Internet Engineering Task Force (IETF)
Request for Comments: 9661
Category: Standards Track
ISSN: 2070-1721

K. Murchison
Fastmail

September 2024

The JSON Meta Application Protocol (JMAP) for Sieve Scripts

Abstract

This document specifies a data model for managing Sieve scripts on a server using the JSON Meta Application Protocol (JMAP). Clients can use this protocol to efficiently search, access, organize, and validate Sieve scripts.

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/rfc9661.

Copyright Notice

Copyright (c) 2024 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. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

1. Introduction
   1.1. Notational Conventions
   1.2. Addition to the Capabilities Object
      1.2.1. urn:ietf:params:jmap:sieve
      1.2.2. Example
2. Sieve Scripts
   2.1. Sieve Script Properties
   2.2. Sieve Script Content
   2.3. SieveScript/get
      2.3.1. Examples
   2.4. SieveScript/set
      2.4.1. Examples
   2.5. SieveScript/query
   2.6. SieveScript/validate
3. Quotas
4. Compatibility with JMAP Vacation Response

5. Security Considerations
6. IANA Considerations
6.1. JMAP Capability Registration for "sieve"
6.2. JMAP Data Type Registration for "SieveScript"
6.3. JMAP Error Codes Registry
6.3.1. invalidSieve
6.3.2. sieveIsActive
7. References
7.1. Normative References
7.2. Informative References
Acknowledgments
Author's Address

1. Introduction

The JSON Meta Application Protocol (JMAP) [RFC8620] 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 it aims to provide a consistent interface to different data types.

This specification defines a data model for managing Sieve scripts [RFC5228] on a server using JMAP. The data model is designed to allow a server to provide consistent access to the same scripts via ManageSieve [RFC5804] as well as JMAP.

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]. This document also uses data types and terminology established in Sections 1.2 through 1.6 of [RFC8620].

The term "SieveScript" (with this specific capitalization) is used to refer to the data type defined in Section 2 and instances of this data type used throughout this document. Servers MUST support all properties specified for the data type defined in this document.

For brevity, JMAP API examples (see Section 3 of [RFC8620]) only show the "methodCalls" property of the "Request" object and the "methodResponses" property of the "Response" object. All other examples are shown using the HTTP/1.1 protocol [RFC9112].

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:sieve

The urn:ietf:params:jmap:sieve URI represents support for the SieveScript data type and associated API methods. The value of this property in the JMAP Session "capabilities" property is an object that MUST contain the following information on server capabilities:

implementation: String

The name and version of the Sieve implementation.

The value of this property in an account's "accountCapabilities" property is an object that MUST contain the following information on per-account server capabilities:

maxSizeScriptName: UnsignedInt

The maximum length, in octets, allowed for the name of a SieveScript. For compatibility with ManageSieve, this MUST be at least 512 (up to 128 Unicode characters).

maxSizeScript: UnsignedInt|null

The maximum size (in octets) of a Sieve script the server is willing to store for the user, or null for no limit.

maxNumberScripts: UnsignedInt|null

The maximum number of Sieve scripts the server is willing to store for the user, or null for no limit.

maxNumberRedirects: UnsignedInt|null

The maximum number of Sieve "redirect" actions a script can perform during a single evaluation, or null for no limit. Note that this is different from the total number of "redirect" actions a script can contain.

sieveExtensions: String[]

A list of case-sensitive Sieve capability strings (as listed in the Sieve "require" action; see [RFC5228], Section 3.2) indicating the extensions supported by the Sieve engine.

notificationMethods: String[]|null

A list of URI scheme parts [RFC3986] for notification methods supported by the Sieve "enotify" extension [RFC5435], or null if the extension is not supported by the Sieve engine.

externalLists: String[]|null

A list of URI scheme parts [RFC3986] for externally stored list types supported by the Sieve "extlists" extension [RFC6134], or null if the extension is not supported by the Sieve engine.

