# 开源贡献指南

周明辉

# 大纲

- 开源贡献的准备
  - 如何选择合适的项目；
  - 学习开源社区如何运作；
  - 如何做第一个贡献。

- GitHub实践：如何做第一个贡献
  - 在Github上找到一个自己感兴趣的项目；
  - 从项目中筛选出适合新人解决的issues；
  - 挑选出准备解决的issue，明确需要解决的任务；
  - 以符合项目贡献规范的方式提交贡献。

## 开源贡献的准备

A. How to choose an appropriate open source project to contribute?

B. Learn how the open source community work.

C. How to make your first contribution?

## A. How to choose an appropriate open source project to contribute?

- 1) 首先考虑你已经使用或想要使用的项目。
  - The projects you’ll actively contribute to are the ones you find yourself coming back to.
- 2) 选择你非常有兴趣的项目。
  - For example, if you are interested in search engine and want to know how it works. First of all, you should study technical literature to learn the basic technical points. After that, you can choose Lucene, Sphinx to learn through practice.
- 3) 选择你熟悉的项目。
  - If you are familiar with the programing language and the technology that the project uses, it will be much easier for you to contribute.
- 4) 选择能够独立运行的项目。
  - There are many kinds of open source projects. Although many of them represent stand-alone software, most projects, e.g. the plugins of other projects, libraries, that would require you first to understand the underlying frameworks. Some of the projects may require very complex installation and configure in order to test the software. You may want to avoid them unless the time investment would be worth-while.
- 5) 选择活跃的项目。
  - The activeness of the project usually contains two parts. The frequency of the commits, and the enthusiasm of the discussion in the community. The higher frequency of the commits, the more active it is. Higher enthusiasm can help you find the answer easier when you have problems.
- 当你找到一个你想要贡献的项目时，快速搜索以确保该项目正在接受贡献。 否则，你的努力可能永远得不到回应。

# A handy checklist to evaluate whether a project is suitable for new contributors.

- 满足开源定义：Does it have a license? Usually, this is a file called LICENSE in the root of the repository.
- 项目积极接受贡献: Look at the commit activity on the master branch. On GitHub, you can see this information on a repository's homepage.
- 项目代码提交的状态： When was the latest commit? How many contributors does the project have? How often do people commit? (On GitHub, you can find this by clicking "Commits" in the top bar.)
- 项目任务的状态： How many open issues are there? Do maintainers respond quickly to issues when they are opened? Is there active discussion on the issues? Are the issues recent? Are issues getting closed? (On GitHub, click the "closed" tab on the Issues page to see closed issues.)
- 项目PR的状态： How many open pull requests are there? Do maintainers respond quickly to pull requests when they are opened? Is there active discussion on the pull requests? Are the pull requests recent? How recently were any pull requests merged? (On GitHub, click the "closed" tab on the Pull Requests page to see closed PRs.)
- 项目欢迎新人：A project that is friendly and welcoming signals that they will be receptive to new contributors. Do the maintainers respond helpfully to questions in issues? Are people friendly in the issues, discussion forum, and chat (for example, IRC or Slack)? Do pull requests get reviewed? Do maintainers thank people for their contributions?

