zulipbot
Optimizing the GitHub workflow and more

Cynthia Lin & Joshua Pan
Open Source Summit North America 2017
Terminology

- GitHub
  - Repository
    - Where most of a project’s source code is located (codebase)
  - Workflow
    - Organized, systematic sequence of practices to commit code
  - Issue tracker
    - Feature for project-related issues
  - Pull request
    - Way to submit contributions to repository
Terminology

- Open source community
  - Project maintainer
    - Leader of open source project
  - Contributor
    - Any user that has progressed the codebase through code contributions
Good workflow qualities
Organized, efficient, productive

- Active community
  - Reported issues immediately responded to and resolved
  - Pull requests are quickly amended
  - Code reviews are performed often
  - Contributions promptly merged

- Organized repository
  - Issues labeled
  - Continuous integration/coverage suites
  - Documentation
Ideal GitHub workflow

1. **Issue filed in the issue tracker**
   a. Project maintainer reviews and labels issue
      i. Assigns the issue to a contributor to work on
      ii. Interested contributor comments takes up issue

2. **Contributor creates pull request**
   a. Project maintainer reviews pull request
      i. Additional changes requested; contributor amends changes
      ii. Changes approved = pull request merged
Common GitHub workflow issues

- Repository permissions
  (help.github.com/articles/repository-permission-levels-for-an-organization/)
  - Possibilities: Read/write/admin
  - Most contributors don’t have write access to repository but can help out with issue triage
  - Project maintainers forced to waste time on issue triage
  - Less code reviews and merged PRs
Common GitHub workflow issues

- **Repository notifications**
  
  (help.github.com/articles/about-notifications/)

  - Possibilities: watching/not watching/ignoring
  - Active repositories generate high amounts of notifications
  - Contributors choose to not watch events
  - Will not be notified of any important issues unless explicitly mentioned/subscribed
Common GitHub workflow issues

- **Inactive contributors**
  - Progress ceased on fixing claimed issues/pull requests
  - Project maintainers waste time revisiting and reassigning issues

- **Pull request status**
  - Unclear what pull request’s review status is sometimes
  - Merge conflicts and failing continuous integration builds don’t notify PR author (only visible through GitHub UI)
Tim Abbott

Right now it's kinda hard for new contributors to figure out which issues are currently being worked on. I think it'd be great to build a bot that manages this, using something like the following protocol:

- When someone wants to claim an issue, they @-mention the bot in a comment on the issue, and the bot then either assigns the issue to that person in the GitHub UI (if possible) and adds an in-progress label to the issue.
- If the issue remains open for a week without activity or a PR being opened, it posts a comment on the issue and marks it as no-longer-in-progress (maybe after a day for the person to say they're still working on it).

Then, developers looking for projects can filter for issues that don't have the in-progress label on them to find something to pick up without looking through a bunch of issues that maybe someone is working on.

Thoughts?

Steve Howell

So I think there are a few different ways to make our Github tooling better. There are some fairly simple, isolated things we can do like automatically find PRs that seems to have been abandoned, or automatically find issues that need to be labeled. It's always good to find areas where there are quick, small wins to productivity. And this should really be our first priority.

But I also think it's useful to brainstorm about grander, more-far-reaching goals, like intelligent tooling for collaboration and filtering/searching of PRs.
How zulipbot resolves these problems

- Makes issue triage accessible to contributors
  - Keywords in issue comments act as commands
  - Issue assignment:
    - @zulipbot claim & @zulipbot abandon
    - Labels issue as “in progress”
  - Labelling issues:
    - @zulipbot label “bug” & @zulipbot remove “bite size”
  - Configurable command aliases & “in progress” label name
How zulipbot resolves these problems

- **Notifies contributors on issues within their areas of expertise**
  - System of “area” labels denoting corresponding part of codebase
  - Each label has accompanying GitHub organization team
    (help.github.com/articles/organizing-members-into-teams/)
  - Examples:
    - “area: testing-coverage” → “server-testing”
    - “area: testing-infrastructure” → “server-testing”
    - “area: tooling” → “server-tooling”
  - Area team members notified when area label is added
  - Configurable area labels, teams, and notification messages
How zulipbot resolves these problems

● Monitors issues & pull requests for inactivity
  ○ Checks for time when issue was last updated
  ○ Issue assignee/PR author prompted for progress after certain time period of inactivity
    ■ Posts comment if they are still working
    ■ Abandon issue/close PR if not interested in continuing progress
    ■ Lack of response = automatic issue abandon
  ○ Configurable inactivity time limits and notifications
How zulipbot resolves these problems

- Tracks pull request states
  - Pull requests labelled according to review status
    - “needs review”
    - “reviewed”
  - Pull request authors notified when PRs develop merge conflicts
  - Primary pull request reviewers assigned to PRs
  - Pull request authors notified when Travis CI builds complete
  - Configurable pull request state checks and review label names
Technical specifications

- Written in Node.js
  - Asynchronous, event-driven, non-blocking I/O model
  - Fast Google V8 JavaScript engine
  - Community-developed modules easily installed and implemented
Technical specifications

- GitHub REST API v3 (https://developer.github.com/v3/)
  - JSON-formatted data sent and received from https://api.github.com
  - Requests sent to different endpoints for different actions
  - Utilizes node-github API wrapper (https://github.com/mikedeboer/node-github)

- Express web server
  - Handles all requests
  - Verifies webhook payloads
zulipbot in action

@zulipbot claim

zulipbot assigned lonerz on May 21

zulipbot added the in progress label on May 21

zulipbot commented 13 days ago

Hello @zulip/server-development, @zulip/server-tooling members, this issue was labeled with the area: emails, area: tooling labels, so you may want to check it out!
zulipbot in action
**zulipbot in action**

**zulipbot commented on Jun 29**

Heads up @lonerz, we just merged some commits that conflict with the changes you made in this pull request! You can review this repository's recent commits to see where the conflicts occur. Please rebase your feature branch against the `upstream/master` branch and resolve your pull request's merge conflicts accordingly.

**zulipbot commented on Jul 31**

Hello @lonerz, a Zulip maintainer reviewed this pull request over a week ago, but you haven't updated your pull request since. Please take a look at the requested changes and update your pull request accordingly.

Thank you for your valuable contributions to Zulip!
Hello @lonerz, you claimed this issue to work on it, but this issue and any referenced pull requests haven't been updated for 7 days. Are you still working on this issue?

If so, please update this issue by leaving a comment on this issue to let me know that you’re still working on it. Otherwise, I’ll automatically remove you from this issue in 3 days.

If you've decided to work on something else, simply comment @zulipbot abandon so that someone else can claim it and continue from where you left off.

Thank you for your valuable contributions to Zulip!
Community growth

- **Zulip**
  - Active, fastest-growing open source group chat
  - Main web app repository gained over 150 new contributors
  - Over 3000 commits merged in 4 months

- **zulipbot**
  - Over 4000 comments made on 2000 issues and pull requests
  - Deployed on Zulip mobile app, desktop app, Python API, and of course zulipbot repositories
  - 300 commits and 5 unique contributors to the project
Reflections

- Refactoring is painful but necessary
- Command handlers are cool
- Documentation is important
- Automated tests and code linting save time
- Every contribution counts
- A good workflow leads to a flourishing project
Future plans

- Additional refactoring to the codebase
- Coverage tests
- Zulip integration
- Other continuous integration service integrations
- Public release v1.0.0!
Thank you for listening!

https://github.com/zulip/zulipbot

Cynthia Lin
@synicalsyntax
synicalsyntax
synicalsyntax.me

Joshua Pan
@_lonerz_
lonerz
Joshua Pan