1.2.2. Example

This example JMAP Session object shows a user that has access to their own Sieve scripts with support for a few Sieve extensions:

```
{
  "capabilities": {
    "urn:ietf:params:jmap:core": {
      ...
    },
    "urn:ietf:params:jmap:mail": {},
    "urn:ietf:params:jmap:quota": {},
    "urn:ietf:params:jmap:blob": {},
    "urn:ietf:params:jmap:sieve": {
      "implementation": "ACME Email Filtering"
    },
    "urn:ietf:params:jmap:vacationresponse": {},
    ...
  },
  "accounts": {
    "ken": {
```

```
      "name": "ken@example.com",
      "isPersonal": true,
      "isReadOnly": false,
      "accountCapabilities": {
        "urn:ietf:params:jmap:core": {},
        "urn:ietf:params:jmap:quota": {},
        "urn:ietf:params:jmap:mail": {
          ...
        },
        "urn:ietf:params:jmap:blob": {
          "supportedTypeNames": [
            "Email"
            "SieveScript",
            ...
          ],
          ...
        },
        "urn:ietf:params:jmap:sieve": {
          "maxSizeScriptName": 512,
          "maxSizeScript": 65536,
          "maxNumberScripts": 5,
          "maxNumberRedirects": null,
          "sieveExtensions": [
            "fileinto",
            "imap4flags",
            "enotify",
            ...
          ],
          "notificationMethods": [
            "mailto"
          ],
          "externalLists": null,
        },
        "urn:ietf:params:jmap:vacationresponse": {},
        ...
      },
      ...
    }
  },
  "primaryAccounts": {
    "urn:ietf:params:jmap:mail": "ken",
    "urn:ietf:params:jmap:sieve": "ken",
    "urn:ietf:params:jmap:vacationresponse": "ken",
    ...
  },
  "username": "ken@example.com",
  "apiUrl": "/jmap/",
  "downloadUrl":
    "/jmap/download/{accountId}/{blobId}/{name}?accept={type}",
  "uploadUrl": "/jmap/upload/{accountId}/",
  ...
}
```

2. Sieve Scripts

A "SieveScript" object represents a single Sieve script [RFC5228] for filtering email messages at the time of final delivery.

2.1. Sieve Script Properties

A "SieveScript" object has the following properties:

id: Id (immutable; server-set)

The id of the script.

name: String|null (optional; default is server dependent)

User-visible name for the SieveScript. If non-null, this MUST be a Net-Unicode string [RFC5198] of at least 1 character in length, subject to the maximum size given in the "capability" object.

For compatibility with ManageSieve, servers MUST reject names that contain any of the following Unicode characters: U+0000-U+001F, U+007F-U+009F, U+2028, or U+2029.

Servers MAY reject names that violate server policy (e.g., names containing a slash (/)).

The name MUST be unique among all SieveScripts within an account.

blobId: Id

The id of the blob containing the raw octets of the script.

isActive: Boolean (server-set; default: false)

Indicator that the SieveScript is actively filtering incoming messages.

A user may have at most one active script. The SieveScript/set method (Section 2.4) is used for changing the active script or disabling Sieve processing.

2.2. Sieve Script Content

A script MUST be UTF-8 content [RFC3629] of at least 1 character in length, subject to the syntax of Sieve [RFC5228]. A script MUST NOT contain any "require" statement(s) mentioning Sieve capability strings not present in the "capability" object (Section 1.2.1). Note that if the Sieve "ihave" capability string [RFC5463] is present in the "capability" object, the script MAY mention unrecognized/unsupported extensions in the "ihave" test.

Script content is treated as a binary blob and uploaded/downloaded via the mechanisms provided in Sections 6.1 and 6.2 of [RFC8620], respectively, and/or via the JMAP Blob management methods provided in Sections 4.1 and 4.2 of [RFC9404], respectively.

Downloading script content via the JMAP downloadUrl or the Blob/get method provides functionality equivalent to that of the GETSCRIPT command defined in [RFC5804].

2.3. SieveScript/get

This is a standard "/get" method as described in [RFC8620], Section 5.1. The "ids" argument may be null to fetch all scripts at once.

This method provides functionality equivalent to that of the LISTSCRIPTS command defined in [RFC5804].

2.3.1. Examples

List all scripts:

```
[
  ["SieveScript/get", {
    "accountId": "ken"
  }, "0"]
]
```

```
[
  [
    "SieveScript/get",
    {
      "state": "1634915373.240633104-120",
      "list": [
        {
          "id": "2d647053-dded-418d-917a-63eda3ac8f7b",
          "name": "test1",
          "isActive": true,
          "blobId": "S7"
        }
      ],
      "notFound": [],
      "accountId": "ken"
    },
    "0"
  ]
]
```

