YANG Groupings for HTTP Clients and HTTP Servers
draft-kwatsen-netconf-http-client-server-00

Abstract

This document defines two YANG modules: the first defines a grouping for configuring a generic HTTP client, and the second defines a grouping for configuring a generic HTTP server. It is intended that these groupings will be used by applications using the HTTP protocol.

Editorial Note (To be removed by RFC Editor)

This draft contains many placeholder values that need to be replaced with finalized values at the time of publication. This note summarizes all of the substitutions that are needed. No other RFC Editor instructions are specified elsewhere in this document.

Artwork in this document contains placeholder values for the date of publication of this draft. Please apply the following replacement:

- "2019-03-09" --> the publication date of this draft

The following Appendix section is to be removed prior to publication:

- Appendix A. Change Log

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 September 10, 2019.
1. Introduction

This document defines two YANG 1.1 [RFC7950] modules: the first defines a grouping for configuring a generic HTTP client, and the second defines a grouping for configuring a generic HTTP server. It is intended that these groupings will be used by applications using the HTTP protocol. For instance, these groupings could help define the configuration module for an SSH, TLS, or HTTP based application.
2. 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.

3. The HTTP Client Model

3.1. Tree Diagram

This section provides a tree diagram [RFC8340] for the "ietf-http-client" module.

```
module: ietf-http-client

  grouping http-client-grouping
    +-- http-client-identity
    |    +-- (auth-type)?
    |    |   +-- (basic)
    |    |    |   +-- basic
    |    |    |    |   +-- user-id? string
    |    |    |    |   +-- password? string
    |    |   +-- (bearer)
    |    |    |   +-- bearer
    |    |    |    |   +-- token? string
    |    |   +-- (digest)
    |    |    |   +-- digest
    |    |    |    |   +-- username? string
    |    |    |    |   +-- password? string
    |    |   +-- (hoba)
    |    |    |   +-- hoba
    |    |   +-- (mutual)
    |    |    |   +-- mutual
    |    |   +-- (negotiate)
    |    |    |   +-- negotiate
    |    |   +-- (oauth)
    |    |    |   +-- oauth
    |    |    |    |   +-- (scram-sha-1)
    |    |    |    |   +-- scram-sha-1
    |    |    |    |   +-- (scram-sha-256)
    |    |    |    |   +-- scram-sha-256
    |    |    |   +-- (vapid)
    |    |    |    |   +-- vapid
    +-- http-keepalives (http-client-keepalives)?
        +-- max-wait? uint16
        +-- max-attempts? uint8
```
grouping http-client-identity-grouping
  +- http-client-identity
  |  +- (auth-type)?
  |     +- :(basic)
  |        |  +- basic
  |        |     |  +- user-id? string
  |        |     |  +- password? string
  |        +- :(bearer)
  |        |  +- bearer
  |        |     |  +- token? string
  |        +- :(digest)
  |        |  +- digest
  |        |     |  +- username? string
  |        |     |  +- password? string
  |        +- :(hoba)
  |        |  +- hoba
  |        +- :(mutual)
  |        |  +- mutual
  |        +- :(negotiate)
  |        |  +- negotiate
  |        +- :(oauth)
  |        |  +- oauth
  |        |     +- :(scram-sha-1)
  |        |     |  +- scram-sha-1
  |        |     +- :(scram-sha-256)
  |        |     |  +- scram-sha-256
  |        |     +- :(vapid)
  |        |     |  +- vapid

grouping http-keepalives-grouping
  +- http-keepalives (http-client-keepalives)?
    +- max-wait?  uint16
    +- max-attempts?  uint8

3.2. Example Usage

This section presents an example showing the http-client-grouping populated with some data.
3.3. YANG Module

This YANG module has normative references to [RFC6991].

<CODE BEGINS> file "ietf-http-client@2019-03-09.yang"
module ietf-http-client {
    yang-version 1.1;
    prefix httpc;

    organization
        "IETF NETCONF (Network Configuration) Working Group";
    contact
        "WG Web:  <http://datatracker.ietf.org/wg/netconf/>
        WG List:  <mailto:netconf@ietf.org>
        Author:  Kent Watsen <mailto:kent+ietf@watsen.net>";
    description
        "This module defines reusable groupings for HTTP clients that
        can be used as a basis for specific HTTP client instances.

        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.

