JMAP for Quotas
draft-ietf-jmap-quotas-00

Abstract

   This document specifies a data model for handling quotas on mailboxes
   with a server using JMAP.

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 March 8, 2020.

Table of Contents

1. Introduction

   JMAP ([RFC8620] - JSON Meta Application Protocol) is a generic
   protocol for synchronising data, such as mail, calendars or contacts,
   between a client and a server. It is optimised for mobile and web
   environments, and aims to provide a consistent interface to different
   data types.

   This specification defines a data model for handling mail quotas over
   JMAP, allowing you to read and explain quota information.

   This specification does not address quota administration, which
   should be handled by other means.

1.1. Notational conventions

   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.

   Type signatures, examples and property descriptions in this document
   follow the conventions established in section 1.1 of [RFC8620]. Data

   types defined in the core specification are also used in this
   document.

   Servers MUST support all properties specified for the new data types
   defined in this document.

1.2. Terminology

1.2.1. Quota

   A quota is a numeric upper limit that the server is enforcing.
   Quotas are applied to JMAP Mailbox objects.

1.3. Addition to the capabilities object

   The capabilities object is returned as part of the JMAP Session
   object; see [RFC8620], section 2.

   This document defines one additional capability URI.

1.3.1. urn::ietf::params::jmap::quota

   This represents support for the Quota data type and associated API
   methods. Servers supporting this specification MUST add this
   property to the capabilities object.

2. Quota definition

   The quota is an object that displays the limit set to a mailbox usage
   as well as the current usage in regard to that limit.

2.1. The Scope Data Type

   The *Scope* is a "String" from an enumeration defined list of values,
   handled by the server.

   It explains the entities this value applies to. Some custom
   specifications might provide some additional values. If the client
   does not specify custom scope specifications in the "using" parameter
   of the request, the server should respond the JSON value "null",
   instead of answering a scope value that the client does not support.
   Standard values are: * "account": Applies for this account *
   "domain": All users of this domain share this part of the quota *
   "global": All users of this mail server share this part of the quota

2.2. The Quota Object

The quota object MUST contain the following fields:

o *id*: "Id" The unique identifier for this object. It should
   respect the JMAP ID datatype defined in section 1.2 of [RFC8620]

o *type*: "String" The type of quota. It should either be
   "messageCount" or "messageStorageSize". There is two ways we can
   define a quota: either on a number of emails, or in the amount of
   bytes.

o *used*: "UnsignedInt" The current usage of the mailbox.
   Computation of this value is handled by the server.

o *usedScope*: "Scope" The "Scope" of the current mailbox usage.

o *limit*: "UnsignedInt" The limit set by this quota object to the
   current mailbox.

o *limitScope*: "Scope" The "Scope" of the mailbox limit.

The quota object MAY contain the following field:

o *description*: "String|null" Arbitrary free, human readable,
   description of this quota. Might be used to explain where the
   limit comes from and explain the entities this quota applies to.

2.3. Example

```
{
  "id": "2a06df0d-9865-4e74-a92f-74dcc814270e",
  "type": "messageCount",
  "used": 1056,
  "usedScope": "account",
  "limit": 2000,
  "limitScope": "domain",
  "description": "Personal account usage, bounded by a domain limit"
}
```

2.4. Quota/get

Standard "/get" method as described in [RFC8620] section 5.1. The
ids argument may be "null" to fetch all at once.

2.4.1. **Example**

   Request fetching all quotas related to an account :

```
                     [[ "Quota/get", {
                       "accountId": "u33084183",
                       "ids": null
                     }, "0" ]]
```

   With response :

```
[[ "Quota/get", {
  "accountId": "u33084183",
  "state": "78540",
  "list": [{
  "id": "2a06df0d-9865-4e74-a92f-74dcc814270e",
  "type": "messageCount",
  "used": 1056,
  "usedScope": "account",
  "limit": 2000,
  "limitScope": "domain",
  "description": "Personal account usage, bounded by a domain limit"
  }, {
    "id": "3b06df0e-3761-4s74-a92f-74dcc963501x",
    "type": "messageStorageSize",
    ...
  }, ...],
  "notFound": []
}, "0" ]]
```

2.5. **Quota/changes**

   Standard "/changes" method as described in [RFC8620] section 5.2.

2.5.1. **Example**

   Request:

```
                     [[ "Quota/changes", {
                       "accountId": "u33084183",
                       "sinceState": "10824",
                       "maxChanges": 20
                     }, "0" ]
```

   Response:

```
        [[ "Quota/changes", {
          "accountId": "u33084183",
          "oldState": "10824",
          "newState": "10826",
          "hasMoreChanges": false,
          "created": [],
          "updated": ["2a06df0d-9865-4e74-a92f-74dcc814270e"],
          "destroyed": []
        }, "0" ],
        ... ]
```

3. Addition to the Mailbox object

The following extra property is defined:

o *quotaIds:* "Id[]" (default: "[]") A list of quota ids bound to
   that mailbox, or "[]" if that mailbox has no quota restrictions.
   This value is server-set.

4. Security considerations

All security considerations of JMAP ([RFC8620]) apply to this
specification.

5. IANA Considerations

5.1. JMAP Capability Registration for "quota"

IANA will register the "quota" JMAP Capability as follows:

Capability Name: "urn:ietf:params:jmap:quota"

Specification document: this document

Intended use: common

Change Controller: IETF

Security and privacy considerations: this document, section 4.

6. 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>.

   [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>.

   [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

   Rene Cordier (editor)
   Linagora
   100 Terrasse Boieldieu - Tour Franklin
   Paris - La Defense CEDEX 92042
   France

   Email: rcordier@linagora.com
   URI: https://www.linagora.com

   Michael Bailly (editor)
   Linagora
   100 Terrasse Boieldieu - Tour Franklin
   Paris - La Defense CEDEX 92042
   France

   Email: mbailly@linagora.com
   URI: https://www.linagora.com