Network Working Group
Request for Comments: 5466
Category: Standards Track

A. Melnikov
C. King
Isode Ltd

February 2009

IMAP4 Extension for Named Searches (Filters)

Status of This Memo

Copyright Notice

Abstract

The document defines a way to persistently store named IMAP (RFC 3501) searches on the server. Such named searches can be subsequently referenced in a SEARCH or any other command that accepts a search criterion as a parameter.

Table of Contents

1. Introduction and Overview

Persistent named searches described in this document allow clients to save favorite searches on the server. Such saved searches can save bandwidth for clients that need to regularly repeat them.

The FILTERS IMAP extension adds a new FILTER search criterion for referencing persistent named searches (a.k.a. "filters"), as well as reuses GETMETADATA/SETMETADATA commands [METADATA] for listing/creating/updating/deleting such filters.

A filter can be private (only accessible to the logged-in user) or public (accessible to all logged-in users). Both a private and a public filter with the same name can exist at the same time. If both filter types with the same name exist, the FILTER SEARCH criterion (see Section 3.1) MUST use the value of the private filter; otherwise, it MUST use the value of the filter that exists.

Let us call a pair of filter name and filter type a "typed filter". Each typed filter can have a value (which is a valid IMAP SEARCH criteria conforming to ABNF for the "search-criteria" non-terminal) and an optional human-readable description. The SETMETADATA command creates or updates the value and/or the description of a typed filter.

Values of all search keys stored in a filter MUST be encoded in UTF-8.

2. Conventions Used in This Document

In examples, "C:" and "S:" indicate lines sent by the client and server, respectively. If a single "C:" or "S:" label applies to multiple lines, then the line breaks between those lines are for editorial clarity only and are not part of the actual protocol exchange.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

Basic familiarity with the METADATA-SERVER extension [METADATA] and terms defined therein is required to understand this document.

3. IMAP Protocol Changes

The IMAP extension for persistent named searches is present in any IMAP4 implementation that advertises "FILTERS" as one of the supported capabilities in the CAPABILITY response or response code.

3.1. FILTER SEARCH Criterion

The FILTER criterion for the SEARCH command allows a client to reference by name a filter stored on the server. Such filter was created by setting the server annotation named "/private/filters/values/<filter_name>" (or the server annotation "/shared/filters/values/<filter_name>", if "/private/filters/values/<filter_name>" doesn't exist) using the SETMETADATA command as described in Section 3.2.

Syntax: FILTER <filter_name>

When the named filter exists, its search criterion (i.e., the associated entry value) is inserted verbatim instead of the FILTER search-key. For example, the following SEARCH command

C: a SEARCH UID 300:900 FILTER on-the-road SINCE "3-Dec-2002"

would be equivalent to the following

C: a SEARCH UID 300:900 OR SMALLER 5000 FROM "boss@example.com" SINCE "3-Dec-2002"

assuming the filter "on-the-road" exists and contains the value 'OR SMALLER 5000 FROM "boss@example.com"'.

A reference to a nonexistent or unaccessible (e.g., due to access control restrictions) filter MUST cause failure of the SEARCH command with the tagged NO response that includes the UNDEFINED-FILTER response code followed by the name of the nonexistent/unaccessible filter.

Note the server SHOULD verify that each search criterion referenced by the FILTER search key is a full and correct search criterion. For example, the server should fail the SEARCH command if its SEARCH criterion references a filter containing "OR SMALLER" search criterion, because this value is lacking one parameter and thus is not a fully specified search criterion.

Note that a named filter itself can reference another filter using the FILTER search-key. Implementations MUST be able to perform at least 3 substitution passes on the SEARCH command criterion. If an implementation allows for more passes, it MUST implement some kind of loop detection. If an implementation detects a loop or still sees a FILTER search-key after performing at least 3 substitutions, it MUST behave as if the specified filter doesn't exist (as described above).

Note that use of the FILTER search key implies the CHARSET "UTF-8" parameter to the SEARCH/UID SEARCH command. If the SEARCH/UID SEARCH command includes the explicit CHARSET parameter with the value other than "UTF-8" or "US-ASCII", then such command MUST result in the tagged BAD response from the server. Such tagged response MUST contain the BADCHARSET response code (see [RFC3501]).

3.2. Managing Filters Using SETMETADATA/GETMETADATA Commands

Any server compliant with this document MUST either implement the METADATA-SERVER (or METADATA) [METADATA] extension, or implement SETMETADATA/GETMETADATA commands described in [METADATA] so that they work for the case of empty mailbox name (i.e., for managing server annotations) and for the entries specified in this section.

