

Independent Submission
Internet-Draft
Intended status: Standards Track
Expires: November 30, 2018

K. Murchison
FastMail
May 29, 2018

A JSON Meta Application Protocol (JMAP) Subprotocol for WebSocket
draft-murchison-jmap-websocket-00

Abstract

This document defines a binding for the JSON Meta Application Protocol (JMAP) over a WebSocket transport layer. A WebSocket binding for JMAP provides higher performance than the current HTTP binding for JMAP.

Open Issues

- o Discovering support for JMAP over WebSocket.
- o Should we allow push notifications over the WS connection?

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 November 30, 2018.

Copyright Notice

Copyright (c) 2018 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents

carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	2
2. Conventions Used in This Document	2
3. JMAP Subprotocol	3
3.1. Handshake	3
3.2. WebSocket Messages	3
3.3. Examples	4
4. Discovering Support for JMAP over WebSocket	6
5. Security Considerations	7
6. Privacy Considerations	7
7. IANA Considerations	7
7.1. Registration of the WebSocket JMAP Sub-Protocol	7
8. References	7
8.1. Normative References	7
8.2. Informative References	8
8.3. URIs	8
Author's Address	8

1. Introduction

JMAP over HTTP requires that every JMAP API request be authenticated. Depending on the type of authentication used by the JMAP client and the configuration of the JMAP server, authentication could be an expensive operation both in time and resources. In such circumstances, authenticating every JMAP API request may harm performance.

The WebSocket binding for JMAP eliminates this performance hit by authenticating just the WebSocket handshake request and having those credentials remain in effect for the duration of the WebSocket connection.

2. Conventions Used in This Document

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 [1] [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. JMAP Subprotocol

The term WebSocket subprotocol refers to an application-level protocol layered on top of a WebSocket connection. This document specifies the WebSocket JMAP subprotocol for carrying JMAP API requests and responses through a WebSocket connection. Binary data MUST NOT be uploaded or downloaded through a WebSocket JMAP connection.

3.1. Handshake

The JMAP WebSocket client and JMAP WebSocket server negotiate the use of the WebSocket JMAP subprotocol during the WebSocket handshake, either via a HTTP/1.1 Upgrade request (see Section 1.3 of [RFC6455]) or a HTTP/2 Extended CONNECT request (see Section 5 of [I-D.ietf-httpbis-h2-websockets]).

Regardless of the method used for the WebSocket handshake, the client MUST make an authenticated [RFC7235] HTTP request on the JMAP 'apiURL' (see Section 2 of [I-D.ietf-jmap-core]), and the client MUST include the value 'jmap' in the list of protocols for the 'Sec-WebSocket-Protocol' header field. The reply from the server MUST also contain 'jmap' in its corresponding 'Sec-WebSocket-Protocol' header field in order for a JMAP subprotocol connection to be established.

If a client receives a handshake response that does not include 'jmap' in the 'Sec-WebSocket-Protocol' header, then a JMAP subprotocol WebSocket connection was not established and the client MUST close the WebSocket connection.

Once the handshake has successfully completed, the WebSocket connection is established and can be used for JMAP API requests and responses. Messages other than JMAP API requests and responses MUST NOT be transmitted over this connection.

The credentials used for authenticating the HTTP request to initiate the handshake remain in effect for the duration of the WebSocket connection.

3.2. WebSocket Messages

Data frame messages in the JMAP subprotocol MUST be of the text type and contain UTF-8 encoded data. The messages MUST be in the form of a single JMAP request object (see Section 3.2 of [I-D.ietf-jmap-core]) when sent from the client to the server, and in the form of a single JMAP Response object (see Section 3.4 of [I-D.ietf-jmap-core]) when sent from the server to the client.

3.3. Examples

The following examples show WebSocket JMAP handshakes and a subsequent JMAP for Mail [I-D.ietf-jmap-mail] request and response. The examples assume that the JMAP 'apiURL' has been advertised in the JMAP Session object as '/jmap/'. Note that folding of header fields is for editorial purposes only.

