Labelord Documentation

Release 0.5

Michal Klement

Dec 02, 2017

Contents:

So you have tons of GitHub repositories and you want all of them to have your cool custom labels but managing labels is hell, right? Not with *Labelord*! Setup necessary configuration and enjoy synchronized labels across all your repositories.

CHAPTER 1

Installation

1. **via test.pypi** `python -m pip install --extra-index-url https://test.pypi.org/pypi labelord_klememi1`

2. **manually**

   - download the package from GitHub repository

   - unpack it

   - run `python setup.py install`

Configuration

- *Access token*
- *Webhook*
- *Configuration file*

CHAPTER 3

Usage

- *Command-line*
- *Web application*

Examples

Examples

CHAPTER 5

API reference

- *CLI module*
- *GitHub module*
- *Web module*
- *Helpers module*

License

License text

6.1 Configuration

6.1.1 Access token

Labelord is working with *GitHub* so valid access token needs to be provided for the application to work properly. You can generate a new one here. Make sure that **repo** scope is selected if you want to also manage your private repositories.

You can provide your token to application by 3 ways (sorted from lowest priority):

- Directive in configuration file, see *Configuration file* below
- Via environmental variable *GITHUB_TOKEN*
- *-t / –token* option of command-line application

Do not forget that no one should know your personal token so never make it public!

6.1.2 Webhook

Labelord offers an option to handle changes on labels of your repositories automatically. In order to get this done you need to set up a webhook. From your template repository go to **Settings > Webhooks** and create a new one with these settings:

- Payload URL: IP address, where Labelord is running. If you don't own public IP, you can use free hosting services like pythonanywhere.com. Default port is **5000**.
- Content type: **application/json**
- Secret: Your secret passphrase, which you also need to set in Labelord config. It's used to check that incoming request is from real GitHub.

> • Trigger events: **label**

Copy your webhook secret to configuration file, as you can see below.

6.1.3 Configuration file

By default configuration file **config.cfg** is located in package root directory. It's structure looks like this:

```
[github]
token = <your_personal_token>
webhook_secret = <your_webhook_secret>

; Repositories you wish to keep in sync
[repos]
owner/repo1 = on
owner/repo2 = off

; Template labels names and color in hex
[labels]
label1 = ff0000
label2 = 00ff00
label3 = 0000ff

; Repository used as a template for labels. Has higher precedence than [labels]
[others]
template-repo = repoowner/labelsrepo
```

6.2 Usage

6.2.1 Command-line

CLI part of the application provides one-time actions for listing labels, repositories and managing labels across your repositories. You can run CLI application with:

```
labelord [OPTIONS] COMMAND [ARGS]
```

Options

-c, --config PATH	Path of the configuration file. Default **./config.cfg**
-t, --token STRING	GitHub access token.
--version	Shows Labelord version currently installed.
--help	Shows help menu.

Commands

list_repos

Prints all repositories which can be processed with provided token. Each repository on single line in format *owner/repo*.

list_labels <repository>

Prints all current labels from the repository. Each one on single line in format **#XXXXXX name**, where *#XXXXXX* is label color in hex format.

run <mode>

Runs labels processing in one of the modes desribed below.

Modes

update Labels are added or updated from the template.

replace Labels are completely overriden by the template ones, that means labels that are in the repository but not in template are deleted.

Options

-r, --template-repo REPOSITORY Defines repository that should be used as a template of labels.

-a, --all-repos Use all repositories for processing (can be obtained with `list_repos`).

-d, --dry-run Doesn't make any changes to repositories, just prints actions.

-v, --verbose Turns on verbose mode, printing out all actions done.

-q, --quiet Turns on quiet mode, nothing will be printed.

Logging

In **verbose** mode, every action done is printed on single line beginning with two *Tags*. Last line is summary indicating number of successful operations done or number of errors if any.

If **quiet** mode is chosen, nothing is printed, success of operations can be checked from return value (0 successful, 10 errors occured).

If neither is chosen only summary line is printed after actions are done.

Tags

[ADD] Action of adding a label.

[UPD] Action of updating a label.

[DEL] Action of deleting a label.

[LBL] Action of reading a label from repository (if an error occured while doing this).

[DRY] Action done successfully in *dry-run* mode.

[SUC] Action done successfully on GitHub.

[ERR] Action raised an error.

run_server

Starts web application locally.

Options

-h, --host IP	Hostname specification, default **127.0.0.1**.
-p, --port PORT	Port specification, default **5000**.
-d, --debug	Flag turns on debug mode.

6.2.2 Web application

After you have started application locally or deployed remotely you can send *GET* or *POST* requests.

GET

Returns informations about the application and list of repositories configured for processing.

POST

Responds on requests from GitHub webhook propagating any change on label from one repository to all others.

6.3 Examples

6.3.1 Logging a successful action

```
>>> log_suc('ADD', 'SUC', 'labelord/repo1', 'Fix', '#afc100')
[ADD][SUC] labelord/repo1; Fix; #afc100
```

6.3.2 Logging a successful dry action

```
>>> log_suc('UPD', 'DRY', 'labelord/repo2', 'Todo', '#faa234')
[UPD][DRY] labelord/repo2; Todo; #faa234
```

6.3.3 Logging an unsuccesful action

```
>>> log_err('DEL', 'labelord/repo3', 'C00L', '#435123', 400, 'BAD REQUEST')
[DEL][ERR] labelord/repo3; C00L; #435123; 400 – BAD REQUEST
```

6.4 API

6.4.1 CLI module

6.4.2 GitHub module

6.4.3 Web module

6.4.4 Helpers module

6.5 License text

6.5.1 MIT License

Copyright (c) 2017 Michal Klement

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.