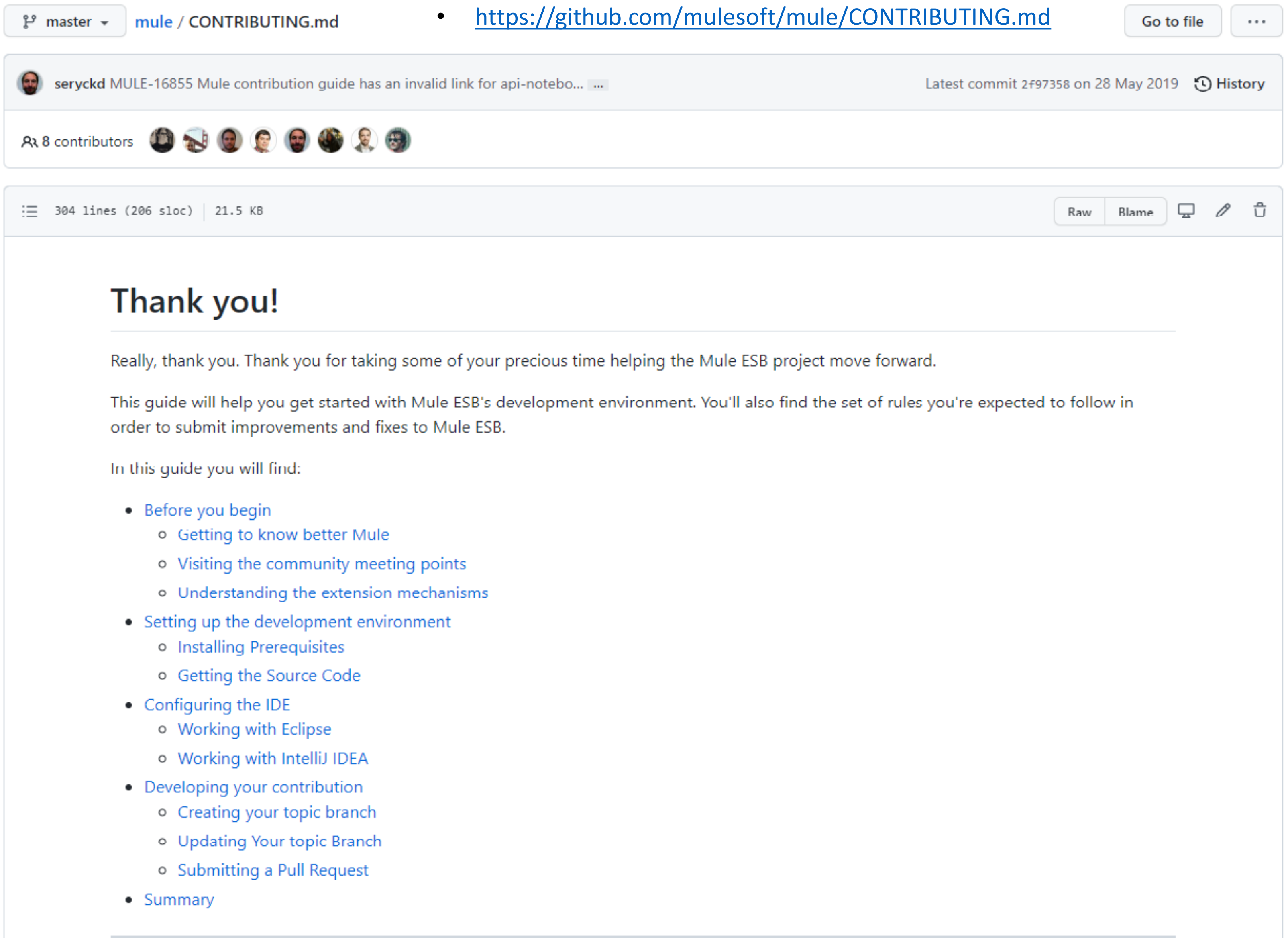
[1] How to contribute to Open Source? https://opensource.guide/how-to-contribute/#orienting-yourself-to-a-new-project

## B. Learn how the open source community works.

- 检查并阅读文档，了解社区规则
  - Be sure to remember that reading documents is much more important than reading code at first. These files are usually listed in the top level of a repository, including license, readme, contributing(.md), tutorials and so on. Reading these documents can give you much valuable information of the project in a short time. For example, readme/contributing.md is the instruction manual that welcomes new community members to the project. It explains why the project is useful and how to get started.
  - Most of large Open Source projects have a series of documents which tell the important details that new contributors need to pay attention to, e.g., In Linux Kernel community, "How to do Linux Kernel development" not only contains some rules of the community but also tells contributors what they should pay attention to in the software development process. Make sure to make good use of these valuable resources.
- 了解社区的讨论和协作方式。开源项目使用下述工具组织讨论：
  - Issue tracker: Where people discuss issues related to the project. Maybe you could find the bug there you want to deal with.
  - Pull requests: Where people discuss and review changes that are in progress.
  - Discussion forums or mailing lists: Some projects may use these channels for conversational topics. Others use the issue tracker for all conversations. Many contributors' first interactions with the Open Source community usually happen there. Please subscribe the project's mailing list, thus you can receive interesting discussions and sharing your idea.