Download the script content via the JMAP downloadUrl as advertised in the example in Section 1.2.2:

```
GET /jmap/download/ken/S7/test1.siv?accept=application/sieve HTTP/1.1
Host: jmap.example.com
Authorization: Basic a2VuOnBhc3N3b3Jk

HTTP/1.1 200 OK
Date: Fri, 22 Oct 2021 15:27:38 GMT
Content-Type: application/sieve; charset=utf-8
Content-Disposition: attachment; filename="test1.siv"
Content-Length: 49

require ["fileinto"];
fileinto "INBOX.target";
```

Fetch script properties and content in a single JMAP API request using the JMAP Blob management extension [RFC9404]:

```
[
  ["SieveScript/get", {
    "accountId": "ken",
    "ids": [ "2d647053-dded-418d-917a-63eda3ac8f7b" ]
  }, "0"],
  ["Blob/get", {
    "accountId": "ken",
    "#ids": {
      "resultOf": "0",
      "name": "SieveScript/get",
      "path": "/list/*/blobId"
    }
  }, "1"]
]

[
  [
    "SieveScript/get",
    {
      "state": "1634915373.240633104-120",
      "list": [
        {
          "id": "2d647053-dded-418d-917a-63eda3ac8f7b",
          "name": "test1",
          "isActive": true,
          "blobId": "S7"
        }
```

```
      ],
      "notFound": [],
      "accountId": "ken"
    },
    "0"
  ],
  [
    "Blob/get",
    {
      "list": [
        {
          "id": "S7",
          "data:asText":
     "require [\"fileinto\"];\\r\\nfileinto \"INBOX.target\";\\r\\n",
          "size": 49
        }
      ],
      "notFound": [],
      "accountId": "ken"
    },
    "1"
  ]
]
```

2.4. SieveScript/set

This is a standard "/set" method as described in [RFC8620], Section 5.3, but with the following additional optional request arguments:

onSuccessActivateScript: Id

The id of the SieveScript to activate if and only if all of the creations, modifications, and destructions (if any) succeed. (For references to SieveScript creations, this is equivalent to a creation-reference, so the id will be the creation id prefixed with a "#".) The currently active SieveScript (if any) will be deactivated before activating the specified SieveScript.

If omitted, or if the id is either invalid or nonexistent, it MUST be ignored, and the currently active SieveScript (if any) will remain as such.

The id of any activated SieveScript MUST be reported in either the "created" or "updated" argument in the response as appropriate, including a value of "true" for the "isActive" property. The id of any deactivated SieveScript MUST be reported in the "updated" argument in the response, including a value of "false" for the "isActive" property.

onSuccessDeactivateScript: Boolean

If "true", the currently active SieveScript (if any) will be deactivated if and only if all of the creations, modifications, and destructions (if any) succeed. If "false" or omitted, the currently active SieveScript (if any) will remain as such.

The id of any deactivated SieveScript MUST be reported in the "updated" argument in the response, including a value of "false" for the "isActive" property.

If both the "onSuccessActivateScript" and "onSuccessDeactivateScript" arguments are present in the request, then "onSuccessDeactivateScript" MUST be processed first. If neither argument is present in the request, the currently active SieveScript (if any) will remain as such.

This method provides functionality equivalent to that of the PUTSCRIPT, DELETESCRIPT, RENAMESCRIPT, and SETACTIVE commands defined in [RFC5804].

Script content must first be uploaded as per Section 2.2 prior to referencing it in a SieveScript/set call.

If the SieveScript cannot be created or updated because it would result in two SieveScripts with the same name, the server MUST reject the request with an "alreadyExists" SetError. An "existingId" property of type "Id" MUST be included on the SetError object with the id of the existing SieveScript.

If the SieveScript cannot be created or updated because its size exceeds the "maxSizeScript" limit, the server MUST reject the request with a "tooLarge" SetError.

If the SieveScript cannot be created because it would exceed the "maxNumberScripts" limit or would exceed a server-imposed storage limit, the server MUST reject the request with an "overQuota" SetError.

The active SieveScript MUST NOT be destroyed unless it is first deactivated in a separate SieveScript/set method call.

The following extra SetError types are defined:

For "create" and "update":

invalidSieve: The SieveScript content violates the Sieve grammar [RFC5228], and/or one or more extensions mentioned in the script's "require" statement(s) are not supported by the Sieve interpreter. The "description" property on the SetError object SHOULD contain a specific error message giving at least the line number of the first error.

For "destroy":

sieveIsActive: The SieveScript is active.

