
ntfy Documentation

Release 1.3.2

Daniel Schep

March 24, 2016

1	Demo	3
2	Quickstart	5
3	Install	7
3.1	Extras	7
3.2	Shell integration	7
4	Configuring <code>ntfy</code>	9
4.1	Backends	9
4.2	Pushover - <code>pushover</code>	9
4.3	Pushbullet - <code>pushbullet</code>	10
4.4	XMPP - <code>xmpp</code>	10
4.5	Linux Desktop Notifications - <code>linux</code>	11
4.6	Windows Desktop Notifications - <code>win32</code>	11
4.7	Mac OS X Notification Center - <code>darwin</code>	11
4.8	Other options	11
4.9	Backends ToDo	11
5	Testing	13
6	Contributors	15
7	Indices and tables	17

`ntfy` is a utility for sending notifications, on demand and when commands finish. It supports many delivery mechanisms, both local and remote.

Demo

Quickstart

```
$ sudo pip install ntfy
$ ntfy send test
# send a notification when the command `sleep 10` finishes
# this send the message '"sleep 10" succeeded in 0:10 minutes'
$ ntfy done sleep 10
$ ntfy -b pushover -o user_key t0k3n send 'Pushover test!'
$ ntfy -t 'ntfy' send "Here's a custom notification title!"
$ echo -e 'backends: ["pushover"]\npushover: {"user_token": "t0k3n"}' > ~/.config/ntfy/ntfy.yml
$ ntfy send "Pushover via config file!"
$ ntfy done --pid 6379 # PID extra
$ ntfy send ":tada: ntfy supports emoji! :100:" # emoji extra
```

Install

It is suggested to globally (as indicated in quickstart above, without a virtualenv) install `ntfy`. It *can* be installed in a virtualenv, with caveats. Linux notifications requires `--system-site-packages` to be used and OS X notifications don't work at all.

3.1 Extras

`ntfy` supports emoji shortcodes, support can be installed by installing `ntfy[emoji]`. XMPP support requires SleekXMPP which can be installed by installing `ntfy[xmpp]`. `ntfy done` can watch existing processes by their PID if you install `ntfy[pid]`. See the quickstart for usage examples.

3.2 Shell integration

`ntfy` has support for **automatically** sending notifications when long running commands finish in `bash` and `zsh`. In `bash` it emulates `zsh`'s `preexec` and `precmd` functionality with [rcaloras/bash-preexec](#). To enable it add the following to your `.bashrc` or `.zshrc`:

```
eval "$(ntfy shell-integration) "
```

By default it will only send notifications for commands lasting longer than 10 seconds. This can be configured with the `AUTO_NTFY_DONE_TIMEOUT` environment variable.

To avoid unnecessary notifications when running interactive programs programs listed in `AUTO_NTFY_DONE_IGNORE` don't generate notifications. for example:

```
export AUTO_NTFY_DONE_IGNORE="vim screen meld"
```

Configuring `ntfy`

`ntfy` is configured with a YAML file stored at `~/.config/ntfy/ntfy.yml`

4.1 Backends

The `backends` key specifies what backends to use by default. Each backend has its own configuration, stored in a key of its own name. For example:

```
---
backends:
  - pushover
  - linux
  - xmpp
pushover:
  user_key: hunter2
pushbullet:
  access_token: hunter2
xmpp:
  jid: "user@gmail.com"
  password: "xxxx"
  mtype: "chat"
  recipient: "me@jit.si"
```

If you want multiple configs for the same backend type, you can specify any name and then specify the backend with a backend key. For example:

```
---
pushover:
  user_key: hunter2
cellphone:
  backend: pushover
  user_key: hunter2
```

See the backends below for available backends and options.

4.2 Pushover - `pushover`

Required parameters:

- `user_key`

Optional parameters:

- `sound`
- `priority`
- `expire`
- `retry`
- `callback`
- `access_token` - use your own application token
- `device` - target a device, if omitted, notification is sent to all devices
- `url`
- `url_title`
- `html`

4.3 Pushbullet - `pushbullet`

Required parameter:

- `access_token` - Your Pushbullet access token, created at <https://www.pushbullet.com/#settings/account>

Optional parameters:

- `device_iden` - a device identifier, if omitted, notification is sent to all devices
- `email` - send notification to pushbullet user with the specified email or send an email if they aren't a pushbullet user

4.4 XMPP - `xmpp`

Requires parameters:

- `jid`
- `password`
- `recipient`

Optional parameters

- `hostname` (if not from `jid`)
- `port`
- `path_to_certs`
- `mtype` ('chat' required for Google Hangouts)

To verify the SSL certificates offered by a server: `path_to_certs = "path/to/ca/cert"`

Without `dnspython` library installed, you will need to specify the server hostname if it doesn't match the `jid`.

For example, to use Google Talk you would need to use: `hostname = 'talk.google.com'`

Specify port if other than 5222. NOTE: Ignored without specified hostname

4.5 Linux Desktop Notifications - `linux`

Works via `dbus`, works with most DEs like Gnome, KDE, XFCE and with `libnotify`.

4.6 Windows Desktop Notifications - `win32`

Uses `pywin32`.

4.7 Mac OS X Notification Center - `darwin`

Requires `ntfy` to be installed globally (not in a `virtualenv`).

4.8 Other options

Title is configurable with the `title` key in the config. Example:

```
---
title: Customized Title
```

4.9 Backends `ToDo`

- `Prowl`
- `Airgram`
- `Pushjet`
- `Pushalot`
- `Boxcar`
- `Instapush`

Testing

```
python setup.py test
```

Contributors

- [dschep](#) - Maintainer & Lead Developer
- [danryder](#) - XMPP Backend & emoji support

Indices and tables

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