.5.	New Terms	4
3.	Datastores	4
3.1.	The Staging Configuration Datastore	4
4.	New Operations	5
4.1.	Commit	5
4.2.	Reset	5
5.	Access Control	5
6.	Compatibility	6
7.	YANG Module	6
8.	IANA Considerations	8
9.	Security Considerations	8
10.	References	8
10.1.	Normative References	8
10.2.	URIs	10
	Author's Address	10

1. Introduction

The RESTCONF protocol [RFC8040] was introduced as a simpler alternative to the original NETCONF protocol [RFC6241]. Due to the simplicity requirement, some more complex features and functions of NETCONF, such as locks, subtree filtering or candidate configuration datastore, are not available in RESTCONF.

On the other hand, RESTCONF offers several advantages over NETCONF, including:

- o the use of HTTP methods and well-known Representational State Transfer (REST) approaches make it more accessible to developers and increases the choice of software libraries and tools
- o cleaner semantics of edit operations,
- o alternative encodings in which resources can be represented, currently JSON and XML; NETCONF supports only XML
- o certain HTTP mechanisms, such as "Last-Modified" and "ETag" headers.

This document extends the RESTCONF protocol with transaction capabilities, at the cost of adding two RPC operations and some complexity in the server implementation. This makes RESTCONF suitable for network management environments where concurrent access of multiple client is needed.

A RESTCONF server indicates support for transactions as defined in this document by including the YANG module "ietf-restconf-transactions" (Section 7) among implemented modules in its YANG library data [I-D.ietf-netconf-rfc7895bis].

2. Terminology

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].

2.1. YANG

The following terms are defined in [RFC7950]:

- o RPC operation

2.2. HTTP

The following terms are defined in [RFC7231]:

- o resource

The following terms are defined in [RFC7232]:

- o entity-tag

2.3. RESTCONF

The following terms are defined in [RFC8040]:

- o client
- o RESTCONF root resource, {+restconf}

2.4. NMDA

The following terms are defined in [RFC8342]:

- o candidate configuration datastore, <candidate>
- o intended configuration datastore, <intended>

- o operational state datastore, <operational>
- o running configuration datastore, <running>

2.5. New Terms

The following new term is used in this document:

- o staging configuration datastore: a configuration datastore that represents a staging area private to each RESTCONF user, and that is eventually committed into <intended>.

3. Datastores

A RESTCONF server implementing this document MUST be NMDA-compliant [I-D.ietf-netconf-nmda-restconf]. Apart from the operational state datastore, it MUST also support the intended configuration datastore.

The intended configuration datastore SHOULD persist across server reboots. In terms of the NMDA architecture [RFC8342], <intended> can be considered identical to <running>, although this document does not explicitly use the latter datastore.

3.1. The Staging Configuration Datastore

This document introduces a new configuration datastore named 'staging' that represents a staging area private to each user (as identified by RESTCONF username, see [RFC8040] Section 2.5).

In NETCONF terms ([RFC8040] Section 8.3), the staging datastore is essentially a non-shared candidate configuration datastore. The new name is used in order to emphasize the narrower semantics: the staging datastore MUST be private to each user.

Note that the above requirement does not necessarily mean that each user is provided with a separate copy of configuration data. For instance, several efficient methods utilizing persistent data structures and copy-on-write are available. However, the particular implementation approach is outside the scope of this document.

The staging datastore assumes the place of the datastore resource as defined in [RFC8040] Section 3.4. This means that all resources inside the "{+restconf}/data" subtree correspond to data instances in the staging datastore. Therefore, the contents of the staging datastore can be retrieved by means of the GET method and modified by means of PUT, POST and PATCH methods exactly as specified in [RFC8040]. However, the changes to the staging datastore MUST NOT

impact operational state of the device until they are merged into <intended> via the "commit" operation (Section 4.1).

4. New Operations

In order to support transactions in RESTCONF, the YANG module "ietf-restconf-transactions" defines two RPC operations described below.

4.1. Commit

The "commit" operation atomically merges the contents of the client's staging datastore into <intended>.

The concrete strategy and implementation of the merge procedure is outside the scope of this document. The resolution of merge conflicts can be fully automatic, which is preferable, or may require client's intervention. In the latter case, the server SHALL send an error response with the following properties:

- o HTTP status code 412
- o error-tag of "operation-failed"
- o error-app-tag of "merge-conflict"
- o error-info containing additional information to aid the user in resolving the conflict.

4.2. Reset

The "reset" operation resets the user's staging datastore to the current contents of <intended>.

If the server supports entity-tags (see [RFC8040] Section 3.5.2), then after completing the "reset" operation the entity-tags for the staging and intended datastore resources MUST be identical.

5. Access Control

A server that implements this document along with NETCONF Access Control Model [RFC8341] MUST guarantee that all NACM rules are observed. This means in particular:

- o Configuration data that is not readable for a given user MUST NOT be exposed in the user's staging datastore.

- o A commit operation executed by a given user MUST NOT modify configuration data in <intended> in a way that is not compliant with NACM rules that are in effect for that user.

6. Compatibility

RESTCONF with transactions, as defined in this document, is compatible with the original RESTCONF specification [RFC8040] and RESTCONF NMDA extensions [I-D.ietf-netconf-nmda-restconf].

