

RESTCONF Issues

draft-bierman-netconf-restconf-04
NETCONF WG
IETF #89 London, UK

Andy Bierman <andy@yumaworks.com>
Martin Björklund <mbj@tail-f.com>
Kent Watsen <kwatsen@juniper.net>
Rex Fernando <rex@cisco.com>

v0.3

Agenda

- Discuss and resolve open issues

B.1. message-id

- There is no "message-id" field in a RESTCONF message.
 - Is a message identifier needed? If so, should either the "Message-ID" or "Content-ID" header from RFC 2392 be used for this purpose?

B.2. select parameter

- What syntax should be used for the "select" query parameter?
 - The current choices are "XPath" and "path-expr".
 - Perhaps an additional parameter to identify the select string format is needed to allow extensibility?

B.3. server support verification

- Are all header lines used by RESTCONF supported by common application frameworks, such as FastCGI and WSGI?
 - If not, then should query parameters be used instead, since the QUERY_STRING is widely available to WEB applications?

B.4. error media type

- Should the <errors> element returned in error responses be a separate media type?

B.5. additional datastores

- How should additional datastores be supported, which may be added to the NETCONF/NETMOD framework in the future?

B.6. PATCH media type discovery

- How does a client know which PATCH media types are supported by the server in addition to application/yang.data and application/yang.patch?

B.7. RESTCONF version

- Is the /restconf/version field considered meta-data?
 - Should it be returned as XRD (Extensible Resource Descriptor)? In addition or instead of the version field?
 - Should this be the ietf-restconf YANG module revision date, instead of the string 1.0?

B.8. YANG to resource mapping

- Since data resources can only be YANG containers or lists, what should be done about top-level YANG data nodes that are not containers or lists?
 - Are they allowed in RESTCONF?
- Can a choice be a resource?
 - YANG choices are invisible to RESTCONF at this time.

B.9. .well-known usage

- Does RESTCONF need to Use a .well-known link relation to re-map API entry point?
 - The client first discovers the server's root for the RESTCONF API. In this example, "/api/restconf":

Request

```
GET /.well-known/host-meta users HTTP/1.1
Host: example.com
Accept: application/xrd+xml
```

Response

```
HTTP/1.1 200 OK
Content-Type: application/xrd+xml
Content-Length: nnn
```

```
<XRD xmlns='http://docs.oasis-open.org/ns/xri/xrd-1.0'>
  <Link rel='restconf' href='/api/restconf'/>
</XRD>
```

B.9. .well-known usage (2)

- Once discovering the RESTCONF API root, the client **MUST** prepend it to any access to a RESTCONF resource:

Request

```
GET /api/restconf/version HTTP/1.1
Host: example.com
Accept: application/yang.api+json
```

Response

```
HTTP/1.1 200 OK
Date: Mon, 23 Apr 2012 17:01:00 GMT
Server: example-server
Cache-Control: no-cache
Pragma: no-cache
Last-Modified: Sun, 22 Apr 2012 01:00:14 GMT
Content-Type: application/yang.api+json
```

```
{ "version": "1.0" }
```

B.10. `_self` links for HATEOAS support

- Should there be a mode where the client can request that the resource identifier is returned in a GET request?

B.11. netconf-state monitoring support

- Should long-term RESTCONF operations (i.e. SSE long-poll) be considered sessions with regards to NETCONF monitoring "session" list?
 - If so, what text is needed in RESTCONF draft to standardize the RESTCONF session entries?

B.12. secure transport

- Details to support secure operation over TLS are needed
- Security considerations need to be written