# Example of contributing guideline

- https://github.com/mulesoft/mule/CONTRIBUTING.md



master · mule / CONTRIBUTING.md · Go to file · ···

seryckd MULE-16855 Mule contribution guide has an invalid link for api-notebo... ··· Latest commit 2f97358 on 28 May 2019 · History

8 contributors

304 lines (206 sloc) | 21.5 KB · Raw · Blame

## Thank you!

Really, thank you. Thank you for taking some of your precious time helping the Mule ESB project move forward.

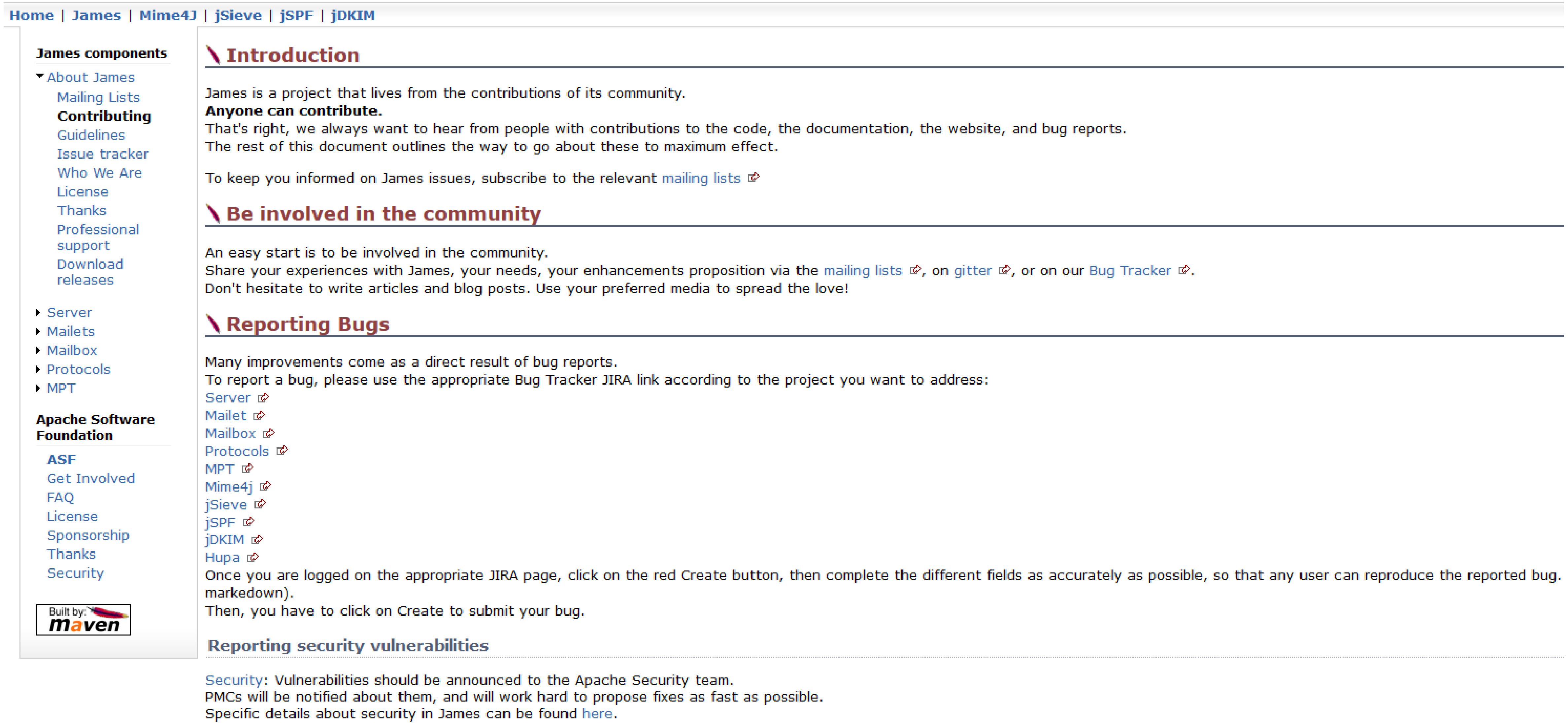
This guide will help you get started with Mule ESB's development environment. You'll also find the set of rules you're expected to follow in order to submit improvements and fixes to Mule ESB.