In practical terms, standard RESTCONF clients are able to retrieve and edit data in the staging configuration datastore. If they support NMDA, they can also retrieve data from <intended> and <operational>. In order to make them useful for network management, it is only necessary to allow for executing the "commit" and "reset" operations. This can be accomplished through simple scripts utilizing curl [1] or similar tools.

7. YANG Module

RFC Editor: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

```
<CODE BEGINS> file "ietf-restconf-transactions@2018-06-11.yang"
```

```
module ietf-restconf-transactions {  
  
  namespace  
    "urn:ietf:params:xml:ns:yang:ietf-restconf-transactions";  
  
  prefix rct;  
  
  organization  
    "IETF NETCONF (Network Configuration) Working Group";  
  
  contact  
    "WG Web: <https://tools.ietf.org/wg/netconf/>  
    WG List: <mailto:netconf@ietf.org>  
  
    WG Chair: Kent Watsen  
              <mailto:kwatsen@juniper.net>  
  
    WG Chair: Mahesh Jethanandani  
              <mailto:mjethanandani@gmail.com>  
  
    Editor: Ladislav Lhotka  
            <mailto:lhotka@nic.cz>";
```

description

"This module defines operations that implement transactions in the RESTCONF protocol.

Copyright (c) 2018 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 (<https://trustee.ietf.org/license-info>).

The key words 'MUST', 'MUST NOT', 'REQUIRED', 'SHALL', 'SHALL NOT', 'SHOULD', 'SHOULD NOT', 'RECOMMENDED', 'MAY', and 'OPTIONAL' in the module text are to be interpreted as described in RFC 2119 (<https://tools.ietf.org/html/rfc2119>).

This version of this YANG module is part of RFC XXXX (<https://tools.ietf.org/html/rfcXXXX>); see the RFC itself for full legal notices."

```
revision 2018-06-11 {
  description
    "Initial revision.";
  reference
    "RFC XXXX: RESTCONF with Transactions";
}

/* Operations */

rpc commit {
  description
    "Atomically merge the contents of client's staging datastore
    into the intended datastore.

    If a merge conflict occurs that cannot be automatically
    resolved, the server SHALL send an error report with
    error-app-tag set to 'merge-conflict' and error-info
    indicating the reason for the conflict.";
}

rpc reset {
  description
    "Reset the client's staging datastore so that its contents is
    identical to the contents of the intended repository.";
}
```

```
}
```

```
<CODE ENDS>
```

8. IANA Considerations

RFC Editor: In this section, replace all occurrences of 'XXXX' with the actual RFC number and all occurrences of the revision date below with the date of RFC publication (and remove this note).

This document registers one URI in the IETF XML Registry [RFC3688]. The following registration has been made:

URI: urn:ietf:params:xml:ns:yang:ietf-restconf-transactions

Registrant Contact: The IESG.

XML: N/A, the requested URI is an XML namespace.

This document registers one YANG module in the YANG Module Names Registry [RFC6020]. The following registration has been made:

```
name:          ietf-restconf-transactions
namespace:    urn:ietf:params:xml:ns:yang:ietf-restconf-transactions
prefix:       rct
reference:    RFC XXXX
```

9. Security Considerations

TBD

10. References

10.1. Normative References

[I-D.ietf-netconf-nmda-restconf]

Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "RESTCONF Extensions to Support the Network Management Datastore Architecture", draft-ietf-netconf-nmda-restconf-04 (work in progress), April 2018.

[I-D.ietf-netconf-rfc7895bis]

Bierman, A., Bjorklund, M., Schoenwaelder, J., Watsen, K., and R. Wilton, "YANG Library", draft-ietf-netconf-rfc7895bis-06 (work in progress), April 2018.

- [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>>.
- [RFC3688] Mealling, M., "The IETF XML Registry", BCP 81, RFC 3688, DOI 10.17487/RFC3688, January 2004, <<https://www.rfc-editor.org/info/rfc3688>>.
- [RFC6020] Bjorklund, M., Ed., "YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF)", RFC 6020, DOI 10.17487/RFC6020, October 2010, <<https://www.rfc-editor.org/info/rfc6020>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7231] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Semantics and Content", RFC 7231, DOI 10.17487/RFC7231, June 2014, <<https://www.rfc-editor.org/info/rfc7231>>.
- [RFC7232] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Conditional Requests", RFC 7232, DOI 10.17487/RFC7232, June 2014, <<https://www.rfc-editor.org/info/rfc7232>>.
- [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>>.
- [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>>.
- [RFC8341] Bierman, A. and M. Bjorklund, "Network Configuration Access Control Model", STD 91, RFC 8341, DOI 10.17487/RFC8341, March 2018, <<https://www.rfc-editor.org/info/rfc8341>>.
- [RFC8342] Bjorklund, M., Schoenwaelder, J., Shafer, P., Watsen, K., and R. Wilton, "Network Management Datastore Architecture (NMDA)", RFC 8342, DOI 10.17487/RFC8342, March 2018, <<https://www.rfc-editor.org/info/rfc8342>>.

10.2. URIs

[1] <https://curl.haxx.se>

Author's Address

Ladislav Lhotka
CZ.NIC

Email: lhotka@nic.cz