
lablabel Documentation

Release 2018.1

Alan Christie

Nov 24, 2018

Contents:

1	Installation	3
2	The Label Definition File	5
2.1	List Labels	6
3	Running lablabel	7
4	Limitations	9
4.1	Known Limitations	9

`lablabel` is a label utility for GitLab projects. Given a simple markdown text file of label definitions this utility sets up (or updates) a GitLab repository's labels and lists. The text file accommodates the definition of labels, colours, priorities and descriptions. It interprets any black labels as lists, creating a set of lists for your boards.

To use the utility you will need: -

- A text file of label definitions
- A GitLab project
- A GitLab *Personal Access Code* (or [PAT](#))

CHAPTER 1

Installation

lablabel can be found on [GitLab](#) and is published on [PyPI](#) and can be installed from there:

```
pip install matildapeak-lablabel
```


CHAPTER 2

The Label Definition File

`lablabel` creates labels and lists based on the content of a `markdown` file whose name is `GITLAB-LABELS.md` that is located in the current working directory or execution directory.

It's a standard markdown file but it is expected to contain the following sub-heading, which starts a block that defines the project labels:

```
## Common labels
```

Labels are interpreted between this sub-heading and the next (or the end of the file). Each label is defined in a sub-subheading (###) and is a list of bullet-points defining the Colour, Priority and Description of each label.

Here's an example of a block that contains definitions for the labels `Blocker` and `Documentation`:

```
## Common labels

### Blocker

- Colour: #FF0000
- Priority: 0
- Description: Urgent faults presenting significant problems that need
               to be addressed at the highest priority. They either
               block features or break production functionality.

### Documentation

- Colour: #428BCA
- Priority: 4
- Description: An issue relating to documentation, either a fault
               with an existing document or the need for a new
               document

## Some other stuff
```

- The colour should be a 6-digit hex value preceded with #
- The Priority should be a number

- The `Description` is text and it may span more than one line (as shown)

2.1 List Labels

Black labels (those with a colour value of `#000000`) are used to create lists and are expected to define a `position` rather than a `priority`.

CHAPTER 3

Running lablabel

You can get help with the utility using `--help`:

```
lablabel --help
```

To setup the `proj` repository for the user `blob`, with the personal access code `1234`, run the following in a directory that contains your `GITLAB-LABELS.md` file:

```
lablabel blob proj 1234 set
```

And update the project with:

```
lablabel blob proj 1234 update
```

It is safe to run a `set` or `update` on a project that already contains the defined labels.

A simple test mode can be used to run the utility on a file without writing the label definitions to Git:

```
lablabel blob proj 1234 --test
```

A good working practice is to add a `GITLAB-LABELS.md` file to the root of your project and run `lablabel` from there. You then have a permanent record of the label definitions and, if you need to add or adjust your labels in the future, edit the file and just re-run `lablabel` (see [Limitations](#) when doing this).

This is a simple developer tool. It works but is not extensively tested for *boundary conditions* and as such has limited error-handling so it might emit some rather unpleasant errors if something goes wrong. Having said that, it works but it's a *work-in-progress* and will be refined, as needed, as time goes by.

4.1 Known Limitations

1. List re-ordering is not implemented. This program sets the lists in their `position` order but cannot re-order them if you change the `position` field in the definition file.
2. Issue boards are not created automatically here. You will need to navigate to your new project and navigate to `Issues > Boards` so that GitLab can create an empty board for you that this tool can then initialise. if board does nto exist `lablabel` wil tell you.