In this guide you will find:

- Before you begin
  - Getting to know better Mule
  - Visiting the community meeting points
  - Understanding the extension mechanisms
- Setting up the development environment
  - Installing Prerequisites
  - Getting the Source Code
- Configuring the IDE
  - Working with Eclipse
  - Working with IntelliJ IDEA
- Developing your contribution
  - Creating your topic branch
  - Updating Your topic Branch
  - Submitting a Pull Request
- Summary

# Example of contributing guideline

- https://james.apache.org/contribute.html



Home | James | Mime4J | jSieve | jSPF | jDKIM

**James components**

- About James
  - Mailing Lists
  - **Contributing**
  - Guidelines
  - Issue tracker
  - Who We Are
  - License
  - Thanks
  - Professional support
  - Download releases
- Server
- Mailets
- Mailbox
- Protocols
- MPT

**Apache Software Foundation**

- **ASF**
- Get Involved
- FAQ
- License
- Sponsorship
- Thanks
- Security

Built by: maven

## Introduction

James is a project that lives from the contributions of its community.
**Anyone can contribute.**
That's right, we always want to hear from people with contributions to the code, the documentation, the website, and bug reports.
The rest of this document outlines the way to go about these to maximum effect.

To keep you informed on James issues, subscribe to the relevant mailing lists

## Be involved in the community

An easy start is to be involved in the community.
Share your experiences with James, your needs, your enhancements proposition via the mailing lists, on gitter, or on our Bug Tracker.
Don't hesitate to write articles and blog posts. Use your preferred media to spread the love!

## Reporting Bugs

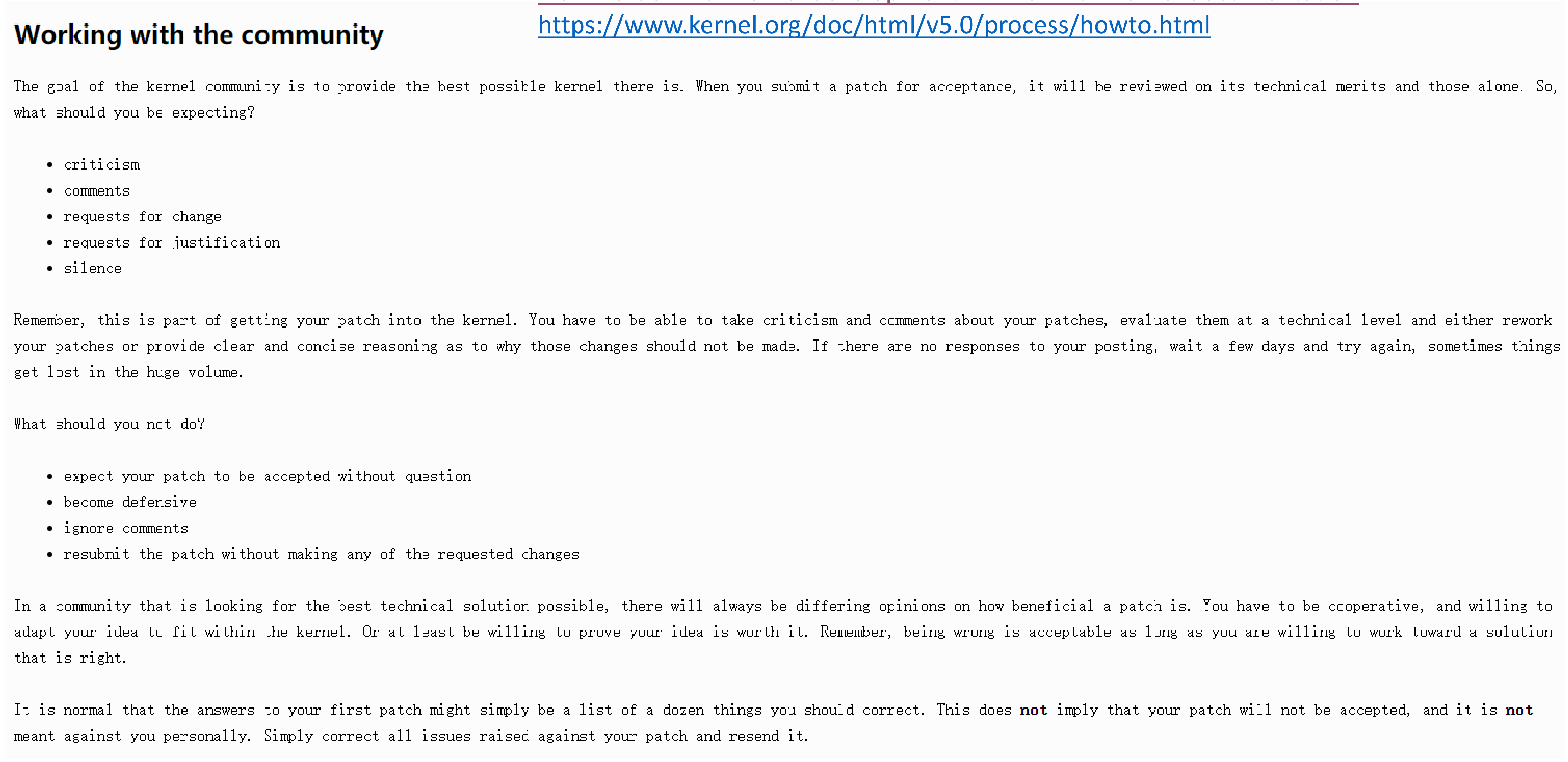
Many improvements come as a direct result of bug reports.
To report a bug, please use the appropriate Bug Tracker JIRA link according to the project you want to address:
Server
Mailet
Mailbox
Protocols
MPT
Mime4j
jSieve
jSPF
jDKIM
Hupa
Once you are logged on the appropriate JIRA page, click on the red Create button, then complete the different fields as accurately as possible, so that any user can reproduce the reported bug.
markedown).
Then, you have to click on Create to submit your bug.

