managesieve Documentation

Release 0.6

Hartmut Goebel

Jun 27, 2018

User Documentation

A pure Python application (the interactive *sieveshell*) for remotely managing Sieve scripts. For developers it includes a Python module implementing the ManageSieve client protocol (**RFC 5804**).

CHAPTER 1

Download & Installation

1.1 Instructions for Windows Users

1. *managesieve* requires Python. If you don't have Python installed already, download and install Python 3.6 from https://python.org/download/3.6/

   During installation, make sure to check "Include into PATH".

2. If you already have Python installed, please check that your Python directory (normally `C:Python36` for python 3.6) and the Python Scripts directory (normally :file'C:Python36Scripts') are in the system path. If not, just add them in *My Computer → Properties → Advanced → Environment Variables* to the `Path` system variable.

3. Install *managesieve* by running

```
pip install managesieve
```

   Then run the console command `managesieve --help` to get detailed help.

   If the command `pip` is unknown to you system, please refer to the pip homepage for help.

1.2 Instructions for GNU/Linux and other Operating Systems

Most current GNU/Linux distributions provide packages for *managesieve*. Simply search your distribution's software catalog.

Also many vendors provide Python, and some even provide *managesieve*. Please check your vendor's software repository.

If your distribution or vendor does not provide a current version of *managesieve* please read on.

If your vendor does not provide **python** please download Python 3.6 from https://www.python.org/download/ and follow the installation instructions there.

If you distribution or vendor missed providing **pip**, alongside **python**, please check your vendor's or distribution's software repository for a package called *pip* or *python-pip*. If this is not provided, please refer to the pip homepage for help.

Then continue with *Installing managesieve using pip* below.

1.3 Installing *managesieve* using `pip`

After installing *Python* (and optionally *PyPDF2*), just run:

```
sudo pip install managesieve
```

to install *managesieve* for all users. For installing *managesieve* for yourself only, run:

```
pip install --user managesieve
```

If your system does not have network access download *managesieve* from https://pypi.org/project/managesieve/, and run

```
sudo pip install managesieve-*.tar.gz
```

respective

```
pip install --user managesieve-*.tar.gz
```

Donations

If you like managesieve, please consider supporting me in some way.

Bank transfers

While PayPal donations are still very much appreciated PayPal takes a large cut and has rather questionable business practices.

If you have the ability to make SEPA bank transfers at a low cost (for example if you live within the EU) please contact Hartmut to ask for his bank details.

I trust in the power of free and open source software and thus made the entire source code publicly available for every one to use. However, it takes a lot of time to develop the software and manage the community. And I still need to make my living.

So, if you like the software, please consider donating, especially if your organization benefits from this project.

Thank   you!

- **Bitcoin:** 13qryeeJR5Hc7vR5AmQMWDuLzDEgSDNJi2
- **Bank-Transfer (prefered):** If you have the ability to make SEPA bank transfers at a low cost please contact me and I'll send you the account details. If you want a bill, please state so and also leave your name and address.
- **PayPal:** Donate - if you want a bill, please state so and leave your name, address and email-address.
- Please contact me to arrange some kind of project grant, e.g. consulting or support.

Fig. 1: Bitcoin address

Changes

3.1 0.6 (2018-06-27)

Author: Hartmut Goebel

- Add support for Python 3. Minimum required Python version is now Python 2.7.

- Homepage is now hosted at https://managesieve.readthedocs.io/

- Documentation is extended and includes API documentation.

    sieveshell

    - Security fix: No longer leak environment variable SIEVE_PASSWORD when displaying usage help.

    - Per default enforce secure transport. Suggested by Jan Zerebecki.

    - Add possibility to use username/password from the .netrc file. The order is: command line options -> environment variables -> .netrc file -> ask user. Thanks to Grégoire Détrez.

    managesieve

    - Fail if TLS is requested, but server doesn't support TLS. Suggested by Jan Zerebecki.

    project

    - Rework and enhance test-suite. Thanks to Matěj Cepl for nudging to proper pytest integration.