This document reserves two hierarchies of per-server entries under the "/private/filters/values" and "/shared/filters/values" roots (see [METADATA]) for storing filter values. The value of a "/private/filters/values/<filter_name>" or a "/shared/filters/values/<filter_name>" server annotation is an IMAP SEARCH criteria, conforming to ABNF for the "search-criteria" non-terminal. A name of a filter is governed by the ABNF for the "filter-name" non-terminal.

Note that values of all search keys stored in these entries MUST be encoded in UTF-8.

A new filter named "<filter_name>" can be created (or an existing filter can be modified) by storing a non-NIL value in the "/private/filters/values/<filter_name>" server entry (or in the "/shared/filters/values/<filter_name>") using the SETMETADATA [METADATA] command. The server SHOULD verify that each search criterion stored in such a server entry is a full and correct search criterion.

A filter can be deleted by storing the NIL value in both the "/private/filters/values/<filter_name>" and the "/shared/filters/values/<filter_name>" entries.

A filter can be renamed by first creating a filter with the new name (that has the same value as the old one) and then deleting the filter with the old one.

If both "/private/filters/values/<filter_name>" and "/shared/filters/values/<filter_name>" server annotations exist, then the value of the "/private/filters/values/<filter_name>" is used when evaluating the corresponding FILTER SEARCH key (see Section 3.1). Otherwise the non-NIL (existent) value is used.

If the server is unable to create a new typed filter because the maximum number of allowed filters has already been reached, the server MUST return a tagged NO response with a "[METADATA TOOMANY]" response code, as defined in [METADATA].

```
C: a007 SETMETADATA "" ("/private/filters/values/on-the-road"
    "OR SMALLER 5000 FROM \"boss@example.com\"")
S: a007 OK SETMETADATA complete
```

Client implementation note: As multiple clients might read and write filter values, it is possible that one client will use a SEARCH key that might not be recognized by another client that tries to present a user interface for editing a filter value. In order to help other clients to (partially) parse filter values for editing purposes, a client storing a filter value SHOULD use () around any SEARCH key not defined in [RFC3501]. For example, if there is an IMAP extension that defines a new x-dsfa SEARCH key that takes 2 parameters, then the following SEARCH criterion 'from "@example.com>" x-dsfa from 5' should be stored as 'from "@example.com>" (x-dsfa from 5)'.

Note that filter names are restricted to a subset of US-ASCII, as described in Section 4. So they might not always be meaningful to users and thus not necessarily suitable for display purposes. In order to help with storing human-readable descriptions of filters, this document also reserves two hierarchies of server entries under the "/private/filters/descriptions" and "/shared/filters/descriptions" roots. The value of a "/private/filters/descriptions/<filter_name>" or a "/shared/filters/descriptions/<filter_name>" server annotation is a human-readable description for the <filter_name> filter, encoded in UTF-8 [UTF-8]. If the "/private/filters/descriptions/<filter_name>" server annotation exists, its value is used by the client as the filter description. Otherwise, the value of the "/shared/filters/descriptions/<filter_name>" server annotation is used as the filter description. In the absence of both the "/private/filters/descriptions/<filter_name>" and the "/shared/filters/descriptions/<filter_name>" entries, the client MAY display the name of the filter as its description.

The description string SHOULD be annotated with one or more language tags [RFC4646] as specified in Chapter 16.9 of [Unicode]. In the absence of any language tag, the "i-default" [RFC2277] language SHOULD be assumed. Description in multiple languages MAY be present in a single description string. This is done by concatenating descriptions in multiple languages into a single string, each description prefixed with its language tag, for example "<ru><...description in Russian...><fr-ca><...description in French...>". Note that here <ru> is a language tag consisting of 3 Unicode characters: <U+E0001>, <U+E0072>, <U+E0075>; and <fr-ca> is a

language tag consisting of 6 Unicode characters: <U+E0001>, <U+E0066>, <U+E0072>, <U+E002D>, <U+E0063>, <U+E0061>.

4. Formal Syntax

The following syntax specification uses the Augmented Backus-Naur Form (ABNF) notation as specified in [ABNF].

Non-terminals referenced but not defined below are as defined by [RFC3501] or [IMAPABNF].