2.4.1. Examples

Upload a script requiring the Imap4Flags Extension [RFC5232] using the JMAP uploadUrl as advertised in the example in Section 1.2.2:

```
POST /jmap/upload/ken/ HTTP/1.1
Host: jmap.example.com
Authorization: Basic a2VuOnBhc3N3b3Jk
Content-Type: application/sieve
Content-Length: 98

require "imapflags";

if address :is ["To", "Cc"] "jmap@ietf.org" {
  setflag "\\Flagged";
}


HTTP/1.1 201 Created
Date: Thu, 10 Dec 2020 17:14:31 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 171

{
  "accountId": "ken",
```

```
  "blobId": "Gabcc83e44a6e19991c4568d0b94e1767c83dd123",
  "type": "application/sieve"
  "size": 98
}
```

Create and activate a script using the uploaded blob. Note that the response shows that an existing active script has been deactivated in lieu of the newly created script being activated.

```
[
  ["SieveScript/set", {
    "accountId": "ken",
    "create": {
      "A": {
        "name": null,
        "blobId": "Gabcc83e44a6e19991c4568d0b94e1767c83dd123"
      }
    },
    "onSuccessActivateScript": "#A"
  }, "0"]
]
```

```
[
  [
    "SieveScript/set",
    {
      "oldState": "1603741717.50737918-4096",
      "newState": "1603741751.227268529-4096",
      "created": {
        "A": {
          "id": "dd1b164f-8cdc-448c-9f54",
          "name": "ken-20201210T171432-0",
          "blobId": "Sdd1b164f-8cdc-448c-9f54",
          "isActive": true
        }
      },
      "updated": {
        "8abd6f4a-bcb4d-87650-3fcd": {
          "isActive": false
        }
      },
      "destroyed": null,
      "notCreated": null,
      "notUpdated": null,
      "notDestroyed": null,
      "accountId": "ken"
    },
    "0"
  ]
]
```

Update the script content using the JMAP Blob management extension [RFC9404]:

```
{
[
  ["Blob/upload", {
    "accountId": "ken",
    "create": {
      "B": {
        "data": [ {
          "data:asText":
            "redirect \"ken@example.com\"\r\n;"
         } ],
        "type": "application/sieve"
      }
```

```
    }
  }, "1"],
  ["SieveScript/set", {
    "accountId": "ken",
    "update": { "dd1b164f-8cdc-448c-9f54": {
      "blobId": "#B"
      }
    }
  }, "2"]
]

[
  [
    "Blob/upload",
    {
      "oldState": null,
      "newState": "1603741700.309607123-0128",
      "created": {
        "B": {
          "id": "G969c83e44a6e10871c4568d0b94e1767c83ddeae",
          "blobId": "G969c83e44a6e10871c4568d0b94e1767c83ddeae",
          "type": "application/sieve",
          "size": 29
        }
      },
      "notCreated": null,
      "accountId": "ken"
    },
    "1"
  ],
  [
    "SieveScript/set",
    {
      "oldState": "1603741751.227268529-4096",
      "newState": "1603742603.309607868-4096",
      "created": null,
      "updated": {
        "dd1b164f-8cdc-448c-9f54": null
      },
      "destroyed": null,
      "notCreated": null,
      "notUpdated": null,
      "notDestroyed": null,
      "accountId": "ken"
    },
    "2"
  ]
]

Update the script name, and deactivate it:

[
  ["SieveScript/set", {
    "accountId": "ken",
    "update": { "dd1b164f-8cdc-448c-9f54": {
      "name": "myscript"
      }
    },
    "onSuccessDeactivateScript": true
  }, "3"]
]

[
  [
    "SieveScript/set",
    {
```

```
      "oldState": "1603742603.309607868-4096",
      "newState": "1603742967.852315428-4096",
      "created": null,
      "updated": {
        "dd1b164f-8cdc-448c-9f54": {
          "isActive": false
        }
      },
      "destroyed": null,
      "notCreated": null,
      "notUpdated": null,
      "notDestroyed": null,
      "accountId": "ken"
    },
    "3"
  ]
]
```

Reactivate the script:

```
[
  ["SieveScript/set", {
    "accountId": "ken",
    "onSuccessActivateScript": "dd1b164f-8cdc-448c-9f54"
  }, "4"]
]
```

```
[
  [
    "SieveScript/set",
    {
      "oldState": "1603742967.852315428-4096",
      "newState": "1603744460.316617118-4096",
      "created": null,
      "updated": {
        "dd1b164f-8cdc-448c-9f54": {
          "isActive": true
        }
      },
      "destroyed": null,
      "notCreated": null,
      "notUpdated": null,
      "notDestroyed": null,
      "accountId": "ken"
    },
    "4"
  ]
]
```

