
Xymon client Documentation

Release 0.1

Romain Dartigues

October 12, 2016

1	Xymon client - library	3
1.1	Xymon	3
1.2	Xymon helpers	5
2	Examples	7
2.1	Main library	7
2.2	Helpers	7
3	Alternatives	9
3.1	skurfer/python-xymon	9
4	Indices and tables	11
	Python Module Index	13

Contents:

Xymon client - library

Contents

- *Xymon client - library*
 - *Xymon*
 - * *Examples*
 - *Xymon helpers*

1.1 Xymon

a minimalist Xymon client library in Python

1.1.1 Examples

Queries to one Xymon server:

```
>>> x = Xymon('xymonserver01.example.net')
>>> x.ping()
'xymond 4.3.18\n'
```

class `xymon_client.xymon.Xymon` (*server='localhost', port=1984, sender=None*)

Bases: `object`

thin library over Xymon protocol

Attempt to reflect `xymon(1)` ¹.

General concepts

Colors Valid colors are defined in `color_map`.

Times Called “duration”, “lifetime”. If given as a number followed by *s/m/h/d*, it is interpreted as being in seconds/minutes/hours/days respectively. For “until OK”, use -1.

client (*hostname, ostype, collectorid=None, hostclass=None*)

clientlog (*hostname, *sectionname*)

config (*filename*)

¹ <http://xymon.sourceforge.net/xymon/help/manpages/man1/xymon.1.html>

data (*hostname*, *dataname*, *text*='')

disable (*hostname*, *testname*='*', *duration*=-1, *text*='')

Parameters

- **hostname** (*str*) –
- **testname** (*str*) – use an asterisk (*) to disable all tests
- **duration** (*integer or str*) –
- **text** (*str*) –

download (*filename*)

drop (*hostname*, *testname*=None)

remove all data stored about this status

When removing an *hostname* as whole, it is assumed that you have already deleted the host from the `hosts.cfg` configuration file.

Parameters

- **hostname** (*str*) –
- **testname** (*str*) –

enable (*hostname*, *testname*)

re-enables a test that had been disabled

Parameters

- **hostname** (*str*) –
- **testname** (*str*) –

ghostlist ()

headline

hostinfo (*criteria*)

modify (*hostname*, *testname*, *color*, *source*, *cause*)

notes (*filename*)

notify (*hostname*, *testname*, *text*='')

ping ()

pullclient ()

query (*hostname*, *testname*)

query the Xymon server for the latest status reported for this particular test

Parameters

- **hostname** (*str*) –
- **testname** (*str*) –

rename (*old*, *new*, *hostname*=None)

schedule (*timestamp*=None, *command*=None)

schedule command for execution at a later time

E.g. used to schedule disabling of a host or service at sometime in the future. If no parameters are given, the currently scheduled tasks are listed in the response.

Parameters

- **command** (*str* or *None*) – a complete Xymon command such as the ones listed above or ‘cancel JOBID’ to cancel a previously scheduled command
- **timestamp** – the Unix epoch time when the command will be executed
- **timestamp** – int or *None*

status (*hostname*, *testname*, *color*, *text*='', *lifetime*=*None*, *group*=*None*)

Parameters

- **hostname** (*str*) –
- **testname** (*str*) –
- **color** (*str*) –
- **text** (*str*) –
- **lifetime** (*None* or *integer* or *str*) – defines how long this status is valid after being received by the Xymon server
- **group** (*str* or *None*) – direct alerts from the status to a specific group

usermsg (*identifier*)

xymondbboard (*criteria*=*None*, *fields*=*None*)

xymondlog (*hostname*, *testname*)

xymondxboard ()

xymondxlog (*hostname*, *testname*)

class `xymon_client.xymon.Xymons` (*servers*, *port*=1984, *sender*='build-4504730-project-64104-python-xymon-client', *thread*=*False*)

Bases: `object`

1.2 Xymon helpers

helpers for the Xymon Python-based library

class `xymon_client.helpers.Color`

Bases: `float`

describe a Xymon color (status)

name = *None*

textual representation of the color

class `xymon_client.helpers.Helper` (*xymon*, *hostname*=*None*, *testname*=*None*)

Bases: `object`

Examples:

```
# create an object with default hostname and default service
>>> x = Helper(Xymon('xymon.example.net'), 'www.intra.example.net', 'http')
# add a message
>>> x+= '&red something gone pear shaped\n'
# send the content of the current buffer with text added at the end
# override the global color of the message (from red to yellow)
# clear the buffer
```

```
>>> x.status('but it is not *that* bad', color=yellow)
# now the buffer has been cleared
# send another message, same hostname but different service name
>>> x.status('do not shoott the messenger!', service='logs')
```

color

current highest level color or clear if none found

get_colors (*text*, *default=None*)

extract all colors from text

Parameters

- **text** (*str*) –
- **default** (None or *Color*) –

Return type list of *Color* instances

r_color = <_sre.SRE_Pattern object>

extract status identifiers

status (*message=''*, ***kwargs*)

