Saamwerk

Resources to collaborate on open-source projects

Contents

1 Introduction **5**

2 Filing a new issue **7**
2.1 Before filing an issue 7
2.2 Actually filing an issue 7

3 Labelling Strategy **9**
3.1 Overview 9
3.2 Type 10
3.3 Status 10
3.4 Priority 10
3.5 Complexity 10
3.6 Meta 11

4 Creating a pull request **13**
4.1 Before creating a pull request 13
4.2 Actually creating a pull request 13

Chapter 1

Introduction

saamwerk (collaborate in Africaans) is a collection of resources that should help facilitate the collaboration on open-source projects.

Feel free to open an issue on potential extensions of saamwerk.

Chapter 2

Filing a new issue

2.1 Before filing an issue

Before you file an issue, please do the following:

- make sure you are using the development version of the project. It's possible that you are not the first person experiencing a particular issue so we want to avoid duplicate issues.
- make sure your issue is not already filed. Quickly look at the open issues of the repository or - if there are many open issues - do a quick keyword search. Filtering the issues according to labels might help.

2.2 Actually filing an issue

- use a concise title for your issue that describes the problem. Don't categories the issue with a keyword like "feature request: feature xyz". This will be taken care of by the appropriate labeling from a collaborator of the repository.
- create a minimal reproducible example (see also reprex) that showcases the offending behavior.

Chapter 3

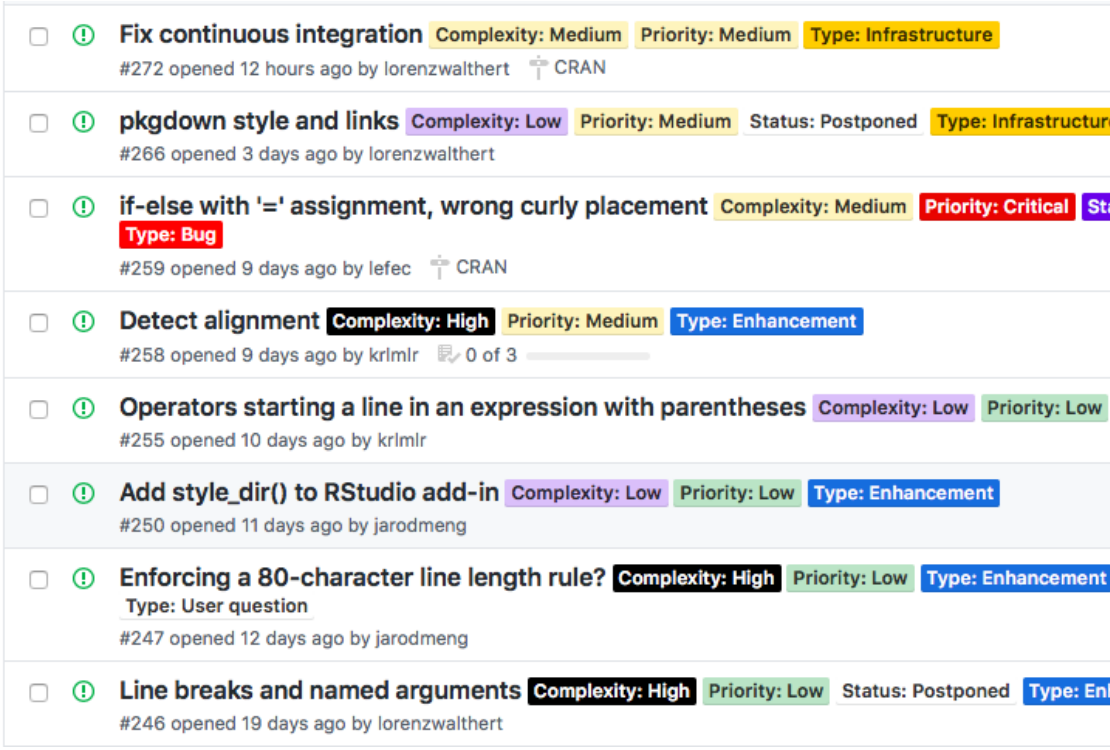
Labelling Strategy

3.1 Overview

This documentation outlines the labelling strategy that was used for dealing with issues. It was inspired by this blog post. There are four types of categories used to label issues.

