Internet Engineering Task Force (IETF)
Request for Comments: 9698
Category: Standards Track
ISSN: 2070-1721

A. Gulbrandsen
ICANN
B. Gondwana
Fastmail

January 2025

The JMAPACCESS Extension for IMAP

Abstract

This document defines an IMAP extension to let clients know that the messages in this IMAP server are also available via the JSON Meta Application Protocol (JMAP), and how. It is intended for clients that want to migrate gradually to JMAP or use JMAP extensions within an IMAP client.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc9698.

Table of Contents

1. Introduction

An IMAP server can declare that the messages in its mailstore are also available via JMAP. For simplicity, only a complete equivalence is supported (the same set of messages are available via both IMAP and JMAP).

2. Requirements Language

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. Details

By advertising the JMAPACCESS capability, the server asserts that if a mailbox or message has a particular object ID when accessed via either IMAP or JMAP (see [RFC3501], [RFC9051], and [RFC8620]), then the same mailbox or message is accessible via the other protocol, and it has the same ID.

The server MUST also advertise the OBJECTID extension, defined by [RFC8474]. The JMAP session resource that allows access to the same messages is called "the JMAP server" below.

This specification does not affect message lifetime: If a client accesses a message via IMAP and half a second later via JMAP, then the message may have been deleted between the two accesses.

When the server processes the client's LOGIN/AUTHENTICATE command and enters Authenticated state, the server considers the way the client authenticated. If the IMAP server can infer from the client's authentication process that its credentials suffice to authenticate via JMAP, then the server MUST include a JMAPACCESS capability in any capability list sent after that point. This includes the capability list that some servers send immediately when authentication succeeds.

Servers are encouraged to report the same message flags and other data via both protocols, as far as possible.

This specification does not require mailboxes to have the same name in IMAP and JMAP, even if they share a mailbox ID. However, the JMAP specification regulates that in the text about the name and role properties described in Section 2 of [RFC8620].

Note that all JMAP servers support internationalized email addresses (see [RFC6530]). If this IMAP server does not or if the IMAP client does not issue ENABLE UTF8=ACCEPT (see [RFC6855]), then it is possible that the client will receive accurate address fields via JMAP and downgraded fields via IMAP (see [RFC6857] and [RFC6858] for examples). Issuing ENABLE UTF8=ACCEPT is a simple way to sidestep the issue.

4. The GETJMAPACCESS Command and the JMAPACCESS Response

The GETJMAPACCESS command requests that the server respond with the session URL for the JMAP server that provides access to the same mail.

If such a JMAP server is known to this server, the server MUST respond with an untagged JMAPACCESS response containing the JMAP server's session resource (a URL) followed by a tagged OK response.

If such a JMAP server is not known, the server MUST respond with a tagged BAD response (and MUST NOT include JMAPACCESS in the capability list).

The JMAPACCESS response is followed by a single link to a JMAP session resource.

The formal syntax in [RFC9051] is extended as follows:

```
command-auth =/ "GETJMAPACCESS"

mailbox-data =/ resp-jmapaccess

resp-jmapaccess = "JMAPACCESS" SP quoted
```

The syntax in [RFC3501] is extended similarly (this extension may be used with IMAP4rev1 as well as IMAP4rev2).

5. Examples

Lines sent by the client are preceded by C: and lines sent by the server are preceded by S:. Each example starts with the IMAP banner issued by the server on connection, and generally abbreviates the capability lists to what’s required by the example itself.

Real connections use longer capability lists, much longer AUTHENTICATE arguments and of course use TLS. However, these examples focus on JMAPACCESS.

Example 1:

A client connects, sees that SASL OAuth [RFC7628] is available, and authenticates in that way.

```
S: * OK [CAPABILITY IMAP4rev1 AUTH=OAUTHBEARER SASL-IR] example1
C: 1 AUTHENTICATE OAUTHBEARER bixhPXVzZ...QEB
```

The server processes the command successfully. It knows that the client used OAuth, and that it and its JMAP alter ego use the same OAuth backend subsystem. Because of that it infers that the (next) access token is just as usable via JMAP as via IMAP. It includes a JMAPACCESS capability in its reply (again, real capability lists are much longer):

```
S: 1 OK [CAPABILITY IMAP4rev1 JMAPACCESS] done
C: 1b GETJMAPACCESS
S: * JMAPACCESS "https://example.com/.well-known/jmap"
S: 1b OK done
```

SASL OAuth is specified by [RFC7628], and the argument in this example is abbreviated from the more realistic length used in RFC 7628.

