Workgroup: JMAP
Internet-Draft: draft-ietf-jmap-sharing-02
Published: 6 October 2022
Intended Status: Standards Track
Expires: 9 April 2023
Authors: N.M. Jenkins, Ed.
Fastmail

JMAP Sharing

Abstract

This document specifies a data model for sharing data between users 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 9 April 2023.

Table of Contents

1. Introduction

JMAP ([RFC8620] – (U+2013) JSON Meta Application Protocol) is a generic protocol for synchronizing data, such as mail, calendars or contacts, between a client and a server. It is optimized for mobile and web environments, and aims to provide a consistent interface to different data types.

This specification defines a data model to represent entities in a collaborative environment and a framework for sharing data between them that can be used to provide a consistent sharing model for

different data types. It does not define *what* may be shared, or the granularity of permissions, as this will depend on the data in question.

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.

1.2. Terminology

The same terminology is used in this document as in the core JMAP specification, see [RFC8620], Section 1.6.

The terms Principal, and ShareNotification (with these specific capitalizations) are used to refer to the data types defined in this document and instances of those data types.

1.3. Data Model Overview

A Principal (see Section XXX) represents an individual, team, or resource (e.g., a room or projector). The object contains information about the entity being represented, such as a name, description, and time zone. It may also hold domain-specific information. A Principal may be associated with zero or more Accounts (see [RFC8620], Section 1.6.2) containing data belonging to the principal. Managing the set of principals within a system is out of scope for this specification, as it is highly domain specific. It is likely to map directly from a directory service or other user management system.

Data types may allow users to share data with others by assigning permissions to principals. When a user's permissions are changed, a ShareNotification object is created for them so a client can inform the user of the changes.

1.4. Subscriptions

Permissions determine whether a user *may* access data, but not whether they *want* to. Some shared data is of equal importance as the user's own, while other data is just there should the user wish to explicitly go find it. Clients will often want to differentiate the

two; for example, a company may share mailing list archives for all departments with all employees, but a user may only generally be interested in the few they belong to. They would have *permission* to access many mailboxes, but can *subscribe* to just the ones they care about. The client would provide separate interfaces for reading mail in subscribed mailboxes and browsing all mailboxes they have permission to access in order to manage their subscriptions.

The JMAP Session object (see [RFC8620], Section 2) typically includes an object in the accounts property for every account that the user has access to. Collaborative systems may share data between a very large number of Principals, most of which the user does not care about day-to-day. The Session object MUST only include Accounts where either the user is subscribed to at least one record (see [RFC8620], Section 1.6.3) in the account, or the account belongs to the user. StateChange events for changes to data SHOULD only be sent for data the user has subscribed to and MUST NOT be sent for any account where the user is not subscribed to any records in the account, except where that account belongs to the user.

The server MAY reject the user's attempt to subscribe to some resources even if they have permission to access them, e.g., a calendar representing a location.

A user may query the set of Principals they have access to with "Principal/query" (see Section XXX). The Principal object may then provide Account objects if the user has permission to access data for that principal, even if they are not yet subscribed.

1.5. Addition to the Capabilities Object

The capabilities object is returned as part of the JMAP Session object; see [RFC8620], Section 2. This document defines two additional capability URIs.

1.5.1. urn:ietf:params:jmap:principals

Represents support for the Principal and ShareNotification data types and associated API methods.

The value of this property in the JMAP Session capabilities property is an empty object.

The value of this property in an account’ (U+2019)s accountCapabilities property is an object that MUST contain the

following information on server capabilities and permissions for that account:

- **currentUserPrincipalId**: Id|null The id of the principal in this account that corresponds to the user fetching this object, if any.

1.5.2. urn:ietf:params:jmap:principals:owner

This URI is solely used as a key in an account’ (U+2019)s accountCapabilities property; it does not appear in the JMAP Session capabilities. Support is implied by the urn:ietf:params:jmap:principals session capability.

If present, the account (and data therein) is owned by a principal. Some accounts may not be owned by a principal (e.g., the account that contains the data for the principals themselves), in which case this property is omitted.

The value of this property is an object with the following properties:

- **accountIdForPrincipal**: Id The id of an account with the urn:ietf:params:jmap:principals capability that contains the corresponding Principal object.
- **principalId**: Id The id of the Principal that owns this account.