- Type
- Status
- Complexity
- Priority

That may then look like this.



The colors of the labels were chosen such that dark / red corresponds to urgent / complex etc, whereas bright / green etc. corresponds to not urgent / simple and so on.

Labels should be assigned to issues as soon as possible.

In the next section, we will introduce every category in detail and descending order.

3.2 Type

Type describes the type of the issue, it can be one of the following:

- Bug: Everything ranging from unexpected behavior to faulty behavior.
- Enhancement: An issue that relates to adding a feature to the project that results in an API change.
- Infrastructure: Similar to Enhancement, but this refers to internal changes and re-factoring, maintenance etc. rather than to API changes that affect the user experience.
- User Question: Questions from users that are not really any of the above, but rather just need clarification on the project.
- Unassigned: No type assigned yet.

3.3 Status

Status describes the status of the issue, it can be one of the following:

- WIP: This is work in progress from some contributor and hence, it does not make sense for other people to also work on it.
- Postponed: The issue was postponed, so it should not be expected to be resolved in the near future. Reasons for an issue to get postponed might be that the issue is blocked by some other issue or the lack of time (from the developers side) to resolve it quickly given that there are other (more urgent) issues. **Postponed issues are closed, even though they are not solved.**
- Unassigned: No status assigned yet.

3.4 Priority

Priority describes the relevance of the issue, it can be one of the following:

- Critical: This issue is really pressing and affects the functionality of the project in a substantial fashion. It needs to be solved as soon as possible.
- High: Issue is urgent.
- Medium: Not so urgent.
- Low: This is a detail and does not affect the main functionality of the project.
- Unassigned: No priority assigned yet.

3.5 Complexity

Complexity describes the complexity of the issue, it can be one of the following:

- High: Very complex issue that involves different aspects of the project that need to be re-factored to solve the issue. Very time-consuming and difficult to solve. Requires large background knowledge of the project.
- Medium: Might bear some complexity, time needed to solve this can be anticipated.
- Low: Almost trivial problem, can be resolved very quickly and without much knowledge of the project.
- Unassigned: No complexity assigned yet.

3.6 Meta

Meta describes meta aspects of the issue, it can be one of the following:

- Duplicate.

Chapter 4

Creating a pull request

4.1 Before creating a pull request

Before you create a PR, please do the following:

- think about what your role is in the project. Are you in a position where you can contribute code to the project (via a PR) or does it make more sense if you focus on pointing to the problems and let other people fix it (via issues) that have more experience with the project?
- if you think you are suited to make a code contribution, make sure the changes you suggestion are welcome. You can best check that by opening an issue on the repository. To make clear you are intending to create the PR that would resolve the issue, you can use key words in the issue title to indicate that you want to know whether the maintainers would be open to your suggestions. For example, the issue title "Consider removing dplyr dependency" indicates already that you would like to remove a dependency from the project.
- Once you have the clearance from the repo maintainers, make sure you are using the development version of the project before you start contributing.

4.2 Actually creating a pull request

- first read the contributing guide of the project if there is one and try to understand the structure of the project. If the project is large, there is most likely some documentation available that helps you to get started. Pay attention to the style and conventions used in the project by looking at other code. Your contribution should fit as well as possible in the code base.
- Use git wisely, i.e. break down your contribution in smaller proportions and commit each of them separately with an informative commit message.
- make sure to include / update unit tests and documentation along the new code.
- when creating the pull request, reference the issue and describe the changes you want to introduce on a high level. If you are uncertain about whether some of your contribution is adequate, include questions about it in the pull request to make it easy for reviewers to spot potential issues.