### Reporting security vulnerabilities

Security: Vulnerabilities should be announced to the Apache Security team.
PMCs will be notified about them, and will work hard to propose fixes as fast as possible.
Specific details about security in James can be found here.

# 社区协作方式

HOWTO do Linux kernel development — The Linux Kernel documentation
https://www.kernel.org/doc/html/v5.0/process/howto.html



## Working with the community

The goal of the kernel community is to provide the best possible kernel there is. When you submit a patch for acceptance, it will be reviewed on its technical merits and those alone. So, what should you be expecting?

- criticism
- comments
- requests for change
- requests for justification
- silence

Remember, this is part of getting your patch into the kernel. You have to be able to take criticism and comments about your patches, evaluate them at a technical level and either rework your patches or provide clear and concise reasoning as to why those changes should not be made. If there are no responses to your posting, wait a few days and try again, sometimes things get lost in the huge volume.

What should you not do?

- expect your patch to be accepted without question
- become defensive
- ignore comments
- resubmit the patch without making any of the requested changes

In a community that is looking for the best technical solution possible, there will always be differing opinions on how beneficial a patch is. You have to be cooperative, and willing to adapt your idea to fit within the kernel. Or at least be willing to prove your idea is worth it. Remember, being wrong is acceptable as long as you are willing to work toward a solution that is right.

It is normal that the answers to your first patch might simply be a list of a dozen things you should correct. This does **not** imply that your patch will not be accepted, and it is **not** meant against you personally. Simply correct all issues raised against your patch and resend it.

## C. How to make your first contribution?

- 第一贡献原则：开源项目中有很多类型的任务需要人手，不仅仅是代码开发。
- 贡献第一种：send an email
  - The first contribution usually is not writing code. Researchers found that often a significant period of observation (lurking), ranging from a couple of weeks to several months, was needed before a joiner became a developer [2]. However, the standard deviation was quite high, due to the variance of technical skills and the time that the contributors devote. Therefore, please choose the project that matches with your specialized knowledge and try to spend a little more time. During this time, new contributors usually participate in discussion via email. If you want to join an open-source project, the first thing that you should do is getting on the mailing list. The following list gives the types of activities you could try in your email [2]. Maybe your email will receive no response, don’t give up.
    - 1) Ask question
    - 2) General technical discussion
    - 3) Usage feedback (no bug report)
    - 4) Request for help
    - 5) Point to technical resources/refer to other projects
    - 6) Express interest to contribute
    - 7) Suggestion for improvements
    - 8) Propose/outline bugfix (no code)
    - 9) Coordination and organization discussion
    - 10) User support
    - 11) Answer (technical) question
    - 12) Self introduction
    - 13) Announcing “external” contribution
    - 14) Give Feedback on others contribution

[2] Krogh G V, Spaeth S, Lakhani K R. Community, joining, and specialization in open source software innovation: a case study[J]. Social Science Electronic Publishing, 2003, 32(7):1217-1241.

