

Groupe de travail Réseau
Request for Comments : 5463
Catégorie : Sur la voie de la normalisation

N. Freed, Sun Microsystems
mars 2009
Traduction Claude Brière de L'Isle

Filtrage de messagerie Sieve : extension Ihave

Statut du présent mémoire

Le présent document spécifie un protocole de l'Internet sur la voie de la normalisation pour la communauté de l'Internet, et appelle à des discussions et suggestions pour son amélioration. Prière de se référer à l'édition en cours des "Protocoles officiels de l'Internet" (STD 1) pour voir l'état de normalisation et le statut de ce protocole. La distribution du présent mémoire n'est soumise à aucune restriction.

Notice de droits de reproduction

Copyright (c) 2009 IETF Trust et les personnes identifiées comme auteurs du document. Tous droits réservés.

Le présent document est soumis au BCP 78 et aux dispositions légales de l'IETF Trust qui se rapportent aux documents de l'IETF (<http://trustee.ietf.org/license-info>) en vigueur à la date de publication de ce document. Prière de revoir ces documents avec attention, car ils décrivent vos droits et obligations par rapport à ce document.

Résumé

Le présent document décrit l'extension "ihave" au langage de filtrage de messagerie Sieve. L'extension "ihave" fournit un moyen pour écrire des scripts qui peuvent tirer parti des caractéristique facultatives de Sieve mais peuvent quand même fonctionner quand ces caractéristiques facultatives ne sont pas disponibles. L'extension définit aussi une nouvelle commande de contrôle d'erreur destinée à être utilisée pour rapporter des situations où aucune combinaison des extensions disponibles ne satisfait les besoins du script.

Table des matières

1. Introduction.....	1
2. Conventions utilisées dans ce document.....	2
3. Identifiants de capacités.....	2
4. Essai Ihave.....	2
5. Contrôle d'erreur.....	3
6. Considérations sur la sécurité.....	3
7. Considérations relatives à l'IANA.....	3
8. Références.....	3
8.1 Références normatives.....	3
8.2 Références pour information.....	4
9. Remerciements.....	4
Adresse de l'auteur.....	4

1. Introduction

Sieve [RFC5228] est un langage pour filtrer les messages électroniques au moment ou autour du moment de la livraison finale. Il est conçu pour être mis en œuvre sur un client ou un serveur de messagerie électronique. Il convient pour un serveur de messagerie électronique où les utilisateurs peuvent n'avoir pas la permission d'exécuter des programmes arbitraires, comme ceux d'un serveur de boîte noire du protocole d'accès au message Internet [RFC3501], car il n'a pas de terminaison contrôlée par l'utilisateur ou la capacité de faire appel à des programmes externes.

Diverses extensions à Sieve ont déjà été définies, par exemple, les [RFC5229], [RFC5230], [RFC5231], [RFC5232], [RFC5233], [RFC5235], et de nombreuses autres sont sûres d'être créées bientôt. La clause d'exigence de Sieve est utilisée pour spécifier les extensions dont un Sieve particulier a besoin : un résultat d'erreur si la clause d'exigence du script invoque une extension qui n'est pas disponible. Ce mécanisme est suffisant dans la plupart des situations. Cependant, il peut y avoir des cas où un script peut être capable de tirer parti d'une extension si elle est disponible mais peut quand même fonctionner si elle ne l'est pas, éventuellement avec une certaine dégradation de la fonctionnalité. Il peut aussi y avoir des cas où un script préférerait une extension mais peut en employer une différente si la première n'est pas disponible.

L'extension "ihave" donne le moyen d'écrire des scripts qui utilisent des extensions seulement quand elles sont réellement disponibles. Elle définit un nouvel essai "ihave" qui prend une liste de noms de capacités comme argument et réussit si et seulement si toutes ces capacités sont présentes. De plus, la spécification de l'extension "ihave" dans la clause "require" désactive la vérification au moment de l'analyse de l'utilisation de l'extension dans les scripts ; la vérification au moment du fonctionnement doit être utilisée à la place. Cela rend possible d'écrire des scripts portables qui peuvent fonctionner dans plusieurs environnements, rendant effective l'utilisation de toute extension disponible même si des ensembles d'extensions différents sont fournis à différents endroits.

L'extension "ihave" définit aussi une nouvelle commande de contrôle d'erreur. Une erreur cause la terminaison de l'exécution du script avec le message d'erreur donné comme argument au contrôle d'erreur.

