Write Better Docs

Moar inclusivity!
Moar quality!
Open Source Summit 2018
Hi!

I’m Christie Wilson!
Engineer @ Google
Container Tools Team
The Open Source Dream
Github 2017 stats

24 million TOTAL USERS

0.5 billion PUBLIC COMMITS SINCE SEPTEMBER 2016
The Open Source Dream
The Open Source Dream

NEEDS DOCS
Docs for Developers
1. Issues
2. CONTRIBUTING.md
   + DEVELOPMENT.md
3. A surprise!
Issues
Contributing: finding an issue
Explore integration with foobar #68

@bobcatfish opened this issue on Mar 7 · 0 comments

bobcatfish commented on Mar 7

github.com/bobcatfish/foobar

bobcatfish added the help wanted label on Mar 7
We should support the caching pattern in the wizzbang whatsit.

It's not clear to me that wizzbang can cache yet (I haven't followed), but once it can we should have the whatsit use an `emptyDir` thingamabob as a cache that's overridable as a parameter, and document how to provide a PVC as an interwhatsit cache.
Example of a good issue

github.com/knative/serving/issues/1859
I want to contribute!

Easy to find issues and get started?

Give up

Ask maintainers

Let’s do it!

Yep!
Talking with maintainers
The 3 keys to great issues!

- Observed behaviour
  - Links to the relevant code
  - Snippets
- Expected behaviour
- Steps to repro
  - Environment
  - Commands

Screenshots, images, links
Github features

- Use an issue template
Github features

- Use labels!
  - good-first-issue
  - help-wanted
How does Jaz get started on the issue?

- Needs to know how to run tests
- Your lib also runs as a service, needs to be able to run it
- Needs to know how the pull request workflow works
foobar

A simple implementation

Build Setup

# install dependencies
npm install
1. Issues
2. CONTRIBUTING.md + DEVELOPMENT.md
3. A surprise!
CONTRIBUTING.md
and
DEVELOPMENT.md
I’ve got a fix!

How do I do dev for this project?

???

Throw it over the fence

Tests fail!

Fix with maintainer help

Fix easily

Clear process!

Verify the fix, open a PR!

Give up

Give up
<table>
<thead>
<tr>
<th>CONTRIBUTING.md</th>
<th>DEVELOPMENT.md</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard recognized by GitHub (e.g., integrated into PRs)</td>
<td>Extends CONTRIBUTING.md (less standard)</td>
</tr>
<tr>
<td>CONTRIBUTING.md</td>
<td>DEVELOPMENT.md</td>
</tr>
<tr>
<td>-----------------</td>
<td>----------------</td>
</tr>
<tr>
<td><strong>HOW</strong> to contribute to your repo.</td>
<td><strong>WHAT</strong> a maintainer/contributor does.</td>
</tr>
<tr>
<td><strong>Examples:</strong></td>
<td><strong>Examples:</strong></td>
</tr>
<tr>
<td>- PR process</td>
<td>- Build steps</td>
</tr>
<tr>
<td>- Issue requirements</td>
<td>- Running tests</td>
</tr>
<tr>
<td>- Commit message style</td>
<td>- Debugging tips</td>
</tr>
</tbody>
</table>
Hey open source contributors: what’s that one thing you love to see on a project that makes it easier to contribute to? How about that thing that makes you “happy” to contribute?

RTs for visibility.

Kind, welcoming project admins who approach discussion from a collaborative attitude vs. putting up barriers.
How do I find open source projects to help with?

All of the ones I’ve found so far are too complicated for me to understand, but people tell me that I should help with open source projects to improve my knowledge. What do I do?
foo

A simple implementation

Build Setup

# install dependencies
npm install
How to write a great CONTRIBUTING.md

- Be friendly :D
- This is the welcome mat to your repo!
- A greeting to potential new contributors!
- The beginning of a new relationship
How to write a great CONTRIBUTING.md

- How to open an issue
- How to suggest features
- What contributions you’re looking for
- Project philosophy
- Project roadmaps
- Slack/group chat/mailing lists
If all projects were the same....

- Who here uses Go?
If all projects were the same....

- Who here uses Go?
- Who builds and tests **ONLY** with:
  - `go test`
  - `go build`
If all projects were the same....

