# JMAP REST Mapping

## Abstract

This document specifies a REST Mapping for JMAP endpoints to impose fewer requirements on applications compared to conventional JMAP endpoints.

## Status of This Memo

## Copyright Notice

Table of Contents

## 1. Introduction

Structured data exchange over JMAP [RFC8620] usually involves processing JMAP Request JSON payloads. This might impose unnecessary requirements for certain use cases of JMAP. Likely scenarios in which this is beneficiary are situations in which portability needs to be provided due to regulatory requirements or when migrating user data away from legacy platforms.

For rapid development of a JMAP API without support for batching, the essential properties of the Request object can instead be implemented as a URI.

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

The definitions of JSON keys and datatypes in the document follow the conventions described in the core JMAP specification [RFC8620].

### 1.2. 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.

### 1.2.1. urn:ietf:params:jmap:rest

The capability urn:ietf:params:jmap:rest being present in the "capabilities" property represents support for the simplified JMAP REST API.

The value of this property in the JMAP Session capabilities property
and the account's accountCapabilities property is an empty object.

## 1.3.  Addition to the Session Resource

The JMAP Session Resource will be extended by the following
property:

  *__apiUrlSimple__: String The URL to use for JMAP API requests. THE
   URL MUST contain the variables using and methodCall as defined in
   [RFC8620] Section 3.3. One MAY use additional variables specific
   to a JMAP method like ids or accountId. Implementations MAY
   implement the required variables as query parameters to avoid
   implementing routing logic.

### 1.3.1.  Capability Example

```
{
  ...
  "capabilities": {
    ...,
    "urn:ietf:params:jmap:rest": {}
  },
  "apiUrlRest": "https://jmap.me/api
    /?accountId=<account-id>&methodCall=<methodCall>&ids=<ids>"
}
```

## 2.  Security considerations

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

## 3.  IANA considerations

## 3.1.  JMAP Capability registration for "rest"

IANA is requested to register the "rest" JMAP Capability as follows:

Capability Name: urn:ietf:params:jmap:rest

Specification document: this document

Intended use: common

Change Controller: IETF

Security and privacy considerations: this document, Section 2.

## 4. Acknowledgements

Bron Gondwana, Neil Jenkins, Alexey Melnikov, Ken Murchison, Robert Stepanek and the JMAP working group at the IETF.

## 5. 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

Joris Baum (editor)
audriga
Alter Schlachthof 57
76137 Karlsruhe
Germany

Email: joris@audriga.com
URI: https://www.audriga.com

Hans-Joerg (editor)
audriga
Alter Schlachthof 57
76137 Karlsruhe
Germany

Email: hans-joerg@audriga.com
URI: https://www.audriga.com