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:

- "2019-04-07" --> 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 October 9, 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.
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-parameters
  |  +-- protocol-version?  enumeration
  |  +-- client-identity
  |  |  +-- (auth-type)?
  |  |  |  +--:(basic)
  |  |  |  |  +-- basic {basic-auth}? 
  |  |  |  |  |  +-- user-id?  string
  |  |  |  |  |  +-- password?  string
  |  |  |  +--:(bearer)
  |  |  |  |  +-- bearer {bearer-auth}? 
  |  |  |  |  |  +-- token?  string
  |  |  |  +--:(digest)
  |  |  |  |  +-- digest {digest-auth}? 
  |  |  |  |  |  +-- username?  string
  |  |  |  |  |  +-- password?  string
  |  |  |  +--:(hoba)
  |  |  |  |  +-- hoba {hoba-auth}? 
  |  |  |  +--:(mutual)
  |  |  |  |  +-- mutual {mutual-auth}? 
  |  |  |  +--:(negotiate)
  |  |  |  |  +-- negotiate {negotiate-auth}? 
  |  |  |  +--:(oauth)
  |  |  |  |  +-- oauth {oauth-auth}? 
  |  |  |  +--:(scram-sha-1)
  |  |  |  |  +-- scram-sha-1 {scram-sha-1-auth}? 
  |  |  |  +--:(scram-sha-256)
  |  |  |  |  +-- scram-sha-256 {scram-sha-256-auth}? 
  |  |  |  +--:(vapid)
  |  |  |  |  +-- vapid {vapid-auth}?
  +-- proxy-server! {proxy-connect}?
  +-- tcpc:tcp-client-grouping
  +-- tlsc:tls-client-grouping
  +-- proxy-client-identity
  |  +-- user-id?  string
  |  +-- password?  string

3.2. Example Usage

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

3.3. YANG Module

This YANG module has normative references to [RFC6991].

<CODE BEGINS> file "ietf-http-client@2019-04-07.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";
    }

    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.

Watsen Expires October 9, 2019 [Page 5]
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 2019-04-07 {
  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
    "fixme";
}

feature bearer-auth {
  description
    "fixme";
}

feature digest-auth {
  description
    "fixme";
}
feature hoba-auth {
    description
    "fixme";
}

feature mutual-auth {
    description
    "fixme";
}

feature negotiate-auth {
    description
    "fixme";
}

feature oauth-auth {
    description
    "fixme";
}

feature scram-sha-1-auth {
    description
    "fixme";
}

feature scram-sha-256-auth {
    description
    "fixme";
}

feature vapid-auth {
    description
    "fixme";
}

// 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.";

    container http-client-parameters {
        description
        "A container to hold HTTP client configuration. As HTTP clients vary wildly, it is expected that this container will be augmented by application-level modules as needed.";
    }
}
leaf protocol-version {
  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.";
} // leaf protocol-version

container client-identity {
  description
    "The credentials used by the client to authenticate to the HTTP server.";
  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 {
        type string;
        description
          "The password for the authenticating client.";
      }
    }
    description
      "The ‘basic’ HTTP scheme credentials.";
    reference
      "RFC 7617: The ‘Basic’ HTTP Authentication Scheme";
  }
  container bearer {
    if-feature "bearer-auth";
    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 {
  if-feature "digest-auth";
  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 {
  if-feature "hoba-auth";
  // FIXME
  description
    "The 'hoba' HTTP scheme credentials.";
  reference
    "RFC 7486: HTTP Origin-Bound Authentication (HOBA)";
}
container mutual {
  if-feature "mutual-auth";
  // FIXME
  description
    "The 'mutual' HTTP scheme credentials.";
  reference
    "RFC 8120: Mutual Authentication Protocol for HTTP";
}
container negotiate {
  if-feature "negotiate-auth";
  // FIXME
  description
    "The 'negotiate' HTTP scheme credentials.";
reference
   "RFC 4559: SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows";
"
container oauth {
   if-feature "oauth-auth";
   // FIXME
   description
   "The ‘oauth’ HTTP scheme credentials.";
   reference
   "RFC 6749: The OAuth 2.0 Authorization Framework";
} container scram-sha-1 {
   if-feature "scram-sha-1-auth";
   // FIXME
   description
   "The ‘scram-sha-1’ HTTP scheme credentials.";
   reference
   "RFC 7804: Salted Challenge Response HTTP Authentication Mechanism";
} container scram-sha-256 {
   if-feature "scram-sha-256-auth";
   // FIXME
   description
   "The ‘scram-sha-256’ HTTP scheme credentials.";
   reference
   "RFC 7804: Salted Challenge Response HTTP Authentication Mechanism";
} container vapid {
   if-feature "vapid-auth";
   // FIXME
   description
   "The ‘vapid’ HTTP scheme credentials.";
   reference
   "RFC 8292: Voluntary Application Server Identification (VAPID) for Web Push";
}
} // container client-identity

container proxy-server {
   if-feature "proxy-connect";
   presence true; // only so ex-http-client can pass validation?
   uses "tcpc:tcp-client-grouping";
   uses "tlsc:tls-client-grouping";
   container proxy-client-identity {

} // container proxy-server
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-server-parameters
  |    +-- server-name?         string
  |    +-- protocol-versions
  |    |    +-- protocol-version*   enumeration

4.2. Example Usage

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

4.3. YANG Module

This YANG module has normative references to [RFC6991].

<CODE BEGINS> file "ietf-http-server@2019-04-07.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.

    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

    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."


Watsen                   Expires October 9, 2019               [Page 12]
'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 2019-04-07 {
  description
  "Initial version";
  reference
  "RFC XXXX: YANG Groupings for HTTP Clients and HTTP Servers";
}

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

container http-server-parameters {
  description
  "A container that holds HTTP server configuration. As HTTP servers vary wildly, it is expected that this container will
  be augmented by application-level modules as needed.";

  leaf server-name {
    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 {
  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
"An HTTP protocol version supported by this server."
}
}

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


Author’s Address

Kent Watsen
Watsen Networks

EMail: kent+ietf@watsen.net