# Make your first contribution -- continued

- 贡献第二种：Activities in issue tracking system
  - 报告bug：Report an issue Before reporting an issue, you need make sure this issue has not been reported by others (You could check it in the issue tracker system). Then, you need to describe what the issue is, the context and how to reproduce it.
  - 评论问题：Comment on an issue These comments might be an interpretation for a confusing report, or suggesting a possible solution. Researchers [3] found that starting from a comment instead of reporting an issue also reflects such an attitude — it means she is intentionally getting involved, perhaps by first finding a similar issue and commenting on it instead of simply reporting an issue she encounters as a user.
  - 提交代码: It is much difficult for new contributors to contribute code. If you want to challenge yourself at beginning, the first thing you should do is choosing the appropriate component. The following three items may help you.
    - a) Choose the component which is easy to modify and coding. For example, some components perform simple tasks, such as documentation. On the contrary, in a product, some domains are always considered to be more difficult than others, e.g., some critical modules. For new contributors, try to avoid these kind of components.
    - b) Choose the component with easier computer language. Some languages are complex and difficult to learn, while others, i.e. simple script languages can be mastered fairly easily.
    - c) Choose the component which has less dependent on others. This allows contributors to use existing functionalities without having to understand the rest of the specific functionalities used by other components. And such components can be added or removed at any point, often without having to recompile the whole source code.

[3] Zhou M, Mockus A. What make long term contributors: willingness and opportunity in OSS community[C]// International Conference on Software Engineering. IEEE, 2012:518-528.

## GitHub上的学习资源

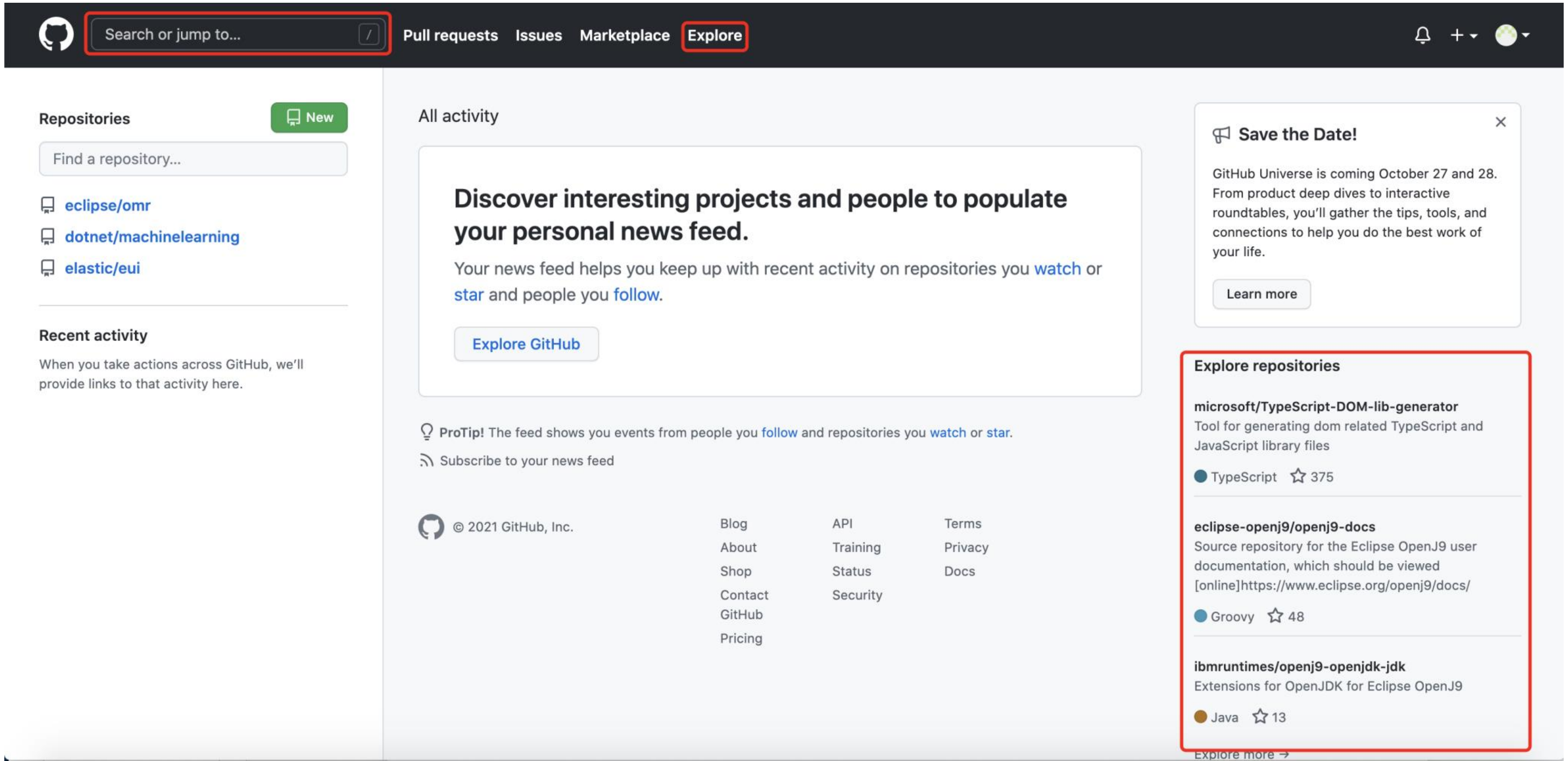
- If the project is on GitHub, check out Make a Pull Request (https://makeapullrequest.com/), which @kentcdodds created as a walkthrough video tutorial.
- You can also practice making a pull request in the First Contributions repository, created by @Roshanjossey.
- Or, you can search GitHub with “make a pull request”.

