
scdlbot

Release 0.12.2

gpchelkin

Jan 14, 2020

CONTENTS

1 Bot Usage	3
2 Development	5
3 Contents	9
4 Indices and tables	21

Telegram Bot for downloading MP3 rips of tracks/sets from SoundCloud, Bandcamp, YouTube with tags and artwork.

- Free software: [MIT License](#)
- Documentation: <https://scdlbot.readthedocs.io>

Contents

- *Music Downloader Telegram Bot aka scdlbot*
 - *Bot Usage*
 - *Development*
 - *Contents*
 - *Indices and tables*

BOT USAGE

Send `/start` or `/help` command to bot or refer directly to the [help message](#).

Please report all bugs and issues and suggest your improvements to [issues](#).

1.1 Supported sites and mainly used packages

- Telegram Bot API: `python-telegram-bot`
- SoundCloud: `sctl`
- Bandcamp: `bandcamp-dl`
- YouTube, Mixcloud, everything else from the list: `youtube-dl`

DEVELOPMENT

2.1 Installation

2.1.1 Requirements

Those should be available in your PATH:

- Python 3.6+ (pyenv recommended)
- FFmpeg 4.0+ if not running on Heroku (fresh builds for Windows and Linux are recommended)
- Heroku CLI is recommended

2.1.2 Install / Update stable from PyPI (recommended)

```
pip3 install scdlbot
```

2.1.3 Install / Update unstable from Git source

```
git clone https://github.com/gpchelkin/scdlbot.git
cd scdlbot
pip3 install --requirement requirements.txt

# Update:
git pull
pip3 install --requirement requirements.txt

# System-wide install link to current sources, recommended:
python3 setup.py develop

# System-wide install copy of current sources, not recommended:
python3 setup.py install
```

2.2 Configuration

Download or copy config file sample and set up config environment variables in it:

```
# PyPI-installed: download sample config:
curl -o .env https://raw.githubusercontent.com/gpchelkin/scdlbot/master/.env.sample

# Git source-installed: copy sample config:
cp .env.sample .env

# Use your favourite editor:
nano .env
```

2.2.1 Required

- **TG_BOT_TOKEN:** Telegram Bot API Token, [obtain here](#)

2.2.2 Optional

- **SC_AUTH_TOKEN:** SoundCloud Auth Token, [obtain here](#)
- **STORE_CHAT_ID:** Chat ID for storing audios of inline mode
- **USE_WEBHOOK:** use webhook for bot updates: 1, use polling (default): 0, [more info](#)
- **APP_URL:** app URL like `https://<appname>.herokuapp.com/`, required for webhook
- **PORT:** port for webhook to listen to; Heroku sets this automatically for web dynos
- **BOTAN_TOKEN:** [Botan.io token](#)
- **NO_FLOOD_CHAT_IDS:** Comma-separated chat IDs with no replying and caption spam
- **BIN_PATH:** Custom directory where `scdl` and `bandcamp-dl` binaries are available, e.g. `~/pyenv/shims/` if you use `pyenv`, default: empty (binaries are available in `PATH`)
- **DL_DIR:** Parent directory for downloads directories, default: `/tmp/scdlbot`
- **DL_TIMEOUT:** Download timeout in seconds, stop downloading if it takes longer than allowed, default: 300
- **MAX_CONVERT_FILE_SIZE:** Don't try to split and send files over this number of bytes, default: 8000000
- **SYSLOG_ADDRESS:** Syslog server, for example `logsX.papertrailapp.com:ABCDE`
- **SYSLOG_DEBUG:** Enable verbose debug logging: 1
- **HOSTNAME:** Hostname to show up in Syslog messages
- **GOOGL_API_KEY:** [Goo.gl URL shortener API key](#)

Webhooks: These three links should help. In NGINX use `TOKEN1` as `TG_BOT_TOKEN` without “:” symbol, and port in `proxy_pass` according to `PORT` environment variable.

- https://nginx.org/en/linux_packages.html#mainline
- <https://github.com/python-telegram-bot/python-telegram-bot/wiki/Webhooks#using-nginx-with-one-domainport-for-all-bots>
- <https://certbot.eff.org/#ubuntuxenial-nginx>

2.2.3 Telegram Bot Settings

Send the commands from respective filenames in `telegram_settings` dir to `@BotFather`, choose your bot and copy corresponding values in order to use the bot conveniently. Also disable privacy mode if you want to.