2. Principals

A Principal represents an individual, group, location (e.g. a room), resource (e.g. a projector) or other entity in a collaborative environment. Sharing in JMAP is generally configured by assigning rights to certain data within an account to other principals, for example a user may assign permission to read their calendar to a principal representing another user, or their team.

In a shared environment such as a workplace, a user may have access to a large number of principals.

In most systems the user will have access to a single Account containing Principal objects, but they may have access to multiple if, for example, aggregating data from different places.

A **Principal** object has the following properties:

- **id**: Id The id of the principal.
- **type**: String This MUST be one of the following values:

  - individual: This represents a single person.
  - group: This represents a group of people.

-resource: This represents some resource, e.g. a projector.
-location: This represents a location.
-other: This represents some other undefined principal.

* **name**: String The name of the principal, e.g. "Jane Doe", or "Room 4B".
* **description**: String|null A longer description of the principal, for example details about the facilities of a resource, or null if no description available.
* **email**: String|null An email address for the principal, or null if no email is available.
* **timeZone**: String|null The time zone for this principal, if known. If not null, the value MUST be a time zone id from the IANA Time Zone Database TZDB.
* **capabilities**: String[Object] A map of JMAP capability URIs to domain specific information about the principal in relation to that capability, as defined in the document that registered the capability.
* **accounts**: Id[Account]|null A map of account id to Account object for each JMAP Account containing data for this principal that the user has access to, or null if none.

2.1. Principal/get

This is a standard "/get" method as described in [RFC8620], Section 5.1.

2.2. Principal/changes

This is a standard "/changes" method as described in [RFC8620], Section 5.2.

2.3. Principal/set

This is a standard "/set" method as described in [RFC8620], Section 5.3.

Users SHOULD be allowed to update the "name", "description" and "timeZone" properties of the Principal with the same id as the "currentUserPrincipalId" in the Account capabilities.

However, the server may, and probably will, reject any change with a forbidden SetError. Managing principals is likely tied to a directory service or some other vendor-specific solution, and may occur out-of-band, or via an additional capability defined elsewhere.

2.4. Principal/query

This is a standard "/query" method as described in [RFC8620], Section 5.5

2.4.1. Filtering

A **FilterCondition** object has the following properties:

- ***accountIds**: String[] A list of account ids. The Principal matches if any of the ids in this list are keys in the Principal's "accounts" property (i.e., if any of the account ids belong to the principal).
- ***email**: String Looks for the text in the email property.
- ***name**: String Looks for the text in the name property.
- ***text** String Looks for the text in the name, email, and description properties.
- ***type**: String The type must be exactly as given to match the condition.
- ***timeZone**: String The timeZone must be exactly as given to match the condition.

All conditions in the FilterCondition object must match for the Principal to match.

2.5. Principal/queryChanges

This is a standard "/queryChanges" method as described in [RFC8620], Section 5.6.

3. Share Notifications

The ShareNotification data type records when the user's permissions to access a shared object changes. ShareNotification are only created by the server; users cannot create them explicitly. Notifications are stored in the same Account as the Principals.

Clients SHOULD present the list of notifications to the user and allow them to dismiss them. To dismiss a notification you use a standard "/set" call to destroy it.

The server SHOULD create a ShareNotification whenever the user's permissions change on an object. It SHOULD NOT create a notification for permission changes to a group principal, even if the user is in the group.

3.1. Auto-deletion of Notifications

The server MAY limit the maximum number of notifications it will store for a user. When the limit is reached, any new notification will cause the previously oldest notification to be automatically deleted.

The server MAY coalesce notifications if appropriate, or remove notifications that it deems are no longer relevant or after a

certain period of time. The server SHOULD automatically destroy a notification about an object if the user subscribes to that object.