## 如何作出第一个贡献

1. 在Github上找到一个自己感兴趣的项目；
2. 从项目中筛选出适合新人解决的issues；
3. 挑选出准备解决的issue，明确需要解决的任务；
4. 以符合项目贡献规范的方式提交贡献。

## 1、在github.com上找到一个自己感兴趣的项目：

- 在左上角的搜索栏用关键词搜索感兴趣的领域的项目；
- 在Explore页面浏览Github个性化推荐的项目。



eclipse/omr
dotnet/machinelearning
elastic/eui

Recent activity

When you take actions across GitHub, we'll provide links to that activity here.

All activity

Discover interesting projects and people to populate your personal news feed.

Your news feed helps you keep up with recent activity on repositories you watch or star and people you follow.

Explore GitHub

ProTip! The feed shows you events from people you follow and repositories you watch or star.

Subscribe to your news feed

© 2021 GitHub, Inc.

Blog
About
Shop
Contact GitHub
Pricing
API
Training
Status
Security
Terms
Privacy
Docs

Save the Date!

GitHub Universe is coming October 27 and 28. From product deep dives to interactive roundtables, you'll gather the tips, tools, and connections to help you do the best work of your life.

Learn more

Explore repositories

microsoft/TypeScript-DOM-lib-generator
Tool for generating dom related TypeScript and JavaScript library files
TypeScript ☆ 375