2.3 Running Locally or at Dedicated Server

2.3.1 Using Heroku Local (preferred)

You will need `Heroku CLI` installed.

```
# PyPI-installed: download Procfile:
curl -O https://raw.githubusercontent.com/gpchelkin/scdlbot/master/Procfile

# For long polling:
heroku local worker
# For webhook:
heroku local web
```

2.3.2 Using Python only

```
# For PyPI or Git source system-wide installs:
export $(cat .env | xargs)
scdlbot

# or in one line:
env $(cat .env | xargs) scdlbot

# For non-installed Git source repository directory:
export $(cat .env | xargs)
python -m scdlbot
# or in one line:
env $(cat .env | xargs) python -m scdlbot
```

2.4 Deploying to Heroku

Register on Heroku, press the button above and configure variables for deploying. When app is deployed you **must** set only one dyno working on “Resources” tab in your app settings depending on [which way of getting updates](#) you have chosen and set in config variables: `worker` for polling or `web` for webhook.

2.4.1 Manually

You can do the same as the button above but using **Heroku CLI**, not much of a fun. Assuming you are in scdlbot repository directory:

```
heroku login
# Create app with Python3 buildpack and set it for upcoming builds:
heroku create --buildpack heroku/python
heroku buildpacks:set heroku/python
# Add FFmpeg buildpack needed for youtube-dl:
heroku buildpacks:add --index 1 https://github.com/laddhadhiraj/heroku-buildpack-
↪ffmpeg.git --app scdlbot
# Deploy app to Heroku:
git push heroku master
# Set config vars automatically from your .env file
heroku plugins:install heroku-config
heroku config:push
# Or set them one by one:
heroku config:set TG_BOT_TOKEN="<TG_BOT_TOKEN>" STORE_CHAT_ID="<STORE_CHAT_ID>" ...
```

If you use webhook, start web dyno and stop worker dyno:

```
heroku ps:scale web=1 worker=0
heroku ps:stop worker
```

If you use polling, start worker dyno and stop web dyno:

```
heroku ps:scale worker=1 web=0
heroku ps:stop web
```

Some useful commands:

```
# Attach to logs:
heroku logs -t
# Test run ffmpeg
heroku run "ffmpeg -version"
```

2.5 Deploying to Dokku

Use Dokku (your own Heroku) installed on your own server. App is tested and fully ready for deployment with polling (no webhook yet). <https://github.com/dokku/dokku-letsencrypt>

```
export DOKKU=<your_dokku_server>
scp .env $DOKKU:~
ssh $DOKKU
  export DOKKU=<your_dokku_server>
  dokku apps:create scdlbot
  dokku certs:generate scdlbot scdlbot.$DOKKU
  dokku config:set scdlbot $(cat .env | xargs)
  logout
git remote add dokku dokku@$DOKKU:scdlbot
git push dokku master
ssh $DOKKU
  dokku ps:scale scdlbot worker=1 web=0
  dokku ps:restart scdlbot
```

CONTENTS

3.1 How to contribute

3.1.1 Dependencies

We use `poetry` to manage the dependencies.

To install them you would need to run `install` command:

```
poetry install
```

To activate your `virtualenv` run `poetry shell`.

3.1.2 One magic command

Run `make test` to run everything we have!

3.1.3 Tests

We use `pytest` and `flake8` for quality control. We also use `wemake_python_styleguide` to enforce the code quality.

To run all tests:

```
pytest
```

To run linting:

```
flake8 .
```

Keep in mind: default virtual environment folder excluded by `flake8` style checking is `.venv`. If you want to customize this parameter, you should do this in `setup.cfg`. These steps are mandatory during the CI.

3.1.4 Type checks

We use `mypy` to run type checks on our code. To use it:

```
mypy scdlbot tests/**/*.py
```

This step is mandatory during the CI.

3.1.5 Submitting your code

We use `trunk based` development (we also sometimes call it `wemake-git-flow`).