Deactivate and destroy the active script:

```
[
  ["SieveScript/set", {
    "accountId": "ken",
    "onSuccessDeactivateScript": true
  }, "5"],
  ["SieveScript/set", {
    "accountId": "ken",
    "destroy": [ "dd1b164f-8cdc-448c-9f54" ]
  }, "6"]
]
```

```
[
  [
    "SieveScript/set",
    {
```

```
        "oldState": "1603744460.316617118-4096",
        "newState": "1603744637.575375572-4096",
        "created": null,
        "updated": {
          "dd1b164f-8cdc-448c-9f54": {
            "isActive": false
          }
        },
        "destroyed": null,
        "notCreated": null,
        "notUpdated": null,
        "notDestroyed": null,
        "accountId": "ken"
      },
      "5"
    ],
    [
      "SieveScript/set",
      {
        "oldState": "1603744637.575375572-4096",
        "newState": "1603744637.854390875-4096",
        "created": null,
        "updated": null,
        "destroyed": [
          "dd1b164f-8cdc-448c-9f54"
        ],
        "notCreated": null,
        "notUpdated": null,
        "notDestroyed": null,
        "accountId": "ken"
      },
      "6"
    ]
  ]
```

2.5. SieveScript/query

This is a standard "/query" method as described in [RFC8620], Section 5.5. A "FilterCondition" object has the following properties, either of which may be omitted:

name: String

The SieveScript "name" property contains the given string.

isActive: Boolean

The "isActive" property of the SieveScript must be identical to the value given to match the condition.

The following SieveScript properties MUST be supported for sorting:

* *name*

* *isActive*

2.6. SieveScript/validate

This method is used by the client to verify Sieve script validity without storing the script on the server.

The method takes the following arguments:

accountId: Id

The id of the account to use.

blobId: Id

The id of the blob containing the raw octets of the script to validate, subject to the same requirements in Section 2.2.

The response has the following arguments:

accountId: Id

The id of the account used for this call.

error: SetError|null

An "invalidSieve" SetError object if the script content is invalid (see Section 2.4), or null if the script content is valid.

This method provides functionality equivalent to that of the CHECKSCRIPT command defined in [RFC5804].

Script content must first be uploaded as per Section 2.2 prior to referencing it in a SieveScript/validate call.

3. Quotas

Servers SHOULD impose quotas on Sieve scripts to prevent malicious users from exceeding available storage. Administration of such quotas is outside of the scope of this specification; however, [RFC9425] defines a data model for users to obtain quota details over JMAP.

The mechanism for handling SieveScript requests that would place a user over a quota setting is discussed in Section 2.4.

4. Compatibility with JMAP Vacation Response

Section 8 of [RFC8621] defines a "VacationResponse" object to represent an autoresponder to incoming email messages. Servers that implement the VacationResponse as a Sieve script that resides among other user scripts are subject to the following requirements:

* MUST allow the VacationResponse Sieve script to be fetched by the SieveScript/get method (Section 2.3).

* MUST allow the VacationResponse Sieve script to be activated or deactivated via the "onSuccessActivateScript" argument to the SieveScript/set method (Section 2.4).

* MUST NOT allow the VacationResponse Sieve script to be destroyed or have its content updated by the SieveScript/set method (Section 2.4). Any such request MUST be rejected with a "forbidden" SetError. A "description" property MAY be present with an explanation that the script can only be modified by a VacationResponse/set method.

5. Security Considerations

All security considerations discussed in JMAP [RFC8620] and Sieve [RFC5228] apply to this specification.

Additionally, implementations MUST treat Sieve script content as untrusted data. As such, script parsers MUST fail gracefully in the face of syntactically invalid or malicious content and MUST be prepared to deal with resource exhaustion (e.g., allocation of enormous strings, lists, or command blocks).

