

# RESTCONF Protocol

draft-ietf-netconf-restconf-04  
NETCONF WG  
IETF #92 Dallas, TX, USA

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

v0.1

# Agenda

- RESTCONF-04 Update highlights
- Review resolution proposals for open issues
  - <https://github.com/netconf-wg/restconf/issues>
- Next Steps

# Issues resolved in draft-04

- #1 select parameter
  - changes to 'fields'
- #2 monitoring support
  - no session monitoring added
- #3 collection resource
  - moved to draft-ietf-netconf-restconf-resource-00
- #4 defaults handling
  - defaults (to report basic mode) and with-defaults

# Issues resolved in draft-04

- #9 authentication
  - new text added
- #13 ietf-yang-library
  - moved to draft-ietf-netconf-yang-library-00
- #14 module name prefixing rules
  - align with JSON draft
- #15 use of groupings
  - restconf-media-type extension added
- #16 content parameter
  - now mandatory to implement

# New Issues to be verified

- #17 terminal nodes as resources
  - Will clarify that anyxml does not have sub-resources
- #18 depth parameter and anyxml
  - Will clarify that anyxml subtree is treated as a single layer
- #19 fields parameter and anyxml
  - Will clarify that fields cannot reference sub-nodes within anyxml

# Datastore Issue Revisited

- Use of the unified datastore
  - Should running and startup be exposed?
    - Resource locations would be to be datastore specific. How will this be done to be compatible with REST?
  - Edits will become multi-step. How will this be handled without any sessions or any locking?
  - What is the purpose of 'running' if it is different from 'data'? How are edits done (just change the name data to running? What does that solve?)

# Possible Datastore Solution

- Do not mimic NETCONF, just provide control of non-volatile configuration storage
  - New query parameter: nv-store=true|false
    - POST /restconf/data/foo=1?nv-store=false
    - POST /restconf/data/foo=2?nv-store=true
      - true: server is requested to update the NV datastore so it is in synch with the running datastore
      - false: server is requested not to update the NV datastore
  - Applies to the entire datastore, not single edits
  - Allow empty PATCH of datastore to only request NV-store
    - PATCH /restconf/data?nv-store=true

# RESTCONF Specific Data

- example-jukebox has a data-resource-identifier
  - URL format for RESTCONF as a datatype
  - no reason NETCONF client cannot format this string or a NETCONF server could not accept this format
- Is there an expectation that every YANG module will be implemented by every protocol that uses YANG?
  - How will clients know which protocols use which YANG modules?
    - Need to add support in ietf-yang-library.yang



# insert & point parameters

- YANG user-ordered lists cannot be patched without insert and point parameters
  - Could use PUT to replace entire list and its parent container
  - Should these parameters be mandatory to implement? (Currently optional)
    - Heavy-weight operation that is rarely used, or important feature for every RESTCONF device?

# Summary

- Many editorial comments from Jürgen to go through and clarify document text
- Need more cross-area review comments
- Need to finalize scope of RESTCONF
  - Is this a REST API or is it NETCONF over HTTP?
- Need to finish RESTCONF Collections draft