

**JMAP for Sieve Scripts**  
**draft-ietf-jmap-sieve-00**

Abstract

This document specifies a data model for managing Sieve scripts on a server using JMAP.

Open Issues

- o Do we need/want both "content" and "blobId" in the SieveScript object? It may be simpler to have just one way of specifying content and "blobId" is more versatile and doesn't require JSON-encoding of the content. Furthermore, use of the forthcoming(?) Blob/set method would avoid the extra roundtrip of having to upload the blob first.
- o Should changes made by onSuccessActivateScript be reported in the response to the client-initiated request, or via a second implicit SieveScript/set response?
- o The strawman for SieveScript/test only uses blobIds. Will this have to change once the issue regarding content/blobId in SieveScript is resolved?

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, 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](https://trustee.ietf.org/license-info) 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 Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction . . . . .](#) [3](#)
- [1.1. Notational Conventions . . . . .](#) [3](#)
- [1.2. Terminology . . . . .](#) [3](#)
- [1.3. Addition to the Capabilities Object . . . . .](#) [3](#)
- [1.3.1. urn:ietf:params:jmap:sieve . . . . .](#) [3](#)
- [2. Sieve Scripts . . . . .](#) [5](#)
- [2.1. SieveScript/get . . . . .](#) [6](#)
- [2.2. SieveScript/set . . . . .](#) [6](#)
- [2.3. SieveScript/query . . . . .](#) [8](#)
- [2.4. SieveScript/validate . . . . .](#) [9](#)
- [2.5. SieveScript/test . . . . .](#) [10](#)
- [3. Security Considerations . . . . .](#) [11](#)
- [4. IANA Considerations . . . . .](#) [11](#)
- [4.1. JMAP Capability Registration for "sieve" . . . . .](#) [11](#)
- [4.2. JMAP Error Codes Registry . . . . .](#) [12](#)
- [4.2.1. scriptNameExists . . . . .](#) [12](#)
- [4.2.2. tooManyScripts . . . . .](#) [12](#)
- [4.2.3. invalidScript . . . . .](#) [12](#)
- [4.2.4. scriptIsActive . . . . .](#) [13](#)
- [5. Acknowledgments . . . . .](#) [13](#)
- [6. References . . . . .](#) [13](#)
- [6.1. Normative References . . . . .](#) [13](#)
- [6.2. Informative References . . . . .](#) [14](#)
- [Appendix A. Change History \(To be removed by RFC Editor before publication\) . . . . .](#) [15](#)
- [Author's Address . . . . .](#) [15](#)



## **1. Introduction**

JMAP ([\[RFC8620\]](#) - 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 for managing Sieve [\[RFC5228\]](#) scripts 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, however the functionality offered over the two protocols may differ.

### **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 term SieveScript (with this specific capitalization) is used to refer to the data type defined in this document and instances of those data types.

### **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:sieve**

This represents support for the SieveScript data type 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's `accountCapabilities` property is an object that MUST contain the following information on server capabilities:

- o `*supportsTest*`: "Boolean"

If true, the server supports the SieveScript/test ([Section 2.5](#)) method.

- o `*maxNumberRedirects*`: "UnsignedInt|null"

The maximum number of Sieve "redirect" actions a script can perform during a single evaluation (see [RFC5804](#), [Section 1.7](#)), or "null" for no limit.

- o `*maxNumberScripts*`: "UnsignedInt|null"

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

- o `*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.

- o `*sieveExtensions*`: "String[]"

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

- o `*notificationMethods*`: "String[]|null"

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



- o `*externalLists*`: "String[]|null"

A list of URI schema 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.

## 2. Sieve Scripts

A `*SieveScript*` object represents a single Sieve [[RFC5228](#)] script. A user may have multiple SieveScripts on the server, yet only one script may be used for filtering of incoming messages. This is the active script. Users may have zero or one active script. The SieveScript/set ([Section 2.2](#)) method described below is used for changing the active script or disabling Sieve processing.

A `*SieveScript*` object has the following properties:

- o `*id*`: "Id" (immutable; server-set)

The id of the script.

- o `*name*`: "String"

The unique user-visible name for the script, subject to the requirements in [[RFC5804](#)], [Section 1.6](#).

- o `*content*`: "String"

The raw octets of the script [[RFC5228](#)].

Note that both Sieve and JSON require encoding of special characters which MUST be done in the following order:

1. Escape any double quote (") or backslash (\) characters appearing inside of quoted strings in the Sieve code per [Section 2.4.2 of \[RFC5228\]](#). E.g., A string containing the value \foo becomes "\\foo".
2. Escape any double quote ("), backslash (\), tab, carriage return, or line feed characters appearing in the resultant Sieve code per [Section 7 of \[RFC8259\]](#). E.g., The example string in step 1 becomes "\\\"\\foo\"".