WebSocket JMAP handshake via HTTP/1.1 which also negotiates compression [RFC7692]:

```
[[ From Client ]]                [[ From Server ]]  
  
GET /jmap/ HTTP/1.1  
Host: server.example.com  
Upgrade: websocket  
Connection: Upgrade  
Authorization: Zm9vOmJhcg==  
Sec-WebSocket-Key:  
    dGh1IHNhbXBsZSBub25jZQ==  
Sec-WebSocket-Protocol: jmap  
Sec-WebSocket-Version: 13  
Sec-WebSocket-Extensions:  
    permessage-deflate  
Origin: http://www.example.com  
  
HTTP/1.1 101 Switching Protocols  
Upgrade: websocket  
Connection: Upgrade  
Sec-WebSocket-Accept:  
    s3pPLMBiTxaQ9kYGzzhZRbK+xOo=  
Sec-WebSocket-Protocol: jmap  
Sec-WebSocket-Extensions:  
    permessage-deflate  
  
[WebSocket connection established]  
  
{  
  "using": [ "urn:ietf:params:jmap:core",  
            "urn:ietf:params:jmap:mail" ],  
  "methodCalls": [  
    ...  
  ]  
}  
  
  {  
    "methodResponses": [  
      ...  
    ]  
  }
```

WebSocket JMAP handshake on a HTTP/2 stream:

```

[[ From Client ]]                [[ From Server ]]

                                SETTINGS
                                SETTINGS_ENABLE_CONNECT_PROTOCOL = 1

HEADERS + END_HEADERS
:method = CONNECT
:protocol = websocket
:scheme = https
:path = /jmap/
:authority = server.example.com
authorization = Zm9vOmJhcg==
sec-websocket-protocol = jmap
sec-websocket-version = 13
origin = http://www.example.com

                                HEADERS + END_HEADERS
                                :status = 200
                                sec-websocket-protocol = jmap

[WebSocket connection established]

DATA
{
  "using": [ "urn:ietf:params:jmap:core",
            "urn:ietf:params:jmap:mail" ],
  "methodCalls": [
    ...
  ]
}

                                DATA
                                {
                                  "methodResponses": [
                                    ...
                                  ]
                                }

```

4. Discovering Support for JMAP over WebSocket

TODO

- o Do we define a new capability for the Session resource?
- o Do we simply return an Upgrade:websocket header field in HTTP/1.1 responses to API requests?

- o How would we advertise via HTTP/2 protocol as Alt-Svc is per origin, NOT per-source?

5. Security Considerations

TODO

6. Privacy Considerations

TODO

7. IANA Considerations

7.1. Registration of the WebSocket JMAP Sub-Protocol

This specification requests IANA to register the WebSocket JMAP sub-protocol under the "WebSocket Subprotocol Name" Registry with the following data:

Subprotocol Identifier: JMAP

Subprotocol Common Name: WebSocket Transport for JMAP (JSON Meta Application Protocol)

Subprotocol Definition: RFCXXXX (this document)

8. References

8.1. Normative References

- [I-D.ietf-httpbis-h2-websockets]
McManus, P., "Bootstrapping WebSockets with HTTP/2", draft-ietf-httpbis-h2-websockets-05 (work in progress), May 2018.
- [I-D.ietf-jmap-core]
Jenkins, N., "JSON Meta Application Protocol", draft-ietf-jmap-core-05 (work in progress), May 2018.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.
- [RFC6455] Fette, I. and A. Melnikov, "The WebSocket Protocol", RFC 6455, DOI 10.17487/RFC6455, December 2011, <<https://www.rfc-editor.org/info/rfc6455>>.

- [RFC7235] Fielding, R., Ed. and J. Reschke, Ed., "Hypertext Transfer Protocol (HTTP/1.1): Authentication", RFC 7235, DOI 10.17487/RFC7235, June 2014, <<https://www.rfc-editor.org/info/rfc7235>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

8.2. Informative References

- [I-D.ietf-jmap-mail]
Jenkins, N., "JMAP for Mail", draft-ietf-jmap-mail-05 (work in progress), May 2018.
- [RFC7692] Yoshino, T., "Compression Extensions for WebSocket", RFC 7692, DOI 10.17487/RFC7692, December 2015, <<https://www.rfc-editor.org/info/rfc7692>>.

8.3. URIs

- [1] <https://tools.ietf.org/html/bcp14>

Author's Address

Kenneth Murchison
FastMail US LLC

Email: murch@fastmailteam.com