- Who here uses Go?
- Who builds and tests ONLY with:
  - go test
  - go build
- i.e. you never have to use Makefiles, or scripts, etc.
Unless all projects are identical,
If you want contributors,
you need to explain how your project works!
Get the process out of your brain and into markdown!
How to write a great DEVELOPMENT.md

- The less you assume, the more people can contribute!
- Even super experienced folks:
  - Are new to your project
  - May use different tools
  - Have had different experiences
Even experienced devs use different tools

Jaz uses:
- Github
- Go
- Kubernetes

Andy uses:
- Java
- SVN
- In-house bare metal

Wendy uses:
- C#
- Visual Studio
- Azure
How to write a great DEVELOPMENT.md

- How to fork
- How to setup an environment for development
  - Tool and resource requirements
  - Interacting with the system
- How to iterate
  - How do you check something works? e.g. tests, poking manually
  - e.g. keeping vendor/ up to date, redeploying, cleaning up
Development

This doc explains how to setup a development environment so you can get started contributing to Knative Serving. Also take a look at:

- The pull request workflow
- How to add and run tests
- Iterating

Getting started

1. Create a GitHub account
2. Setup GitHub access via SSH
3. Install requirements
4. Set up a kubernetes cluster
5. Set up a docker repository you can push to
6. Set up your shell environment
7. Create and checkout a repo fork

Once you meet these requirements, you can start Knative Serving!

Before submitting a PR, see also CONTRIBUTING.md.

Requirements

You must install these tools:

1. go: The language Knative Serving is built in
2. git: For source control
3. dep: For managing external Go dependencies.
4. ko: For development.
5. kubectl: For managing development environments.

Environment setup

To start your environment you’ll need to set these environment variables (we recommend adding them to your .bashrc):

1. GOPATH: If you don’t have one, simply pick a directory and add export GOPATH...
1. Issues
2. CONTRIBUTING.md
   + DEVELOPMENT.md
3. A surprise!
Jaz found a bug in your lib!

```python
raise InvalidAge(age_in_cat_years)

if not 5 <= age_in_cat_years <= MAX_CAT_AGE:
    raise InvalidAge(age_in_cat_years)
```
Jaz found a bug in your lib!

```python
raise InvalidAge(age_in_cat_years)

if not 5 <= age_in_cat_years <= MAX_CAT_AGE:
    raise InvalidAge(age_in_cat_years)
```
Why 5?

<table>
<thead>
<tr>
<th>Fix tests</th>
<th>13 days ago</th>
<th>26</th>
<th><code>raise</code> InvalidAge(<code>age_in_cat_years</code>)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>27</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>28</td>
<td><code>if</code> not <code>5 &lt;= age_in_cat_years &lt;= MAX_CAT_AGE:</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>29</td>
<td><code>raise</code> InvalidAge(<code>age_in_cat_years</code>)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>30</td>
<td></td>
</tr>
</tbody>
</table>
Why 5?

Fix tests
1. Issues
2. CONTRIBUTING.md
   + DEVELOPMENT.md
3. Commit messages
Commit Messages

Thanks Steven Erenst for teaching me about the value of commit messages!
I found a bug!

Can I fix it?

Nope!

Give up

Open an issue

Fix it!

Yep!
Commits often look like this:

“Fix missing logs”

“Fixes #22 and #5”

“Bump the timeout to 20 min”
Dan Abramov @dan_abramov · 24 Mar 2016
What is the main reason holding you back from contributing to React?

- 7% Review time is slow
- 52% Code is not approachable
- 15% My company doesn’t let me
- 26% Other (please reply)

799 votes • Final results
Example good commit message

Subject Line

Fix stats collector spinning CPU if no stats are collected

Commit fd0e24b changed
the stats collection loop to use a `sleep()` instead
of `time.Tick()` in the for-loop.

This change caused a regression in situations where
no stats are being collected, or an error is hit
in the loop (in which case the loop would `continue`,
and the `sleep()` is not hit).

This patch puts the sleep at the start of the loop
to guarantee it's always hit.

This will delay the sampling, which is similar to the
behavior before fd0e24b.

