
ntfy Documentation

Release 2.3.0

Daniel Schep

September 14, 2016

1	Quickstart	3
2	Install	5
2.1	Shell integration	5
2.2	Extras	5
3	Configuring <code>ntfy</code>	7
3.1	Backends	7
3.2	Pushover - <code>pushover</code>	8
3.3	Pushbullet - <code>pushbullet</code>	8
3.4	Simplepush - <code>simplepush</code>	8
3.5	XMPP - <code>xmpp</code>	8
3.6	Telegram - <code>telegram</code>	9
3.7	Pushjet - <code>pushjet</code>	9
3.8	Notifico - <code>notifico</code>	9
3.9	Slack - <code>Slack</code>	9
3.10	Instapush - <code>insta</code>	10
3.11	Prowl - <code>prowl</code>	10
3.12	Linux Desktop Notifications - <code>linux</code>	10
3.13	Windows Desktop Notifications - <code>win32</code>	10
3.14	Mac OS X Notification Center - <code>darwin</code>	11
3.15	System log - <code>systemlog</code>	11
3.16	Other options	12
3.17	Backends ToDo	12
4	Testing	13
5	Contributors	15
6	Indices and tables	17

`ntfy` brings notification to your shell. It can automatically provide desktop notifications when long running commands finish or it can send push notifications to your phone when a specific command finishes. Confused? This video demonstrates some of this functionality:

Quickstart

```
$ sudo pip install ntfy
$ ntfy send test
# send a notification when the command `sleep 10` finishes
# this sends 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_key": "t0k3n"}' > ~/.ntfy.yml
$ ntfy send "Pushover via config file!"
$ ntfy done --pid 6379 # PID extra
$ ntfy send ":tada: ntfy supports emoji! :100:" # emoji extra
# Enable shell integration
$ echo 'eval "$(ntfy shell-integration)"' >> ~/.bashrc
```

Install

The install technique in the quickstart is the suggested method of installation. It can be installed in a virtualenv, but with some caveats: Linux notifications require `--system-site-packages` for the virtualenv and OS X notifications don't work at all.

2.1 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 and if the terminal is focused. Terminal focus works on X11(Linux) and with Terminal.app and iTerm2 on MacOS. Both options can be configured via the `--longer-than` and `--foreground-too` options.

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"
```

2.2 Extras

`ntfy` has a few features that require extra dependencies.

- `ntfy done -p $PID` requires installing as `ntfy[pid]`
- XMPP requires installing as `ntfy[xmpp]`
- Telegram requires installing as `ntfy[telegram]`
- Instapush requires installing as `ntfy[instapush]`

emojis, Telegram and xmpp are extras.

Configuring `ntfy`

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

3.1 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
  - simplepush
  - slack
  - linux
  - xmpp
pushover:
  user_key: hunter2
pushbullet:
  access_token: hunter2
simplepush:
  key: hunter2
slack:
  token: slacktoken
  channel: "#slackchannel"
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.

3.2 Pushover - pushover

Required parameters:

- `user_key`

Optional parameters:

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

3.3 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

3.4 Simplepush - simplepush

Required parameter:

- `key` - Your Simplepush key, created by installing the Android App (no registration required) at <https://simplepush.io>

3.5 XMPP - xmpp

Requires parameters:

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

Optional parameters

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

Require extras, install like this: `pip install ntfy[xmpp]`.

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

3.6 Telegram - `telegram`

Require extras, install like this: `pip install ntfy[telegram]`.

Requires `ntfy` to be installed as `ntfy[telegram]`. This backend is configured the first time you will try to use it: `ntfy -b telegram send "Telegram configured for ntfy"`.

3.7 Pushjet - `pushjet`

Required parameter:

- `secret` - The Pushjet service secret token, created with <http://docs.pushjet.io/docs/creating-a-new-service>

Optional parameters:

- `endpoint` - custom Pushjet API endpoint (defaults to <https://api.pushjet.io>)
- `level` - The importance level from 1(low) to 5(high)
- `link`

3.8 Notifico - `notifico`

Required parameter:

- `webhook` - The webhook link, created at <https://n.tkte.ch/> (choose Plain Text service when creating the webhook)

3.9 Slack - `slack`

Require extras, install like this: `pip install ntfy[slack]`.

Required parameter:

- `token` - The Slack service secret token, created in <https://api.slack.com/web#authentication>

- `#channel` - The Slack channel to send notification. If you use the `#` symbol the message is send to a Slack channel and if you use the `@` symbol the message is send to a Slack user.

3.10 Instapush - `insta`

Require extras, install like this `pip install ntfy[instapush]`.

Instapush does not support notification title. It sends template-driven notifications, so you have to setup you events on the dashboard first. The backend is called `insta` due to homonymy with the `instapush` python wrapper

Required parameters:

- `appid` - The application id
- `secret` - The application secret
- `event_name` - The instapush event to be used
- `trackers` - The array of trakers to use

Note on trackers: Trackers are placeholders for events (a sort of notification template). If you defined more than one tracker in your event you'll have to provide more messages. At the moment, the only way to do so is to separate each message with a colon (`:`) character. You can also escape the separator character: Example:

```
ntfy -b insta send "message1:message2"
ntfy -b insta send "message1:message2\:with\:colons"
```

3.11 Prowl - `prowl`

Optional parameters:

- `api_key`
- `provider_key`
- `priority`
- `url`

3.12 Linux Desktop Notifications - `linux`

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

The following dependencies should be installed.

```
$ sudo apt-get install libdbus-glib-1-dev libdbus-1-dev
$ pip install --user dbus-python
```

You will need to install some font that supports emojis (in Debian *fonts-symbola* or Gentoo *media-fonts/symbola*).

3.13 Windows Desktop Notifications - `win32`

Uses `pywin32`.

3.14 Mac OS X Notification Center - `darwin`

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

3.15 System log - `systemlog`

Uses the `syslog` core Python module, which is not available on Windows platforms.

Optional parameters:

- `prio` - Syslog priority level. Default is `ALERT`. Possible values are:
 - `EMERG`
 - `ALERT`
 - `CRIT`
 - `ERR`
 - `WARNING`
 - `NOTICE`
 - `INFO`
 - `DEBUG`
- `facility` - Syslog facility. Default is `LOCAL5`. Possible values are:
 - `KERN`
 - `USER`
 - `MAIL`
 - `DAEMON`
 - `AUTH`
 - `LPR`
 - `NEWS`
 - `UUCP`
 - `CRON`
 - `SYSLOG`
 - `LOCAL0`
 - `LOCAL1`
 - `LOCAL2`
 - `LOCAL3`
 - `LOCAL4`
 - `LOCAL5`
 - `LOCAL6`
 - `LOCAL7`

- `fmt` - Format of the message to be sent to the system logger. The title and the message are specified using the following placeholders:

- `{title}`
- `{message}`

Default is `[{title}] {message}`.

3.16 Other options

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

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

3.17 Backends ToDo

- Airgram
- Pushalot
- Boxcar

Testing

```
python setup.py test
```

Contributors

- [dschep](#) - Maintainer & Lead Developer
- [danryder](#) - XMPP Backend & emoji support
- [oz123](#) - Linux desktop notification improvements
- [schwert](#) - PushJet support
- [rahiel](#) - Telegram support
- [tymm](#) - Simplepush support
- [jungle-boogie](#) - Documentation updates
- [tjbenator](#) - Advanced Pushover options
- [mobiusklein](#) - Win32 Bugfix
- [rcaloras](#) - Creator of *bash-prexec*, without which there wouldn't be bash shell integration for *ntfy*
- [eightnoteight](#) - Notifico support
- [juanpabloaj](#) - Slack support
- [giuseongit](#) - Instapush support
- [jlesage](#) - Systemlog support
- [sambrightman](#) - Prowl support

Indices and tables

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