What the point of this method?

1. We use `protected master` branch, so the only way to push your code is via pull request
2. We use `issue branches`: to implement a new feature or to fix a bug create a new branch named `issue- $\$$ TASKNUMBER`
3. Then create a pull request to `master` branch
4. We use `git tags` to make releases, so we can track what has changed since the latest release

So, this way we achieve an easy and scalable development process which frees us from merging hell and long-living branches.

In this method, the latest version of the app is always in the `master` branch.

Before submitting

Before submitting your code please do the following steps:

1. Run `pytest` to make sure everything was working before
2. Add any changes you want
3. Add tests for the new changes
4. Edit documentation if you have changed something significant
5. Update `CHANGELOG.md` with a quick summary of your changes
6. Run `pytest` again to make sure it is still working
7. Run `mypy` to ensure that types are correct
8. Run `flake8` to ensure that style is correct
9. Run `doc8` to ensure that docs are correct

3.1.6 Other help

You can contribute by spreading a word about this library. It would also be a huge contribution to write a short article on how you are using this project. You can also share your best practices with us.

3.2 Credits

3.2.1 Development Lead

- George Pchelkin @gpchelkin <george@pchelk.in>

3.2.2 Contributors

- Leonid Runyshkin @leovp
- Vadim Larionov @vadimlarionov <larionov.vadim@mail.ru>

3.3 Version history

3.3.1 0.12.2 (2020-01-14)

- refined source IP handling

3.3.2 0.12.1 (2020-01-13)

- hotfix for env parsing

3.3.3 0.12.0 (2020-01-13)

- add support for source IPs randomization for downloading from youtube

3.3.4 0.11.0 (2020-01-12)

- add support for cookie file for youtube-dl - it helps with downloading yandex.music from different countries

3.3.5 0.10.0 (2020-01-07)

- use ffmpeg-python for converting instead of RAM-heavy pydub, remove pydub requirement
- update docs and requirements
- many cosmetic and CI fixes

3.3.6 0.9.27 (2019-12-01)

- return timestamp to scdl using some hacks
- don't treat one track's fail as playlist's fail for scdl

3.3.7 0.9.26 (2019-11-30)

- upgrade requirements to use scdl from git
- various deployment and CI/CD improvements

3.3.8 0.9.21-25 (2019-11-04)

- catch scdl 'not streamable' error and add some explanation about youtube '429 too many requests' error
- upgrade requirements
- add python 3.8 support
- various deployment and CI/CD improvements

3.3.9 0.9.20 (2019-10-11)

- upgrade requirements
- add python 3.7 support
- remove pyup
- fix travis pypi access

3.3.10 0.9.19 (2019-02-09)

- edited help message
- upgrade requirements

3.3.11 0.9.18 (2018-09-01)

- upgrade requirements

3.3.12 0.9.17 (2018-05-12)

- upgrade requirements

3.3.13 0.9.16 (2018-03-08)

- markdown in audio captions HOTFIX

3.3.14 0.9.15 (2018-03-03)

- markdown in audio captions

3.3.15 0.9.14 (2018-02-11)

- source is now captions
- more precise help and settings

3.3.16 0.9.13 (2018-02-11)

- short original links in captions
- wait a bit added

3.3.17 0.9.12 (2018-02-11)

- flood is now source
- wait a beet/beat random
- provide youtube video titles in caption
- 200 char caption text wrap

3.3.18 0.9.11 (2018-02-10)

- send tags and time length to Telegram
- extract-artist feature
- wait a beet
- requirements updated

3.3.19 0.9.10 (2017-12-14)

- convert non-mp3 files to mp3
- don't apologize for pictures
- don't allow sc/you links
- some renaming, move get_direct_urls to utils

3.3.20 0.9.9 (2017-12-12)

- serve audios with nginx and send URLs instead of files <https://core.telegram.org/bots/api#sending-files>

3.3.21 0.9.8 (2017-12-05)

- b0tan logging improvement
- requirements updated

3.3.22 0.9.7.2 (2017-12-05)

- console debug logging

3.3.23 0.9.7.1 (2017-12-05)

- easter egg added

3.3.24 0.9.7 (2017-12-04)

