

RFC 5228 : Sieve: An Email Filtering Language

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 17 janvier 2008

Date de publication du RFC : Janvier 2008

<https://www.bortzmeyer.org/5228.html>

L'utilisation massive du courrier électronique fait que, même sans compter le spam, chaque utilisateur professionnel reçoit désormais des dizaines et souvent des centaines de messages quotidiens dans sa boîte aux lettres. Il n'est donc plus question de les gérer entièrement à la main, il faut disposer d'un mécanisme de filtrage automatique.

On n'imagine plus aujourd'hui d'utiliser le courrier au bureau sans un tel mécanisme, permettant de mettre les messages dans des dossiers séparés selon, par exemple, la liste de diffusion concernée. Il en existe plusieurs, certains bâtis sur une interface graphique, d'autres autour d'un langage. Ces derniers, bien plus riches et puissants, sont variés. Le plus connu est procmail, célèbre pour son pouvoir expressif (on peut tout faire en procmail) mais aussi pour sa difficulté.

Sieve, l'objet de ce RFC, est nettement moins riche que procmail, ce qui peut être un inconvénient pour les utilisateurs avancés mais un avantage pour l'administrateur : permettre aux utilisateurs de configurer procmail revient à leur donner un accès shell alors que Sieve est plus contrôlable. Comme le note bien notre RFC, Sieve n'est **pas** un langage de Turing (par exemple, il ne connaît pas les boucles). Ses capacités sont limitées, il est "*sandboxable*", c'est-à-dire que l'administrateur système peut facilement limiter les choses que risquent de faire les auteurs de scripts Sieve.

Autre avantage par rapport à procmail, Sieve est normalisé et il en existe plusieurs mises en œuvre, dans Cyrus, dans Dovecot, dans GNU mailutils, dans Archiveopteryx...

Les scripts Sieve sont écrits par l'utilisateur avec un éditeur ordinaire ou bien via une interface graphique. Ils sont installés sur le serveur de messagerie via FTP ou bien par le protocole spécifique "*Manage Sieve*" (RFC 5804¹). Le programme de webmail SquirrelMail, par exemple, dispose d'une interface d'édition de scripts <<http://email.uoa.gr/projects/squirrelmail/avelsieve.php>> Sieve, qui gère le protocole "*Manage Sieve*".

Voici un exemple de script Sieve inspiré du RFC et amélioré par Mathieu Arnold :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc5804.txt>

```
if header :contains "from" "jfc" {
  discard;
} elsif header :contains ["subject"] ["money", "viagra"] {
  discard;
} else {
  fileinto "INBOX";
}
```

Sieve lui-même est très limité mais le langage dispose d'un mécanisme d'extensions (qui doivent être déclarées dans le script avec `require`). Certaines sont définies dès notre RFC (comme `fileinto`), d'autres sont définies via d'autres RFC, par exemple pour utiliser des tests anti-spam ou antivirus (RFC 5235), ou bien pour tester sur des sous-adresses (RFC 5233).

Notre RFC est la mise à jour du RFC 3028, qui avait été le premier à normaliser Sieve. Les principaux changements (section 15) sont :

- Retrait de l'action `reject`, qui migrera vers un RFC séparé, le RFC 5429.
- Le mécanisme de comparaison de chaînes de caractères, dans la section 2.7.3, auparavant défini spécifiquement pour Sieve, est désormais dans une famille de RFC séparée, introduite par le RFC 4790, qui décrit le registre des algorithmes de collation <<https://www.iana.org/assignments/collation/collation-index.html>>. Seuls les algorithmes `i;octet` et `i;ascii-casemap` sont directement dans Sieve, les autres sont des extensions (et doivent donc être déclarées avec `require`).
- La grammaire formelle a été corrigée. Un des traditionnels talons d'Achille des RFC est que les parties en langage formel, comme les grammaires en ABNF ou les MIB ne sont pas testées par un programme avant publication et que des RFC peuvent donc être publiés avec du code invalide. Notons aussi que notre RFC (section 8) est un des très rares RFC où les structures lexicales (les "*tokens*") et syntaxiques sont séparées. La règle la plus fréquente dans les RFC est au contraire que la grammaire décrive les deux types de structure ensemble.

Pour les utilisateurs de procmail, il existe un script (que je n'ai **pas** testé) pour migrer certaines règles : <<http://www.earth.ox.ac.uk/~steve/sieve/procmail2sieve.pl>>. Si vous êtes curieux des extensions Sieve, vous pouvez lire tous les articles de mon blog qui en parlent </search?pattern=sieve>.

Notez que certains hébergeurs de courrier permettent à leurs clients d'écrire des scripts Sieve, par exemple Gandi <<https://news.gandi.net/fr/2017/03/activation-des-regles-serveur-sieve-sur>>. (Ils ne pourraient pas le faire avec un langage plus général comme procmail, pour des raisons de sécurité.)