- o `*blobId*`: "Id" (immutable)





The id of the blob containing the raw octets of the script [[RFC5228](#)].

- o `*isActive*`: "Boolean" (server-set; default: false)

Is this the user's active script?

Example (using the Imap4Flags [[RFC5232](#)] Extension):

```
{
  "id": "665c423a-6991-4733-8c7c-52b299572c66",
  "name": "example.siv",
  "content":
    "require [ \"imap4flags\" ];\r\nkeep :flags \"\\\\\\\\flagged\";",
  "isActive": false
}
```

### **[2.1.](#) SieveScript/get**

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

This method provides similar functionality to the `GETSCRIPT` and `LISTSCRIPTS` commands in [[RFC5804](#)].

### **[2.2.](#) SieveScript/set**

This is a standard `/set` method as described in [[RFC8620](#)], [Section 5.3](#) but with the following additional request arguments, either of which may be omitted:

- o `*replaceOnCreate*`: "Boolean" (default: false)

If false, any attempt to create a SieveScript having the same `"name"` property as an existing SieveScript MUST be rejected with a `"scriptNameExists"` SetError. If true, the existing SieveScript will be destroyed and the new SieveScript created as a single atomic action. The id of the replaced SieveScript MUST be reported in the `"destroyed"` argument in the response.

- o `*onSuccessActivateScript*`: "Id|null"



The id of the SieveScript to activate if the create/update/destroy succeeds. (For references to SieveScript creations, this is equivalent to a creation-reference, so the id will be the creation id prefixed with a "#".) If "null", the currently active script (if any) will be deactivated. If this argument is not present in the request, the currently active script (if any) will remain as such.

This method provides similar functionality to the PUTSCRIPT, DELETESCRIPT, RENAMESCRIPT, and SETACTIVE commands in [\[RFC5804\]](#).

When creating or updating a script, a client MUST include either a `_content_` or a `_blobId_` property. A request that includes neither or both properties MUST be rejected with an "invalidProperties" SetError. The server MUST check the script content for syntactic validity, which includes checking that all Sieve extensions mentioned in Sieve script "require" statement(s) are supported by the Sieve interpreter. (Note that if the Sieve interpreter supports the Sieve "ihave" extension [\[RFC5463\]](#), any unrecognized/unsupported extension mentioned in the "ihave" test MUST NOT cause the syntactic validation failure.) A script of zero length SHOULD be considered invalid. If the script content is invalid the request MUST be rejected with a "invalidScript" SetError.

Note that simply activating or deactivating a script without changing any script content is accomplished via a request containing an "onSuccessActivateScript" argument and "null" "create", "update", and "delete" arguments.

The following extra SetError types are defined:

For "create":

o `*scriptNameExists*`:

A SieveScript already exists with the given `*name*` property, and the "replaceOnCreate" argument was false. An `_existingId_` property of type `_Id_` MUST be included on the SetError object with the id of the existing SieveScript.

o `*tooManyScripts*`:

Creating the SieveScript would exceed the `*maxNumberScripts*` limit (see [Section 1.3.1](#)).



For "create" and "update":

- o `*invalidScript*`:

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. The `_description_` property on the SetError object SHOULD contain a specific error message giving the line number of the first error.

For "destroy":

- o `*scriptIsActive*`:

The SieveScript is active and the "onSuccessActivateScript" argument was not present.

### **2.3. SieveScript/query**

This is a standard "/query" method as described in [[RFC8620](#)], [Section 5.5](#). A `_FilterCondition_` object has the following properties, any of which may be omitted:

- o `*name*`: "String"

The SieveScript "name" property contains the given string.

- o `*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:

- o "name"

- o "isActive"



#### **2.4. SieveScript/validate**

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

The method provides similar functionality to the CHECKSCRIPT command in [\[RFC5804\]](#).

The `*SieveScript/validate*` method takes the following arguments:

- o `*accountId*`: "Id"

The id of the account to use.

- o `*content*`: "String"

The raw octets of the script [\[RFC5228\]](#).

- o `*blobId*`: "Id" (immutable)

The id of the blob containing the raw octets of the script [\[RFC5228\]](#).

A client MUST include either a `_content_` or a `_blobId_` property. A request that includes neither or both properties MUST be rejected with an "invalidProperties" SetError. The `_content_` property, if used, MUST be encoded following the same procedure as for the `_content_` property in the SieveScript ([Section 2](#)) object. The server MUST check the script content for syntactic validity, which includes checking that all Sieve extensions mentioned in Sieve script "require" statement(s) are supported by the Sieve interpreter. (Note that if the Sieve interpreter supports the Sieve "ihave" extension [\[RFC5463\]](#), any unrecognized/unsupported extension mentioned in the "ihave" test MUST NOT cause the syntactic validation failure.)