Except as noted otherwise, all alphabetic characters are case-insensitive. The use of upper or lower case characters to define token strings is for editorial clarity only. Implementations MUST accept these strings in a case-insensitive fashion.

```
capability            =/ "FILTERS"

search-criteria       =  search-key *(SP search-key)

search-key            =/  "FILTER" SP filter-name
                      ;; New SEARCH criterion for referencing filters

filter-name           =  1*<any ATOM-CHAR except "/">
                      ;; Note that filter-name disallows UTF-8 or
                      ;; the following characters: "(", ")", "{",
                      ;; " ", "%", "*", "]".  See definition of
                      ;; ATOM-CHAR [RFC3501].

resp-text-code        =/  "UNDEFINED-FILTER" SP filter-name
```

5. Security Considerations

General issues relevant to [RFC3501] (in particular to the SEARCH command) and METADATA-SERVER extension [METADATA] are also relevant to this document.

Note that excessive use of filters can potentially simplify denial-of-service attacks, especially if combined with poor implementations and lack of loop detection (i.e., detection of filters referencing each other to create a loop). Servers that allow for anonymous access SHOULD NOT allow anonymous users to create/edit/delete filters.

Also note that stored filters can potentially disclose personal information about users. When confidentiality of such information is important, clients MUST use TLS and/or SASL security layer (or similar) as recommended in [RFC3501]. Also, clients should use

private filters instead of public, unless they desire to share such information with other users.

As always, it is important to thoroughly test clients and servers when implementing this extension.

6. IANA Considerations

IMAP4 capabilities are registered by publishing a Standards Track or IESG-approved Experimental RFC. The IMAP4 capabilities registry is available from http://www.iana.org.

This document defines the FILTERS IMAP capability. IANA has added it to the registry.

IANA has added the following 4 entries to the [METADATA] registry:

To: iana@iana.org
Subject: IMAP METADATA Entry Registration
Type: Server
Name: /private/filters/values/<filter_name>
Description: Contains an IMAP SEARCH criteria. Defined in RFC 5466.
Content-type: text/plain; charset=utf-8
Contact person: Alexey Melnikov
Email: alexey.melnikov@isode.com

To: iana@iana.org
Subject: IMAP METADATA Entry Registration
Type: Server
Name: /shared/filters/values/<filter_name>
Description: Contains an IMAP SEARCH criterion. Defined in RFC 5466.
Content-type: text/plain; charset=utf-8
Contact person: Alexey Melnikov
Email: alexey.melnikov@isode.com

To: iana@iana.org
Subject: IMAP METADATA Entry Registration
Type: Server
Name: /private/filters/descriptions/<filter_name>
Description: Contains a user-specific human-readable description of a named SEARCH criterion stored in the /private/filters/values/<filter_name> or /shared/filters/values/<filter_name> annotation. The value is in UTF-8. Defined in RFC 5466.
Content-type: text/plain; charset=utf-8
Contact person: Alexey Melnikov
Email: alexey.melnikov@isode.com

To: iana@iana.org
Subject: IMAP METADATA Entry Registration
Type: Server
Name: /shared/filters/descriptions/<filter_name>
Description: Contains a global (shared among all users) human-readable description of a named SEARCH criterion stored in the /private/filters/values/<filter_name> or /shared/filters/values/<filter_name> annotation. The value is in UTF-8. Defined in RFC 5466.
Content-type: text/plain; charset=utf-8
Contact person: Alexey Melnikov
Email: alexey.melnikov@isode.com

7. Acknowledgments

Thanks to David Cridland, Arnt Gulbrandsen, Chris Newman, and Timo Sirainen for comments and suggestions on this document. Special thank you to Brian E. Carpenter for the GenArt review.

8. Normative References

[ABNF] Crocker, D., Ed. and P. Overell, Ed., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008.

[IMAPABNF] Melnikov, A. and C. Daboo, "Collected Extensions to IMAP4 ABNF", RFC 4466, April 2006.

[METADATA] Daboo, C., "The IMAP METADATA Extension", RFC 5464, February 2009.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, January 1998.

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003.

[RFC4646] Phillips, A. and M. Davis, "Tags for Identifying Languages", BCP 47, RFC 4646, September 2006.

[UTF-8] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, November 2003.

[Unicode] "The Unicode Standard 5.0", Unicode 5.0, 2007, <http://www.unicode.org/versions/Unicode5.0.0/>.

Authors' Addresses

Alexey Melnikov
Isode Ltd
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

EMail: Alexey.Melnikov@isode.com

Curtis King
Isode Ltd
5 Castle Business Village
36 Station Road
Hampton, Middlesex TW12 2BX
UK

EMail: Curtis.King@isode.com