YANG Groupings for HTTP Clients and HTTP Servers
draft-kwatsen-netconf-http-client-server-04

Abstract

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

- "2019-10-18" --> 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."
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.
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 client-identity-grouping
    +-- (auth-type)
    +--:(basic)
      +-- basic (basic-auth)?
      +-- user-id string
      +-- password string

  grouping http-client-grouping
    +-- protocol-version? enumeration
    +-- client-identity
      | +---u client-identity-grouping
      +-- proxy-server! (proxy-connect)?
      | +---u tcp-client-parameters
      | +---u tls-client-parameters
      | +---u tlsc:tls-client-grouping
      +-- proxy-client-identity
        +-- u client-identity-grouping
```

3.2. Example Usage

This section presents an example showing the http-client-grouping populated with some data.
  <protocol-version>HTTP/1.1</protocol-version>
  <client-identity>
    <basic>
      <user-id>bob</user-id>
      <password>secret</password>
    </basic>
  </client-identity>
</http-client>

3.3. YANG Module

This YANG module has normative references to [RFC6991].

<CODE BEGINS> file "ietf-http-client@2019-10-18.yang"

module ietf-http-client {
  yang-version 1.1;
  prefix httpc;

  import ietf-tcp-client {
    prefix tcpc;
    reference
      "RFC AAAA: YANG Groupings for TCP Clients and TCP Servers";
  }

  import ietf-tls-client {
    prefix tlsc;
    reference
      "RFC BBBB: YANG Groupings for TLS Clients and TLS Servers";
  }

  import ietf-netconf-acm {
    prefix nacm;
    reference
      "RFC 8341: Network Configuration Access Control Model";
  }

  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.

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 (https://trustee.ietf.org/license-info).

This version of this YANG module is part of RFC XXXX (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself for full legal notices.


revision 2019-10-18 {
  description
    "Initial version";
  reference
    "RFC XXXX: YANG Groupings for HTTP Clients and HTTP Servers";
}

// Features

feature proxy-connect {
  description
    "Proxy connection configuration is configurable for HTTP clients on the server implementing this feature.";
}

feature basic-auth {
  description
    "The ‘basic-auth’ feature indicates that the client may be configured to use the ‘basic’ HTTP authentication scheme.";
  reference
    "RFC 7617: The ‘Basic’ HTTP Authentication Scheme";
}
// Groupings

grouping client-identity-grouping {
  description
  "The credentials used by the client to authenticate to
  the HTTP server.";
  choice auth-type {
    nacm:default-deny-write;
    mandatory true;
    description
    "The authentication type.";
    case basic {
      container basic {
        if-feature "basic-auth";
        leaf user-id {
          type string;
          mandatory true;
          description
          "The user-id for the authenticating client.";
        }
        leaf password {
          nacm:default-deny-all;
          type string;
          mandatory true;
          description
          "The password for the authenticating client.";
        }
      }
    }
  }
  description
  "The 'basic' HTTP scheme credentials.";
  reference
  "RFC 7617: The 'Basic' HTTP Authentication Scheme";
}
}
} // grouping client-identity-grouping


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.

  Note that this grouping uses fairly typical descendent
  node names such that a stack of 'uses' statements will
  have name conflicts. It is intended that the consuming
  data model will resolve the issue (e.g., by wrapping
  the 'uses' statement in a container called
  'http-client-parameters'). This model purposely does
Internet-Draft   Groupings for HTTP Clients and Servers     October 2019

not do this itself so as to provide maximum flexibility to consuming models.

leaf protocol-version {
    nacm:default-deny-write;
    type enumeration {
        enum HTTP/1.0 {
            description
            "The client should use the 'HTTP/1.0' protocol."
        }
        enum HTTP/1.1 {
            description
            "The client should use the 'HTTP/1.1' protocol."
        }
        enum HTTP/2.0 {
            description
            "The client should use the 'HTTP/2.0' protocol."
        }
    }
    description
        "The HTTP protocol version the client should use."
}

container client-identity {
    description
        "The identity the HTTP client should use when authenticating itself to the HTTP server."
        uses client-identity-grouping;
}

container proxy-server {
    nacm:default-deny-write;
    if-feature "proxy-connect";
    presence true; // only so ex-http-client can pass validation?
    container tcp-client-parameters {
        description
            "A wrapper around the TCP parameters to avoid name collisions."
            uses "tcpc:tcp-client-grouping";
    }
    container tls-client-parameters {
        description
            "A wrapper around the TLS parameters to avoid name collisions."
            uses "tlsc:tls-client-grouping";
    }
    container proxy-client-identity {
        description
"The identity the HTTP client should use when authenticating itself to the HTTP server."
uses client-identity-grouping;
}
description
"Proxy server settings.";
}
} // grouping http-client-grouping

} // module ietf-http-client

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
  +-- server-name?      string
  +-- protocol-versions
    |  +-- protocol-version*  enumeration
  +-- client-authentication!
    +-- (required-or-optional)
      |    +--:(required)
      |    |    +-- required?               empty
      |    +--:(optional)
      |        +-- optional?              empty
    +-- (local-or-external)
      +--:(local) {local-client-auth-supported}?
        +-- users
          |    +-- user* [user-id]
          |    |    +-- user-id?  string
          |    |    +-- (auth-type)?
          |    |        +--:(basic)
          |    |        |    +-- basic {basic-auth}?
          |    |        |    |    +-- user-id?  string
          |    |        |    |    +-- password?  ianach:crypt-hash
          |    |        +-- :(external) {external-client-auth-supported}?
          |    |        +-- client-auth-defined-elsewhere?  empty
4.2. Example Usage

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

```xml
  <server-name>foo.example.com</server-name>
  <protocol-versions>
    <protocol-version>HTTP/1.1</protocol-version>
    <protocol-version>HTTP/2.0</protocol-version>
  </protocol-versions>
  <client-authentication>
    <required/>
    <client-auth-defined-elsewhere/>
  </client-authentication>
</http-server>
```

4.3. YANG Module

This YANG module has normative references to [RFC6991].