2. Conventions utilisées dans ce document

Les mots clés "DOIT", "NE DOIT PAS", "EXIGE", "DEVRA", "NE DEVRA PAS", "DEVRAIT", "NE DEVRAIT PAS", "RECOMMANDE", "PEUT", et "FACULTATIF" en majuscules dans ce document sont à interpréter comme décrit dans le BCP 14, [RFC2119].

Les termes utilisés pour décrire les divers composants du langage Sieve sont tirés du paragraphe 1.1 de la [RFC5228].

3. Identifiants de capacités

La chaîne de capacités associée à l'extension définie dans le présent document est "ihave".

4. Essai Ihave

Usage : `ihave <capacités : liste de chaînes>`

L'essai "ihave" donne aux scripts Sieve le moyen de vérifier l'existence d'une certaine extension avant de l'utiliser réellement. L'argument de capacités de "ihave" est le même que l'argument de nom similaire à la déclaration de contrôle "require" : il spécifie le nom d'une ou plusieurs extensions ou comparateurs Sieve. L'essai "ihave" réussit si toutes les extensions spécifiées dans la liste des capacités sont disponibles pour le script.

À la différence de la plupart des essais Sieve, "ihave" n'accepte pas d'argument de correspondance ou de comparaison. Le type de correspondance pour "ihave" est toujours ":is" et le comparateur est toujours "i;octet".

Les chaînes dans la liste de capacités sont des chaînes constantes dans le contexte des variables Sieve [RFC5229]. C'est une erreur de passer une chaîne non constante comme argument de "ihave".

La spécification de base de Sieve demande que toutes les extensions Sieve utilisées dans un certain script soient spécifiées dans la déclaration initiale de commande "require". C'est une erreur pour un script d'invoquer des extensions que l'interpréteur ne prend pas en charge ou de tenter d'utiliser des extensions qui ne figurent pas sur la liste de la clause "require" du script. Utiliser "ihave" change le comportement de l'interpréteur Sieve et les exigences sous-jacentes de la façon suivante :

1. L'utilisation d'une extension donnée est permise suite à l'évaluation réussie d'un essai "ihave" sur cette extension jusqu'à la fin du script, même en-dehors du bloc inclus dans l'essai "ihave". En d'autres termes, suite à un "ihave" réussi, les choses se passent juste comme si l'extension avait été spécifiée dans la clause "require" du script. L'extension ne peut pas être utilisée avant l'évaluation d'un tel essai et une erreur de démarrage DOIT être générée si un tel usage est tenté. Cependant, l'utilisation ultérieure de cette extension peut encore devoir être traitée de façon conditionnelle via un essai "ihave" pour traiter le cas où elle n'est pas prise en charge.
2. Les interpréteurs Sieve ont normalement l'option de vérifier l'utilisation de l'extension au moment de l'analyse ou au moment de l'exécution. La spécification de "ihave" dans la clause "require" d'un script change ce comportement : les scripts DOIVENT différer la vérification de l'extension au moment de l'exécution ou autrement prendre la présence des essais "ihave" en compte au moment de l'analyse. Noter que comme "ihave" peut être utilisé à l'intérieur des essais

"anyof", "allof", et "not", la vérification au moment de l'analyse de "ihave" peut être très difficile à mettre en œuvre.

3. Bien qu'il y ait peu de sens à le faire, une extension peut être spécifiée à la fois dans la déclaration de commande "require" et dans un essai "ihave". Si cela est fait et si l'extension a été mise en œuvre, l'extension peut être utilisée n'importe où dans le script et un essai "ihave" de cette extension va toujours retourner "vrai".
4. L'essai "ihave" accepte une liste de capacités. Si une des capacités spécifiées est indisponible, l'essai échoue et aucune des capacités n'est activée.
5. La spécification Sieve de base n'exige pas que les interpréteurs évaluent les arguments dans un ordre particulier ou que l'essai d'évaluation soit court-circuité. Si "ihave" est activé, l'interpréteur DOIT court-circuiter les essais, c'est-à-dire, ne pas effectuer plus d'essais que nécessaire pour trouver le résultat. De plus, l'ordre d'évaluation DOIT être de gauche à droite si "ihave" est activé.

L'extension "ihave" est conçue pour être utilisée avec d'autres extensions qui ajoutent des essais, actions, comparateurs, ou arguments. Les mises en œuvre NE DOIVENT PAS lui permettre d'être utilisée avec des extensions qui changent la grammaire Sieve sous-jacente, ou des extensions comme de caractères codés [RFC5228], ou variables [RFC5229] qui changent la façon dont le contenu des scripts Sieve est interprétée. L'essai DOIT échouer et l'extension NE DOIT PAS être activée si un tel usage est tenté.