The response has the following arguments:

- o `*accountId*`: "Id"

The id of the account used for this call.





- o `*error*`: "SetError|null"

A SetError object if the request or the script content invalid, or "null" if the script content is valid.

## **2.5. SieveScript/test**

This method is used by the client to ask the Sieve interpreter to evaluate a Sieve script against a given email and report what actions would be performed.

The `*SieveScript/test*` method takes the following arguments:

- o `*accountId*`: "Id"

The id of the account to use.

- o `*scriptId*`: "Id"

The id of the SieveScript to test against.

- o `*emailId*`: "Id"

The id of the Email [[RFC8621](#)] to test against.

- o `*envelope*`: "Envelope|null"

Information that the Sieve interpreter should assume was present in the SMTP transaction that delivered the email when evaluating "envelope" tests. If "null", all "envelope" tests MUST evaluate to false. See [Section 7](#) of Email [[RFC8621](#)] for the contents of the Envelope object.

- o `*lastVacationResponse*`: "Date|null"

The date-time at which the Sieve interpreter should assume that it last auto-replied to the sender of the email, or "null" if the



Sieve interpreter should assume that it has not auto-replied to the sender.

The response has the following arguments:

- o \*accountId\*: "Id"

The id of the account used for this call.

- o \*actions\*: "String[]|null"

A set of strings listing the actions that would be performed, or "null" if evaluation of the script failed for any reason.

- o \*error\*: "SetError|null"

A SetError object if the request, the SieveScript, or the Email is invalid, or if the Sieve interpreter experienced a run-time error. Otherwise, "null" indicates that the script evaluation completed successfully. A "serverFail" SetError (see [Section 3.6.2 of \[RFC8620\]](#)) MUST be used to indicate a Sieve interpreter run-time error.

### **3. Security Considerations**

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

### **4. IANA Considerations**

#### **4.1. JMAP Capability Registration for "sieve"**

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

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

Specification document: this document

Intended use: common



Change Controller: IETF

Security and privacy considerations: this document, [Section 3](#)

#### **[4.2.](#) JMAP Error Codes Registry**

The following sub-section register several new error codes in the JMAP Error Codes registry, as defined in [[RFC8620](#)].

##### **[4.2.1.](#) `scriptNameExists`**

JMAP Error Code: `scriptNameExists`

Intended use: common

Change controller: IETF

Reference: This document, [Section 2.2](#)

Description: The client tried to create a SieveScript with the same "name" property as an existing SieveScript and the "replaceOnCreate" argument was false. present.

##### **[4.2.2.](#) `tooManyScripts`**

JMAP Error Code: `tooManyScripts`

Intended use: common

Change controller: IETF

Reference: This document, [Section 2.2](#)

Description: Creating the SieveScript would exceed the "maxNumberScripts" limit.

##### **[4.2.3.](#) `invalidScript`**

JMAP Error Code: `invalidScript`

Intended use: common

Change controller: IETF

Reference: This document, [Section 2.2](#)



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.

#### **4.2.4. scriptIsActive**

JMAP Error Code: scriptIsActive

Intended use: common

Change controller: IETF

Reference: This document, [Section 2.2](#)

Description: The client tried to destroy the active SieveScript, but the "OnSuccessActivateScript" argument was not present.

### **5. 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: Bron Gondwana, Alexey Melnikov, and Ricardo Signes.

### **6. References**

#### **6.1. 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>>.
- [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>>.
- [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>>.





- [RFC5322] Resnick, P., Ed., "Internet Message Format", [RFC 5322](#), DOI 10.17487/RFC5322, October 2008, <<https://www.rfc-editor.org/info/rfc5322>>.
- [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>>.
- [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>>.
- [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>>.
- [RFC8259] Bray, T., Ed., "The JavaScript Object Notation (JSON) Data Interchange Format", STD 90, [RFC 8259](#), DOI 10.17487/RFC8259, December 2017, <<https://www.rfc-editor.org/info/rfc8259>>.
- [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>>.

## **6.2. 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>>.



**Appendix A. Change History (To be removed by RFC Editor before publication)**

Changes since murchison-01:

- o Explicitly stated that Sieve capability strings are case-sensitive.
- o `errorDescription` is now `String|null`.
- o Added `/query` method.
- o Added `/test` method.

Changes since murchison-00:

- o Added IANA registration for "scriptIsActive" JMAP error code.
- o Added open issue about `/set{create}` with an existing script name.

Author's Address

Kenneth Murchison  
Fastmail US LLC  
1429 Walnut Street - Suite 1201  
Philadelphia, PA 19102  
USA

Email: [murch@fastmailteam.com](mailto:murch@fastmailteam.com)