3.2. Object Properties

The **ShareNotification** object has the following properties:

- ***id**: String The id of the ShareNotification.
- ***created**: UTCDate The time this notification was created.
- ***changedBy**: Person Who made the change.
  - **name**: String The name of the person who made the change.
  - **email**: String|null The email of the person who made the change, or null if no email is available.
  - **principalId**: String|null The id of the Principal corresponding to the person who made the change, or null if no associated principal.
- ***objectType**: String The name of the data type for the object whose permissions have changed, e.g. "Calendar" or "Mailbox".
- ***objectAccountId**: String The id of the account where this object exists.
- ***objectId**: String The id of the object that this notification is about.
- ***name**: String The name of the object at the time the notification was made.
- ***oldRights**: String[Boolean]|null The "myRights" property of the object for the user before the change.
- ***newRights**: String[Boolean]|null The "myRights" property of the object for the user after the change.

3.3. ShareNotification/get

This is a standard "/get" method as described in [RFC8620], Section 5.1.

3.4. ShareNotification/changes

This is a standard "/changes" method as described in [RFC8620], Section 5.2.

3.5. ShareNotification/set

This is a standard "/set" method as described in [RFC8620], Section 5.3.

Only destroy is supported; any attempt to create/update MUST be rejected with a forbidden SetError.

3.6. ShareNotification/query

This is a standard "/query" method as described in [RFC8620], Section 5.5.

3.6.1. Filtering

A **FilterCondition** object has the following properties:

- ***after**: UTCDate|null The creation date must be on or after this date to match the condition.
- ***before**: UTCDate|null The creation date must be before this date to match the condition.
- ***objectType**: String The objectType value must be identical to the given value to match the condition.
- ***objectAccountId**: String The objectAccountId value must be identical to the given value to match the condition.

3.6.2. Sorting

The "created" property MUST be supported for sorting.

3.7. ShareNotification/queryChanges

This is a standard "/queryChanges" method as described in [RFC8620], Section 5.6.

4. Framework for shared data

Shareable data types SHOULD define the following three properties:

- ***isSubscribed**: Boolean Has the user indicated they wish to see this data? The initial value for this when data is shared by another user is implementation dependent, although data types may give advice on appropriate defaults.
- ***myRights**: String[Boolean] The set of permissions the user currently has. Appropriate permissions are domain specific and must be defined per data type.
- ***shareWith**: Id[String[Boolean]]|null A map of principal id to rights to give that principal, or null if not shared with anyone. The account id for the principal id can be found in the capabilities of the Account this object is in (see Section XXX). Users with appropriate permission may set this property to modify who the data is shared with. The principal that owns the account this data is in MUST NOT be in the set of sharees; their rights are implicit.

5. Security Considerations

All security considerations of JMAP [RFC8620] apply to this specification. Additional considerations are detailed below.

5.1. Spoofing

Allowing users to edit their own Principal's name (and, to a lesser extent, description) could allow a user to change their name to that of another user in the system, potentially tricking others into sharing private data with them. Servers may choose to forbid this, and SHOULD keep logs of such changes to provide an audit trail.

5.2. Unnoticed sharing

Sharing data with another user allows someone to turn a transitory account compromise (e.g. brief access to an unlocked, logged in client) into a persistant compromise (by setting up sharing with a user controlled by the attacker). This can be mitigated by requiring further authorisation for configuring sharing, or sending notifications to the sharer via another channel whenever a new sharee is added.

5.3. Unauthorised principals

The set of principals within a shared environment SHOULD be strictly controlled. If adding a new principal is open to the public, risks include: * An increased risk of a user accidentally sharing data with an unintended person. * An attacker may share unwanted or offensive information with the user. * An attacker may share items with spam content in the names in order to generate ShareNotification objects, which are likely to be prominently displayed to the sharee.

6. IANA Considerations

6.1. JMAP Capability Registration for "principals"

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

Capability Name: urn:ietf:params:jmap:principals

Specification document: this document

Intended use: common

Change Controller: IETF

Security and privacy considerations: this document, Section XXX

6.2. JMAP Capability Registration for "principals:owner"

IANA will register the "principals:owner" JMAP Capability as follows:

Capability Name: urn:ietf:params:jmap:principals:owner

Specification document: this document

Intended use: common

Change Controller: IETF

Security and privacy considerations: this document, Section XXX

7. 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>.

Author's Address

Neil Jenkins (editor)
Fastmail
PO Box 234, Collins St West
Melbourne VIC 8007
Australia

Email: neilj@fastmailteam.com
URI: https://www.fastmail.com