5. Contrôle d'erreur

Usage : erreur <message : chaîne>

Le contrôle d'erreur cause la fin de l'exécution du script avec une erreur au moment de l'exécution. L'argument de message fournit une description textuelle de la condition d'erreur qui DEVRAIT être incluse dans tout rapport généré concernant l'erreur. Le paragraphe 2.10.6 de la [RFC5228] décrit comment sont traitées les erreurs au démarrage dans Sieve.

Noter que l'argument de message, comme toutes les chaînes Sieve, emploie le jeu de caractères UTF-8 et peut contenir des caractères non US-ASCII. Cela doit être pris en considération quand on rapporte des erreurs de script.

Le contrôle d'erreur est inclus au titre de l'extension "ihave" afin qu'il soit inconditionnellement disponible pour les scripts qui utilisent ihave.

6. Considérations sur la sécurité

Un problème potentiel de sécurité des scripts Sieve apparaît quand un script échoue à cause de l'absence d'une certaine extension, il peut échouer à bloquer des messages dangereux. L'extension "ihave" rend possible d'améliorer la portabilité du script et généralement, cela peut améliorer la sécurité globale fournie par Sieve.

La robustesse du script mise à part, ihave est essentiellement une variante plus souple du mécanisme "require" existant de Sieve. À ce titre, il n'ajoute pas de capacités supplémentaires à la mise en œuvre de Sieve qui pourraient créer des problèmes de sécurité. Bien sûr, toutes les considérations de sécurité données dans la spécification Sieve de base et dans toutes les extensions qui sont employées sont encore pertinentes.

7. Considérations relatives à l'IANA

Le gabarit suivant spécifie l'enregistrement par l'IANA de l'extension Sieve spécifiée dans le présent document:

Pour : iana@iana.org

Sujet : enregistrement d'une nouvelle extension Sieve

Nom de capacité : ihave

Description : l'extension "ihave" donne le moyen d'écrire des scripts qui font usage des autres extensions seulement quand elles sont réellement disponibles.

Numéro de RFC : RFC 5463

Adresse de contact : liste de diffusion Sieve à <ietf-mta-filters@imc.org>

8. Références

8.1 Références normatives

- [RFC2119] S. Bradner, "[Mots clés à utiliser](#) dans les RFC pour indiquer les niveaux d'exigence", BCP 14, mars 1997. (MàJ par [RFC8174](#))
- [RFC5228] P. Guenther et autres, "[Sieve : un langage de filtrage](#) de messagerie électronique", janvier 2008. (P.S. ; Remplace [RFC3028](#), MàJ par [RFC5229](#), [5429](#) 9042)

8.2 Références pour information

- [RFC3501] M. Crispin, "Protocole d'[accès au message Internet - version 4rev1](#)", mars 2003. (P.S. ; MàJ par [RFC4466](#), [4469](#), [4551](#), [5032](#), [5182](#), [7817](#), [8314](#), [8437](#), [8474](#) ; remplacée par la RFC9051
- [RFC5229] K. Homme, "[Filtrage de messagerie Sieve](#) : extension Variables", janvier 2008. (MàJ par [RFC5173](#)) (P.S.)
- [RFC5230] T. Showalter, N. Freed, éd., "[Filtrage de messagerie Sieve](#) : extension Vacation", janvier 2008. (P.S. ; MàJ par [RFC8580](#).)
- [RFC5231] W. Segmuller, B. Leiba, "[Filtrage de messagerie Sieve](#) : extension Relational", janvier 2008. (Remplace [RFC3431](#)) (P.S.)
- [RFC5232] A. Melnikov, "Filtrage de messagerie Sieve : [extension Imap4flags](#)", janvier 2008. (P.S.)
- [RFC5233] K. Murchison, "Filtrage de messagerie Sieve : [extension Subaddress](#)", janvier 2008. (Remplace [RFC3598](#)) (P.S.)
- [RFC5235] C. Daboo, "Filtrage de messagerie Sieve : [extensions Spamtest et Virustest](#)", janvier 2008. (Remplace [RFC3685](#)) (P.S.)

9. Remerciements

Stephan Bosch, Cyrus Daboo, Arnt Gulbrandsen, Andrew McKeon, et Alexey Melnikov ont fourni des suggestions et corrections utiles.

Adresse de l'auteur

Ned Freed
Sun Microsystems
800 Royal Oaks
Monrovia, CA 91016-6347
USA

téléphone : +1 909 457 4293
mél : ned.freed@mrochek.com