        Copyright (c) 2019 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

        "Watsen                 Expires September 10, 2019               [Page 5]"
This version of this YANG module is part of RFC XXXX; see the RFC itself for full legal notices.

revision 2019-03-09 {
  description
  "Initial version";
  reference
  "RFC XXXX: YANG Groupings for HTTP Clients and HTTP Servers";
}

// Features

feature http-client-keepalives {
  description
  "Per socket HTTP keepalive parameters are configurable for HTTP clients on the server implementing this feature.";
}

// Groupings

grouping http-client-grouping {
  description
  "A reusable grouping for configuring a HTTP client, including the IP address and port number it initiates a connections to.";
  uses http-client-identity-grouping;
  uses http-keepalives-grouping;
  // uses http-proxy-grouping;
}

grouping http-client-identity-grouping {
  description
  "A reusable grouping for configuring a HTTP client identity.";
  container http-client-identity {
    description
    "The credentials used by the client to authenticate to the HTTP server.";
    choice auth-type {
      description
      "The authentication type.";
      container basic {
        leaf user-id {
          type string;
          description
          "The user-id for the authenticating client.";
        }
      }
    }
  }
}
leaf password {
  type string;
  description
    "The password for the authenticating client."
}

container bearer {
  leaf token {
    type string;
    description
      "The bearer token for the authenticating client,
       encoded in base64, as described in RFC 6750,
       Section 2.1."
  }
  description
    "The ‘bearer’ HTTP scheme credentials.";
  reference
    "RFC 6750: The OAuth 2.0 Authorization Framework:
       Bearer Token Usage";
}

container digest {
  leaf username {
    type string;
    description
      "The username for the authenticating client."
  }
  leaf password {
    type string;
    description
      "The password for the authenticating client."
  }
  description
    "The ‘digest’ HTTP scheme credentials.";
  reference
    "RFC 7616: HTTP Digest Access Authentication";
}

container hoba {
  // FIXME
  description
    "The ‘hoba’ HTTP scheme credentials."
  reference
    "RFC 7486: HTTP Origin-Bound Authentication (HOBA)";
}

container mutual {

container mutual {
  // FIXME
  description
  "The 'mutual' HTTP scheme credentials.";
  reference
  "RFC 8120: Mutual Authentication Protocol for HTTP";
}

container negotiate {
  // FIXME
  description
  "The 'negotiate' HTTP scheme credentials.";
  reference
  "RFC 4559: SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows";
}

container oauth {
  // FIXME
  description
  "The 'oauth' HTTP scheme credentials.";
  reference
  "RFC 6749: The OAuth 2.0 Authorization Framework";
}

container scram-sha-1 {
  // FIXME
  description
  "The 'scram-sha-1' HTTP scheme credentials.";
  reference
  "RFC 7804: Salted Challenge Response HTTP Authentication Mechanism";
}

container scram-sha-256 {
  // FIXME
  description
  "The 'scram-sha-256' HTTP scheme credentials.";
  reference
  "RFC 7804: Salted Challenge Response HTTP Authentication Mechanism";
}

container vapid {
  // FIXME
  description
  "The 'vapid' HTTP scheme credentials.";
  reference
  "RFC 8292: Voluntary Application Server Identification (VAPID) for Web Push";
}

grouping http-keepalives-grouping {
    description "A reusable grouping for configuring HTTP client keepalive parameters.";
    container http-keepalives {
        if-feature "http-client-keepalives";
        description "Configures the keep-alive policy, to proactively test the aliveness of the HTTP server. An unresponsive HTTP server is dropped after approximately max-wait * max-attempts seconds.";
        leaf max-wait {
            type uint16 {
                range "1..max";
            }
            units "seconds";
            default "30";
            description "Sets the amount of time in seconds after which if no data has been received from the HTTP server, a HTTP level message will be sent to test the aliveness of the HTTP server.";
        }
        leaf max-attempts {
            type uint8;
            default "3";
            description "Sets the maximum number of sequential keep-alive messages that can fail to obtain a response from the HTTP server before assuming the HTTP server is no longer alive.";
        }
    }
}

4. The HTTP Server Model

4.1. Tree Diagram

This section provides a tree diagram [RFC8340] for the "ietf-http-server" module.
module: ietf-http-server

grouping http-server-grouping
  +-- http-keepalives {http-server-keepalives}?
    +-- max-wait?       uint16
    +-- max-attempts?   uint8

grouping keepalives-grouping
  +-- http-keepalives {http-server-keepalives}?
    +-- max-wait?       uint16
    +-- max-attempts?   uint8

4.2. Example Usage

This section presents an example showing the http-server-grouping populated with some data.