Signed-off-by: Sebastiaan van Stijn <github@gone.nl>

master (#36609)

thajeztah committed 2 days ago  Verified
y tho

“logrus use full timestamp”
### Change default action to ‘inplace’

```
# Optional - record format
-
config_param :action, :string, :default => 'replace'
+
config_param :action, :string, :default => 'inplace'

config_param :keep_key, :bool, :default => false
```
y tho

"Change git commit to use short ID instead"
Context switching

```
Change git commit to use short ID instead

shaStr := hex.EncodeToString(sha[:])[16]
suffix = fmt.Sprintf("dirty-%s", shaStr)
}
cmd := exec.Command("git", "rev-parse", "--short", "HEAD")
stdout, _, err = util.RunCommand(cmd, nil)
if err != nil {
    return "", errors.Wrap(err, "determining current commit")
}
commit := strings.TrimSuffix(string(stdout), "\n")
if suffix != "" {
    return fmt.Sprintf("%s:%s-%s", opts.ImageName, suffix, commit), nil
}
return fmt.Sprintf("%s:%s", opts.ImageName, commit), nil
```
Context switching

Change git commit to use short ID instead

Commit

```go
shaStr := hex.EncodeToString(sha[:16])
suffix = fmt.Sprintf("dirty-%s", shaStr)
}
cmd := exec.Command("git", "rev-parse", "--short", "HEAD")
stdout, _, err = util.RunCommand(cmd, nil)
if err != nil {
    return "", errors.Wrap(err, "determining current commit")
}
commit := strings.TrimSuffix(string(stdout), "\n")
if suffix != "" {
    return fmt.Sprintf("%s:%s-%s", opts.ImageName, suffix, commit), nil
}
return fmt.Sprintf("%s:%s", opts.ImageName, commit), nil
```
Context switching

Change git commit to use short ID instead

Commit  ➔  Pull Request
Context switching
Context switching
Example good commit message

Subject Line

Fix stats collector spinning CPU if no stats are collected

Background

Commit fd0e24b changed
the stats collection loop to use a `sleep()` instead of `time.Tick()` in the for-loop.

This change caused a regression in situations where no stats are being collected, or an error is hit in the loop (in which case the loop would `continue`, and the `sleep()` is not hit).

Description

This patch puts the sleep at the start of the loop
to guarantee it's always hit.

This will delay the sampling, which is similar to the behavior before fd0e24b.

Signed-off-by: Sebastiaan van Stijn <github@gone.nl>
How to write great commit messages

- Subject line (what does this commit do?)
- Body that explains what and why vs. how
- Two paragraphs
What + Why v.s. How

How do you fill 2 paragraphs?

- **What** is the problem being solved?
- **Why** is this the best approach?
- **What** other approaches were considered?
- **What** side effects will this approach have?
- **What** future work remains to be done?

Thanks Thomas Stromberg!
Great docs are a prereq for a strong contributor base!
Great docs for a better world!

- Remote first!
- Different lifestyles!
- People all over the globe!
- Welcome different kinds of contributors!

---

i identify as tired
@ScribblingOn

For a new contributor, a little appreciation & an initial push can go a really long way.

#opensource
6:58 AM - 17 Sep 2016
How?

Make docs part of your process!

1. Dogfood your own internal docs
2. Complete PR = Code + Tests + Docs
I want to contribute!

Easy to find issues and get started?

Give up

Ask maintainers

I've got a fix!

How do I do dev for this project?

Give up

Throw it over the fence

Tests fail!

Fix with maintainer help

Fix easily

Clear process!

I found a bug!

Can I fix it?

Give up

Open an issue

I've got a fix!

How do I do dev for this project?

Give up

Throw it over the fence

Tests fail!

Fix with maintainer help

Fix easily

Clear process!
Thanks!
Links!

CONTRIBUTING.md:
https://help.github.com/articles/setting-guidelines-for-repository-contributors/
https://opensource.guide/starting-a-project/#writing-your-contributing-guidelines
https://github.com/nayafia/contributing-template/blob/master/CONTRIBUTING-template.md
Community building:
https://help.github.com/categories/building-a-strong-community/
https://opensource.guide/starting-a-project/
https://medium.freecodecamp.org/how-to-attract-new-contributors-to-your-open-source-project-46f8b791d787
GitHub and issues:
https://github.com/codeforamerica/howto/blob/master/Good-GitHub-Issues.md#writing-good-github-issues
https://help.github.com/articles/helping-new-contributors-find-your-project-with-labels/
Commit messages:
Implementing a Strong Code Review Culture
https://chris.beams.io/posts/git-commit/