6. IANA Considerations

6.1. JMAP Capability Registration for "sieve"

IANA has registered "sieve" in the "JMAP Capabilities" registry as follows:

Capability Name: urn:ietf:params:jmap:sieve

Reference: RFC 9661

Intended Use: common

Change Controller: IETF

Security and Privacy Considerations: RFC 9661, Section 5

6.2. JMAP Data Type Registration for "SieveScript"

IANA has registered "SieveScript" in the "JMAP Data Types" registry as follows:

Type Name: SieveScript

Can Reference Blobs: Yes

Can Use for State Change: Yes

Capability: urn:ietf:params:jmap:sieve

Reference: RFC 9661

6.3. JMAP Error Codes Registry

IANA has registered the following two new error codes in the "JMAP Error Codes" registry, as defined in [RFC8620].

6.3.1. invalidSieve

JMAP Error Code: invalidSieve

Intended Use: common

Change Controller: IETF

Reference: RFC 9661, Section 2.4

Description: The SieveScript violates the Sieve grammar [RFC5228], and/or one or more extensions mentioned in the script’s "require" statement(s) are not supported by the Sieve interpreter.

6.3.2. sieveIsActive

JMAP Error Code: sieveIsActive

Intended Use: common

Change Controller: IETF

Reference: RFC 9661, Section 2.4

Description: The client tried to destroy the active SieveScript.

7. References

7.1. 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>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, DOI 10.17487/RFC3986, January 2005, <https://www.rfc-editor.org/info/rfc3986>.

[RFC5198] Klensin, J. and M. Padlipsky, "Unicode Format for Network Interchange", RFC 5198, DOI 10.17487/RFC5198, March 2008, <https://www.rfc-editor.org/info/rfc5198>.

[RFC5228] Guenther, P., Ed. and T. Showalter, Ed., "Sieve: An Email Filtering Language", RFC 5228, DOI 10.17487/RFC5228, January 2008, <https://www.rfc-editor.org/info/rfc5228>.

[RFC5435] Melnikov, A., Ed., Leiba, B., Ed., Segmuller, W., and T. Martin, "Sieve Email Filtering: Extension for Notifications", RFC 5435, DOI 10.17487/RFC5435, January 2009, <https://www.rfc-editor.org/info/rfc5435>.

[RFC6134] Melnikov, A. and B. Leiba, "Sieve Extension: Externally Stored Lists", RFC 6134, DOI 10.17487/RFC6134, July 2011, <https://www.rfc-editor.org/info/rfc6134>.

[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>.

[RFC8621] Jenkins, N. and C. Newman, "The JSON Meta Application Protocol (JMAP) for Mail", RFC 8621, DOI 10.17487/RFC8621, August 2019, <https://www.rfc-editor.org/info/rfc8621>.

7.2. Informative References

[RFC5232] Melnikov, A., "Sieve Email Filtering: Imap4flags Extension", RFC 5232, DOI 10.17487/RFC5232, January 2008, <https://www.rfc-editor.org/info/rfc5232>.

[RFC5463] Freed, N., "Sieve Email Filtering: Ihave Extension", RFC 5463, DOI 10.17487/RFC5463, March 2009, <https://www.rfc-editor.org/info/rfc5463>.

[RFC5804] Melnikov, A., Ed. and T. Martin, "A Protocol for Remotely Managing Sieve Scripts", RFC 5804, DOI 10.17487/RFC5804, July 2010, <https://www.rfc-editor.org/info/rfc5804>.

[RFC9112] Fielding, R., Ed., Nottingham, M., Ed., and J. Reschke, Ed., "HTTP/1.1", STD 99, RFC 9112, DOI 10.17487/RFC9112, June 2022, <https://www.rfc-editor.org/info/rfc9112>.

[RFC9404] Gondwana, B., Ed., "JSON Meta Application Protocol (JMAP) Blob Management Extension", RFC 9404, DOI 10.17487/RFC9404, August 2023, <https://www.rfc-editor.org/info/rfc9404>.

[RFC9425] Cordier, R., Ed., "JSON Meta Application Protocol (JMAP) for Quotas", RFC 9425, DOI 10.17487/RFC9425, June 2023, <https://www.rfc-editor.org/info/rfc9425>.

Acknowledgments

The concepts in this document are based largely on those in [RFC5804]. The author would like to thank the authors of that document for providing both inspiration and some borrowed text for this document.

The author would also like to thank the following individuals for contributing their ideas and support for writing this specification: Joris Baum, Mauro De Gennaro, Bron Gondwana, Neil Jenkins, Alexey Melnikov, and Ricardo Signes.

Author’s Address

Kenneth Murchison
Fastmail US LLC
1429 Walnut Street, Suite 1201
Philadelphia, PA 19102
United States of America
Email: murch@fastmailteam.com