  <http-keepalives>
    <max-wait>30</max-wait>
    <max-attempts>3</max-attempts>
  </http-keepalives>
</http-server>

4.3. YANG Module

This YANG module has normative references to [RFC6991].

<CODE BEGINS> file "ietf-http-server@2019-03-09.yang"
module ietf-http-server {
  yang-version 1.1;
  prefix https;

  organization
    "IETF NETCONF (Network Configuration) Working Group";
  contact
    "WG Web: <http://datatracker.ietf.org/wg/netconf/>
    WG List: <mailto:netconf@ietf.org>
    Author: Kent Watsen <mailto:kent+ietf@watsen.net>";
  description
    "This module defines reusable groupings for HTTP servers that can be used as a basis for specific HTTP server instances."

    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.

Copyright (c) 2019 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.

revision 2019-03-09 {
  description
    "Initial version";
  reference
    "RFC XXXX: YANG Groupings for HTTP Clients and HTTP Servers";
}

// Features

feature http-server-keepalives {
  description
    "Per socket HTTP keepalive parameters are configurable for HTTP servers on the server implementing this feature.";
}

// Groupings

grouping http-server-grouping {
  description
    "A reusable grouping for configuring a HTTP server, including the IP address and port number it listens for connections on.";
  uses keepalives-grouping;
}


grouping keepalives-grouping {
  description
    "A reusable grouping for configuring HTTP server keepalive parameters.";
  container http-keepalives {
    if-feature "http-server-keepalives";
    description
      "Configures the keep-alive policy, to proactively test
the aliveness of the HTTP client. An unresponsive HTTP client is dropped after approximately max-wait * max-attempts seconds.

leaf max-wait {
  type uint16 {
    range "1..max";
    units "seconds";
    default "30";
    description
    "Sets the amount of time in seconds after which if no data has been received from the HTTP client, a HTTP level message will be sent to test the aliveness of the HTTP client."
  }
}

leaf max-attempts {
  type uint8;
  default "3";
  description
  "Sets the maximum number of sequential keep-alive messages that can fail to obtain a response from the HTTP client before assuming the HTTP client is no longer alive."
}

5. Security Considerations

The YANG modules defined in this document are designed to be accessed via YANG based management protocols, such as NETCONF [RFC6241] and RESTCONF [RFC8040]. Both of these protocols have mandatory-to-implement secure transport layers (e.g., SSH, HTTP) with mutual authentication.

The NETCONF access control model (NACM) [RFC8341] provides the means to restrict access for particular users to a pre-configured subset of all available protocol operations and content.

Since the modules defined in this document only define groupings, these considerations are primarily for the designers of other modules that use these groupings.

There are a number of data nodes defined in the YANG modules that are writable/creatable/deletable (i.e., config true, which is the default). These data nodes may be considered sensitive or vulnerable in some network environments. Write operations (e.g., edit-config)
to these data nodes without proper protection can have a negative
effect on network operations. These are the subtrees and data nodes
and their sensitivity/vulnerability:

NONE

Some of the readable data nodes in the YANG modules may be considered
sensitive or vulnerable in some network environments. It is thus
important to control read access (e.g., via get, get-config, or
notification) to these data nodes. These are the subtrees and data
nodes and their sensitivity/vulnerability:

NONE

Some of the RPC operations in this YANG module may be considered
sensitive or vulnerable in some network environments. It is thus
important to control access to these operations. These are the
operations and their sensitivity/vulnerability:

NONE

6. IANA Considerations

6.1. The IETF XML Registry

This document registers two URIs in the "ns" subregistry of the IETF
XML Registry [RFC3688]. Following the format in [RFC3688], the
following registrations are requested:

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

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

6.2. The YANG Module Names Registry

This document registers two YANG modules in the YANG Module Names
registry [RFC6020]. Following the format in [RFC6020], the following
registrations are requested:
7. References

7.1. Normative References


7.2. Informative References


Watsen Expires September 10, 2019 [Page 14]


Author’s Address

Kent Watsen
Watsen Networks

EMail: kent+ietf@watsen.net