    - Lots if internal cleanup.

3.2 Version 0.5 (2013-03-24)

Author: Hartmut Goebel

    sieveshell

- Changed default port for the MANAGESIEVE protocol to 4190 as proposed by RFC 5804. Thanks to Guido Berhoerster for submitting the patch.

- Added option `--port`. Thanks to Damien Aumaitre for submitting the patch and to Guido Berhoerster for an enhancement.

- Added option `--verbose` for controlling debug output in managesieve. May be given several times to increase verbosity.

managesieve

- Switched to Python standard logging system. This introduces a minor interface change: MANAGESIEVE.debug and global Debug are gone. See source for information about debugging log levels.

3.3 Version 0.4.2 (2010-01-06)

Author: Hartmut Goebel

managesieve

- Use ssl.wrap_socket() instead of deprecated socket.ssl(). Thanks to Guido Berhoerster for submitting the patch.

3.4 Version 0.4.1 (2009-10-08)

Author: Hartmut Goebel

managesieve

- fixed short read (thanks to paurkedal for submitting the patch)

3.5 Version 0.4 (2008-10-13)

Author: Hartmut Goebel

managesieve

- now work with Python 2.3 and later

- added support for TLS (STARTTLS), special thanks to Gregory Boyce for fixing some corner cases here

- added support for PLAIN authentication

- use optparse if available instead of optik.

- API change: login() no longer uses the LOGIN authentication mechanism, but has become a convenience function. It uses the best mechanism available for authenticating the user.

- Bugfixes:

  - If a capability had no values (like 'STARTTLS'), parsing capabilities failed. Fixed.

  - removed dependency on some imaplib's private functions

  - fixed typo: self.supports_tls

Thanks to Tomas 'Skitta' Lindroos, Lorenzo Boccaccia, Alain Spineux, darkness and Gregory Boyce for sending patches.

sieveshell

- added support for different authentication mechanisms

- added option –start-tls

- more verbose output when server says "BYE" (prints out referrals etc.)

- automatic cmd_quit() when server says "BYE"

- several minor bugfixes and enhancements

3.6 Version 0.3

Author: Hartmut Goebel

managesieve

- MANAGESIEVE.authenticate() now only returns a OK/NO/BYE result like any command not asking data from the server

sieveshell

- added 'edit', which may create scripts, too. (posix only)

- now prints out the server capabilities, thus the user knows what the server is capable of (and which Sieve-Commands may be used).

- fixed some minor bugs

3.7 Version 0.2

Author: Hartmut Goebel <h.goebel@crazy-compilers.com>

- renamed to 'managesieve'

- added sieveshell

- major overhaul

- added support for HAVESPACE, CAPABILITY

- added unittest testsuite (covers most cases)

3.8 Version 0.1

Author: Ulrich Eck <ueck at net-labs.de>

- initial release of Sieve.py by Ulrich Eck as part of 'ImapClient' (see http://www.zope.org/Members/jack-e/ImapClient), a Zope product.

Development

The source of *managesieve* is maintained at GitLab. Patches and pull-requests are hearty welcome.

- Please submit bugs and enhancements to the Issue Tracker.

- You may browse the code at the Repository Browser Or you may check out the current version by running

```
git clone https://gitlab.com/htgoebel/managesieve.git
```

Historical Note: managesieve was hosted at origo.ethz.ch, which closed in May 2012. Then *managesieve* was hosted on gitorious.org, which was closed in May 2015 and merged into gitlab.

Module Documentation

5.1 managesieve

ManageSieve (RFC 5804) client module for remotely managing Sieve Scripts.

All Sieve commands are supported by methods of the same name (in lower-case). All arguments to commands are converted to strings, except for authenticate().

class managesieve.**MANAGESIEVE**(*host=''*, *port=4190*, *use_tls=False*, *keyfile=None*, *certfile=None*)
  Sieve client class.