- new telegram logging
- only admins can change settings
- close button in settings

3.3.25 0.9.6 (2017-12-03)

- separate file for utils, botan track refactored
- parse message for markdown text links too
- temporary disable SSL verification for botan (cert expired)

3.3.26 0.9.5.2 (2017-12-03)

- hot fix

3.3.27 0.9.5.1 (2017-12-02)

- fix logging

3.3.28 0.9.5 (2017-12-02)

- settings command
- persistent storage for settings
- refactoring

3.3.29 0.9.4 (2017-11-27)

- exception-driven behaviour
- clutter is now flood
- better alerting

3.3.30 0.9.3 (2017-11-22)

- don't spam on second try
- better logging config for different destinations

3.3.31 0.9.2 (2017-11-21)

- more error checking and responding with errors
- much much much polishing and refactoring

3.3.32 0.9.1 (2017-11-20)

- more error checking and responding
- limit youtube-dl time to download
- avoid live downloads
- logging refactor and refinement
- help updates

3.3.33 0.9.0 (2017-11-20)

- return of inline mode as fast download (link is sent to telegram servers for download)
- refactor and refinement
- help updates
- add some spam captions :)

3.3.34 0.8.3 (2017-11-19)

- gc.collect() according to <https://github.com/jiaaro/pydub/issues/89#issuecomment-75245610>

3.3.35 0.8.2 (2017-11-19)

- cool refinements in logging
- store urls, so button response is faster now

3.3.36 0.8.1 (2017-11-19)

- some logging fixes

3.3.37 0.8.0 (2017-11-19)

- many fixes and workarounds
- alerting & logging

3.3.38 0.7.10 (2017-11-05)

- botanio fix - send user id, not chat id

3.3.39 0.7.9 (2017-11-05)

- botanio fix
- tmpreaper config sample
- clutter fix

3.3.40 0.7.8 (2017-11-04)

- botanio
- maintenance

3.3.41 0.7.7 (2017-09-11)

- maintenance

3.3.42 0.7.6 (2017-09-11)

- SYSLOG_DEBUG env var to disable logging of full messages
- maintenance
- Logentries support

3.3.43 0.7.5.1 (2017-09-03)

- YouTube number remove

3.3.44 0.7.5 (2017-09-03)

- maintenance

3.3.45 0.7.4 (2017-08-03)

- msg_store fixes

3.3.46 0.7.3 (2017-07-20)

- orig_msg_id hotfix and don't send chat action on every link

3.3.47 0.7.2 (2017-07-19)

- Updated requirements

3.3.48 0.7.1 (2017-07-05)

- Hotfix

3.3.49 0.7.0 (2017-07-05)

- Travis CI, tests and docs from cookiecutter

3.3.50 0.6.3 (2017-07-04)

- Back to bandcamp-dl and scdl and download timeouts

3.3.51 0.6.2 (2017-07-04)

- Help message in groups now redirects to PM

3.3.52 0.6.1 (2017-07-03)

- Async run of download/send command
- Link command

3.3.53 0.6.0 (2017-07-02)

- Added text files to sdist
- Bandcamp and SoundCloud-widgets is now downloaded with youtube-dl
- Supported parsing widgets from pages
- Refactor

3.3.54 0.5.1 (2017-07-02)

- New clutter command
- Help refinements
- Some fixes

3.3.55 0.5.0 (2017-06-28)

- Big refactor to class-based
- Syslog support
- Some fixes

3.3.56 0.4.0 (2017-06-15)

- Console script!
- Setup script version improvements
- Ask in groups only, download immediately in private
- Bandcamp: Download links without 'bandcamp' for /dl
- Move TODOs to issues
- Button to destroy music from the Internet

3.3.57 0.3.1 (2017-06-12)

- Markdown to reStructuredText
- Copy tags to parts

3.3.58 0.3.0 (2017-06-10)

- YouTube playlists support
- Split audio by 50 MB size for sending
- Disable privacy mode and ask for download

3.3.59 0.2.0 (2017-06-06)

- Webhooks and async

3.3.60 0.1.0 (2017-06-04)

- First usable and stable version.

INDICES AND TABLES

- genindex
- modindex
- search