```xml
<CODE BEGINS> file "ietf-http-server@2019-10-18.yang"

module ietf-http-server {
  yang-version 1.1;
  prefix https;

  import iana-crypt-hash {
    prefix ianach;
    reference "RFC 7317: A YANG Data Model for System Management";
  }

  import ietf-netconf-acm {
    prefix nacm;
    reference "RFC 8341: Network Configuration Access Control Model";
  }

  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>";

```
This module defines reusable groupings for HTTP servers that can be used as a basis for specific HTTP server instances.

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 (https://trustee.ietf.org/license-info).

This version of this YANG module is part of RFC XXXX (https://www.rfc-editor.org/info/rfcXXXX); see the RFC itself for full legal notices.;

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 (RFC 2119) (RFC 8174) when, and only when, they appear in all capitals, as shown here.;

revision 2019-10-18 {
  description
    "Initial version";
  reference
    "RFC XXXX: YANG Groupings for HTTP Clients and HTTP Servers";
}

// Features

feature local-client-auth-supported {
  description
    "Indicates that the HTTP server supports local configuration of client credentials.";
}

feature external-client-auth-supported {
  description
    "Indicates that the HTTP server supports external configuration of client credentials.";
}

feature basic-auth {
  description
"The ‘basic-auth’ feature indicates that the server may be configured authenticate users using the ‘basic’ HTTP authentication scheme."

reference

"RFC 7617: The ‘Basic’ HTTP Authentication Scheme";

// Groupings

grouping http-server-grouping {
  description
  "A reusable grouping for configuring an HTTP server.

  Note that this grouping uses fairly typical descendent node names such that a stack of ‘uses’ statements will have name conflicts. It is intended that the consuming data model will resolve the issue (e.g., by wrapping the ‘uses’ statement in a container called ‘http-server-parameters’). This model purposely does not do this itself so as to provide maximum flexibility to consuming models.");

  leaf server-name {
    nacm:default-deny-write;
    type string;
    description
      "The value of the ‘Server’ header field. If not set, then underlying software’s default value is used. Set to the empty string to disable.";
  }

  container protocol-versions {
    nacm:default-deny-write;
    description
      "A list of HTTP protocol versions supported by this server.");
    leaf-list protocol-version {
      type enumeration {
        enum "HTTP/1.0" {
          description
            "The server supports the ‘HTTP/1.0’ protocol.";
        }
        enum "HTTP/1.1" {
          description
            "The server supports the ‘HTTP/1.1’ protocol.";
        }
        enum "HTTP/2.0" {
          description
            "The server supports the ‘HTTP/2.0’ protocol.";
        }
      }
    }
  }
}
description
"The server supports the 'HTTP/2.0' protocol.";

description
"An HTTP protocol version supported by this server.";

container client-authentication {
    nacm:default-deny-write;
    presence
    "Indicates that HTTP based client authentication is supported (i.e., the server will request that the HTTP client send authenticate when needed). This is needed as some HTTP-based protocols may only support, e.g., TLS-level client authentication.";
    description
    "Specifies if HTTP client authentication is required or optional, and specifies if the credentials needed to authenticate the HTTP client are configured locally or externally.";
    choice required-or-optional {
        mandatory true;  // or default to 'required' ?
        description
        "Indicates if HTTP-level client authentication is required or optional. This is necessary for some protocols (e.g., RESTCONF) that may optionally authenticate a client via TLS-level authentication, HTTP-level authentication, or both simultaneously).";
        leaf required {
            type empty;
            description
            "Indicates that HTTP-level client authentication is required to access protected resources.";
        }
        leaf optional {
            type empty;
            description
            "Indicates that HTTP-level client authentication is optional to access protected resources.";
        }
    }
    choice local-or-external {
        mandatory true;
        description
        "Indicates if the client credentials are configured locally or externally. The need to support external
configuration for client authentication stems from the desire to support consuming data models that prefer to place client authentication with client definitions, rather than in a data model principally concerned with configuring the transport.

case local {
  if-feature "local-client-auth-supported";
  description
    "Client credentials are configured locally.";
  container users {
    description
      "A list of locally configured users.";
    list user {
      key user-id;
      description
        "The list of local users configured on this device.";
      leaf user-id {
        type string;
        description
          "The user-id for the authenticating client.";
      }
      choice auth-type {
        description
          "The authentication type.";
        container basic {
          if-feature "basic-auth";
          leaf user-id {
            type string;
            description
              "The user-id for the authenticating client.";
          }
          leaf password {
            nacm:default-deny-write;
            type ianach:crypt-hash;
            description
              "The password for the authenticating client.";
          }
        }
        description
          "The 'basic' HTTP scheme credentials.";
        reference
          "RFC 7617:
            The 'Basic' HTTP Authentication Scheme";
      }
    }
  }
}

case external {

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:

FIXME: (pending - TBD)

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:

FIXME: (pending client auth params?)

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:

The modules defined in this document do not define any 'RPC' or 'action' statements.

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:

name: ietf-http-client
prefix: httpc
reference: RFC XXXX

name: ietf-http-server
prefix: https
reference: RFC XXXX
7. References

7.1. Normative References


7.2. Informative References


Internet-Draft   Groupings for HTTP Clients and Servers   October 2019

BCP 215, RFC 8340, DOI 10.17487/RFC8340, March 2018,

Author’s Address

Kent Watsen
Watsen Networks

EMail: kent+ietf@watsen.net