  Instantiate with: MANAGESIEVE(host [, port])

  Parameters

  - **host** – host's name (default: localhost)
  - **port** – port number (default: standard Sieve port).
  - **use_tls** – switch to TLS automatically, fail if the server doesn't support STARTTLS
  - **keyfile** – keyfile to use for TLS (optional)
  - **certfile** – certfile to use for TLS (optional)

exception abort
  Service errors - close and retry

authenticate(*mechanism*, **authobjects*)
  Authenticate to the server.

  Parameters

  - **mechanism** (*str*) – authentication mechanism to use
  - **authobjects** – authentication data for this mechanism

  Returns response (OK, NO, BYE)

capability()
Issue a CAPABILITY command and return the result.

As a side-effect, on succes these attributes are (re)set:

- `capabilities` (list of strings)
- `loginmechs` (list of strings)
- `implementation` (string)
- `supports_tls` (boolean)

> **Returns** tuple(response, capabilities) – If *response* is `OK`, *capabilities* is a list of strings.

deletescript(*scriptname*)
Delete a scripts at the server.

> **Parameters** **scriptname** (`str`) – name of script to be deleted
>
> **Returns** response (`OK`, `NO`, `BYE`)

exception error
Logical errors - debug required

getscript(*scriptname*)
Get a script from the server.

> **Parameters** **scriptname** (`str`) – name of script to be retrieved
>
> **Returns** tuple(response, str) – if *response* is `OK`, *str* is the script content.

havespace(*scriptname*, *size*)
Query the server for available space.

> **Parameters**
>
> - **scriptname** (`str`) – name of script to XXX
> - **size** (`int`) – XXX
>
> **Returns** response (`OK`, `NO`, `BYE`)

listscripts()
Get a list of scripts on the server.

> **Returns** tuple(response, [data]) – if *response* is `OK`, *data* is a list of *(scriptname, active)* tuples.

login(*auth*, *user*, *password*)
Authenticate to the Sieve server using the best mechanism available.

> **Returns** response (`OK`, `NO`, `BYE`)

logout()
Terminate connection to server.

> **Returns** response (`OK`, `NO`, `BYE`)

putscript(*scriptname*, *scriptdata*)
Put a script onto the server.

> **Parameters**
>
> - **scriptname** (`str`) – name of script to be retrieved
> - **scriptdata** (`str`) – script content

> > **Returns** response (`OK`, `NO`, `BYE`)

setactive(*scriptname*)

> Mark a script as the 'active' one.
>
> > **Parameters scriptname** (`str`) – name of script to be marked active
> >
> > **Returns** response (`OK`, `NO`, `BYE`)

starttls(*keyfile=None*, *certfile=None*)

> Puts the connection to the SIEVE server into TLS mode.
>
> If the server supports TLS, this will encrypt the rest of the SIEVE session. If you provide the keyfile and certfile parameters, the identity of the SIEVE server and client can be checked. This, however, depends on whether the socket module really checks the certificates.

Author Hartmut Goebel <h.goebel@crazy-compiler.com>

License Python Software Foundation License for the module, GPL v3 for *sieveshell* and test suite.

Homepage https://managesieve.readthedocs.io/

Download https://pypi.org/project/managesieve

Development https://gitlab.com/htgoebel/managesieve

CHAPTER 6

What is the *sieveshell*?

sieveshell is a command line tool for talking to a remote mail server. Sieve scripts allow users to filter incoming email on the mail server. Typically these servers are sealed so users cannot log into them, yet users must be able to update their scripts on them. This is what *sieveshell* is for.

One can

- list scripts on the server
- upload scripts to the server
- display scripts stored on the server and download or edit them
- delete scripts stored on the server
- activate and deactivate scripts

sieveshell is simple yet functional and useful for user who wish to manage sieve scripts without installing a fat GUI-based mail client.

What is managesieve?

The *managesieve* pure Python module is a ManageSieve ([RFC 5804](#)) client library for managing Sieve scripts on a mail server, more specific the *Sieve* server. For API details see the *Module Documentation*.

Python Module Index

m

managesieve, 13

Index