Example 2:

A client connects, sees no SASL method it recognizes, and issues a LOGIN command.

```
S: * OK [CAPABILITY IMAP4rev2] example2
C: 2 LOGIN "arnt" "trondheim"
```

The server sees that the password is accepted, knows that it and its JMAP alter ego use the same password database, and issues a JMAPACCESS capability:

```
   S: * OK [CAPABILITY IMAP4rev2 JMAPACCESS] done
   S: 2 OK done
   C: 2b JMAPACCESS
   S: * JMAPACCESS "https://example.com/.well-known/jmap"
```

S: 2b OK done

The URL uses the same quoting rules as most other IMAP strings.

Example 3:

A client connects, sees no SASL method it recognizes, and issues a LOGIN command with a correct password.

```
S: * OK [CAPABILITY IMAP4rev1 IMAP4rev2] example3
C: 3 LOGIN "arnt" "trondheim"
```

The server operator has decided to disable password use with JMAP, but allow it for a while with IMAP to cater to older clients. Therefore, the login succeeds, but there is no JMAPACCESS capability.

```
S: 3 OK done
```

Example 4:

A client connects, sees no SASL method it recognizes, and issues a LOGIN command. Its password is incorrect.

```
S: * OK [CAPABILITY IMAP4rev2 AUTH=GSS] example4
C: 4 LOGIN "arnt" "oslo"
```

The server does not enter Authenticated state, so nothing requires it to mention JMAPACCESS. It replies curtly:

```
S: 4 NO done
```

6. IANA Considerations

The IANA has added the JMAPACCESS capability to the "Internet Message Access Protocol (IMAP) Capabilities Registry" and listed this document as the reference.

7. Security Considerations

JMAPACCESS reveals to authenticated IMAP clients that they would be able to authenticate via JMAP using the same credentials and that the object IDs match.

One does not normally reveal anything at all about authentication. However, if the client is an attacker, then the attacker is known to have valid credentials, and Section 2.2 of [RFC8620] tells the attacker how to find the revealed URL without the help of this extension. Therefore, it is believed that this document does not benefit an attacker noticeably, and its value for migration far outweighs its risk.

8. References

8.1. Normative References

[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>.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, DOI 10.17487/RFC3501, March 2003, <https://www.rfc-editor.org/info/rfc3501>.

[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>.

[RFC8474] Gondwana, B., Ed., "IMAP Extension for Object Identifiers", RFC 8474, DOI 10.17487/RFC8474, September 2018, <https://www.rfc-editor.org/info/rfc8474>.

[RFC9051] Melnikov, A., Ed. and B. Leiba, Ed., "Internet Message Access Protocol (IMAP) - Version 4rev2", RFC 9051, DOI 10.17487/RFC9051, August 2021, <https://www.rfc-editor.org/info/rfc9051>.

8.2. Informative References

[RFC6530] Klensin, J. and Y. Ko, "Overview and Framework for Internationalized Email", RFC 6530, DOI 10.17487/RFC6530, February 2012, <https://www.rfc-editor.org/info/rfc6530>.

[RFC6855] Resnick, P., Ed., Newman, C., Ed., and S. Shen, Ed., "IMAP Support for UTF-8", RFC 6855, DOI 10.17487/RFC6855, March 2013, <https://www.rfc-editor.org/info/rfc6855>.

[RFC6857] Fujiwara, K., "Post-Delivery Message Downgrading for Internationalized Email Messages", RFC 6857, DOI 10.17487/RFC6857, March 2013, <https://www.rfc-editor.org/info/rfc6857>.

[RFC6858] Gulbrandsen, A., "Simplified POP and IMAP Downgrading for Internationalized Email", RFC 6858, DOI 10.17487/RFC6858, March 2013, <https://www.rfc-editor.org/info/rfc6858>.

[RFC7628] Mills, W., Showalter, T., and H. Tschofenig, "A Set of Simple Authentication and Security Layer (SASL) Mechanisms for OAuth", RFC 7628, DOI 10.17487/RFC7628, August 2015, <https://www.rfc-editor.org/info/rfc7628>.

[RFC8620] Jenkins, N. and C. Newman, "The JSON Meta Application Protocol (JMAP)", RFC 8620, DOI 10.17487/RFC8620, July 2019, <https://www.rfc-editor.org/info/rfc8620>.

Authors' Addresses

Arnt Gulbrandsen
ICANN
6 Rond Point Schumann, Bd. 1
1040 Brussels
Belgium
Email: arnt@gulbrandsen.priv.no
URI: https://icann.org/ua

Bron Gondwana
Fastmail
Level 2, 114 William St.
Melbourne VIC 3000
Australia
Email: brong@fastmailteam.com
URI: https://fastmail.com