Examples

2.1 Main library

2.1.1 Queries against a single Xymon server

```
from xymon_client.xymon import Xymon
xymon = Xymon('xymonserver01.example.net')
response = xymon.ping()
# response sample: 'xymond 4.3.18\n'

xymon.query('xymonserver02.example.net', 'fping')
# response sample: 'green Message generated by xymonserver01.example.net at mercredi 12 octobre 2016, 14:00:00 CEST'

xymon.ghostlist()
# response sample: 'xyz.example.net/10.234.71.190/1476254463\nabc.example.net/10.234.71.189/1476254463'
```

2.1.2 Queries to multiple Xymon servers at once

```
from xymon_client.xymon import Xymons # note the subtle difference
xymon = Xymons(
    ['xymonserver01.example.net', 'xymonserver02.example.net'],
)
response = xymon.ping()
# response sample: {'xymonserver02.example.net:1984': 'xymond 4.3.18\n', 'xymonserver01.example.net:1984': 'xymond 4.3.18\n'}
```

2.2 Helpers

```
from xymon_client.xymon import Xymon
from xymon_client.helpers import *

# create an object with default hostname and default service
xc = Helper(Xymon('xymon.example.net'), 'www.intra.example.net', 'http')

# add a message
xc+= "I see something wrong, let's make it &yellow\n"
xc+= 'Oops, now &red, something gone pear shaped...\n'
```

```
# what is the current error level?
xc.color # should be red

# 1. send the content of the current buffer with text added at the end
# 2. override the global color of the message (from red to yellow)
# 3. clear the buffer
xc.status('but it is not *that* bad', color=yellow)

# now the buffer has been cleared

# send another message, same hostname but different service name
xc.status('do not shoot the messenger!', service='logs')
```

Alternatives

3.1 skurfer/python-xymon

Homepage <https://github.com/skurfer/python-xymon>

Pros:

- more mature
- Python 3 support
- use the Xymon `appfeed`

Cons:

- I didn't see a way to read responses from the server to passed commands
- only one server

Indices and tables

- `genindex`
- `modindex`
- `search`

X

`xymon_client.helpers`, 5
`xymon_client.xymon`, 3

C

client() (xymon_client.xymon.Xymon method), 3
clientlog() (xymon_client.xymon.Xymon method), 3
Color (class in xymon_client.helpers), 5
color (xymon_client.helpers.Helper attribute), 6
config() (xymon_client.xymon.Xymon method), 3

D

data() (xymon_client.xymon.Xymon method), 3
disable() (xymon_client.xymon.Xymon method), 4
download() (xymon_client.xymon.Xymon method), 4
drop() (xymon_client.xymon.Xymon method), 4

E

enable() (xymon_client.xymon.Xymon method), 4

G

get_colors() (xymon_client.helpers.Helper method), 6
ghostlist() (xymon_client.xymon.Xymon method), 4

H

headline (xymon_client.xymon.Xymon attribute), 4
Helper (class in xymon_client.helpers), 5
hostinfo() (xymon_client.xymon.Xymon method), 4

M

modify() (xymon_client.xymon.Xymon method), 4

N

name (xymon_client.helpers.Color attribute), 5
notes() (xymon_client.xymon.Xymon method), 4
notify() (xymon_client.xymon.Xymon method), 4

P

ping() (xymon_client.xymon.Xymon method), 4
pullclient() (xymon_client.xymon.Xymon method), 4

Q

query() (xymon_client.xymon.Xymon method), 4

R

r_color (xymon_client.helpers.Helper attribute), 6
rename() (xymon_client.xymon.Xymon method), 4

S

schedule() (xymon_client.xymon.Xymon method), 4
status() (xymon_client.helpers.Helper method), 6
status() (xymon_client.xymon.Xymon method), 5

U

usermsg() (xymon_client.xymon.Xymon method), 5

X

Xymon (class in xymon_client.xymon), 3
xymon_client.helpers (module), 5
xymon_client.xymon (module), 3
xymondboard() (xymon_client.xymon.Xymon method), 5
xymondlog() (xymon_client.xymon.Xymon method), 5
xymondxboard() (xymon_client.xymon.Xymon method), 5
xymondxlog() (xymon_client.xymon.Xymon method), 5
Xymons (class in xymon_client.xymon), 5