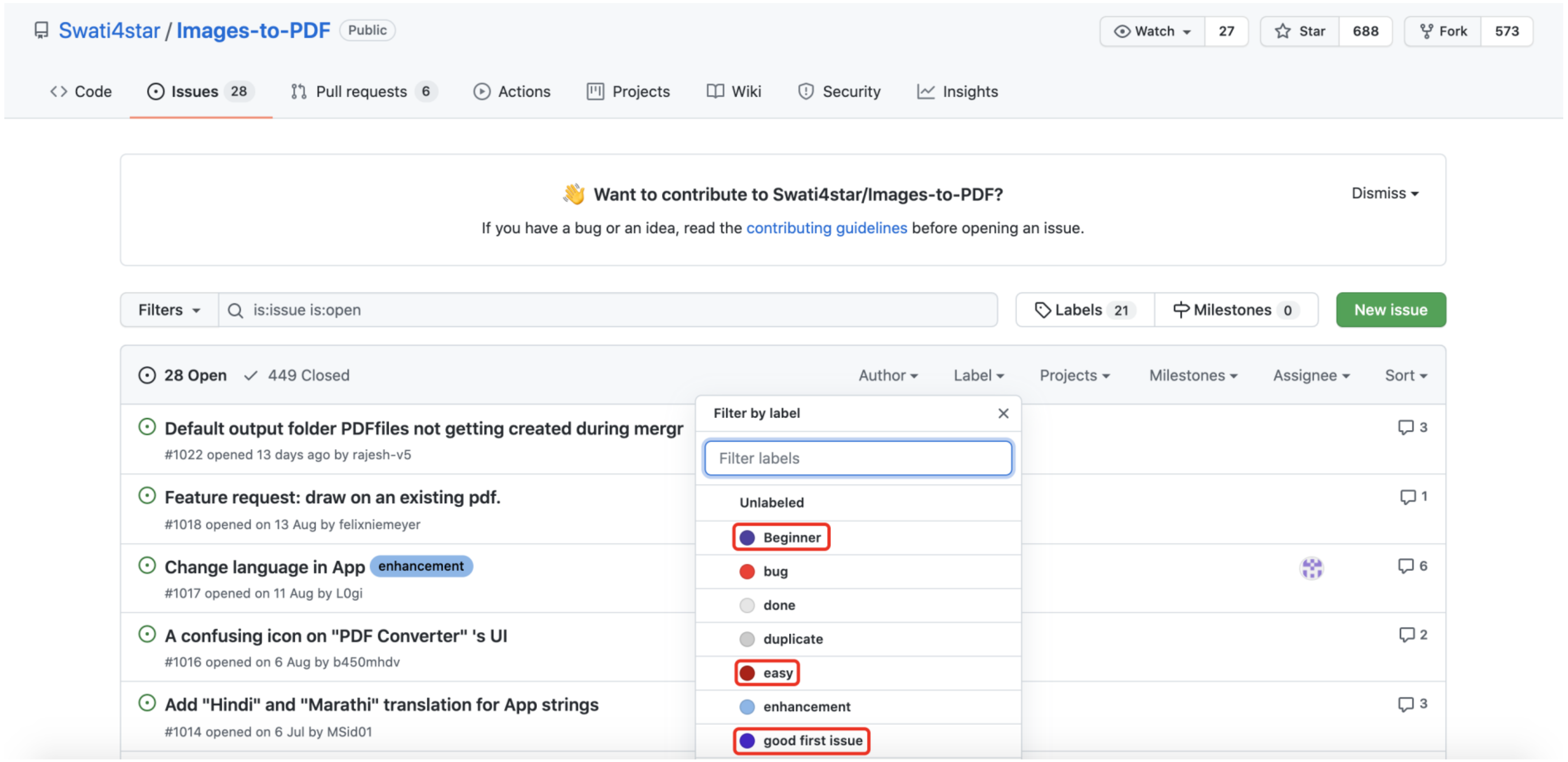
eclipse-openj9/openj9-docs
Source repository for the Eclipse OpenJ9 user documentation, which should be viewed [online]https://www.eclipse.org/openj9/docs/
Groovy ☆ 48

ibmruntimes/openj9-openjdk-jdk
Extensions for OpenJDK for Eclipse OpenJ9
Java ☆ 13

Explore more →

## 2. 从项目中筛选出适合新人解决的issues

- 在open issues里筛选：
  - 选择带有“good first issue”, “easy”, “beginner”等标签的issues。不同项目使用的“适合新人”的标签可能不同，需自行判断。
  - 避免选择正在被解决的issue(例如issue已被分配给某个开发者,或被pull request引用)。


Swati4star / Images-to-PDF Public

Watch 27 | Star 688 | Fork 573

Code | Issues 28 | Pull requests 6 | Actions | Projects | Wiki | Security | Insights

👋 Want to contribute to Swati4star/Images-to-PDF?
If you have a bug or an idea, read the contributing guidelines before opening an issue.
Dismiss

Filters | is:issue is:open | Labels 21 | Milestones 0 | New issue

28 Open ✓ 449 Closed | Author | Label | Projects | Milestones | Assignee | Sort

Default output folder PDFfiles not getting created during mergr
#1022 opened 13 days ago by rajesh-v5

Feature request: draw on an existing pdf.
#1018 opened on 13 Aug by felixniemeyer

Change language in App enhancement
#1017 opened on 11 Aug by L0gi

A confusing icon on "PDF Converter" 's UI
#1016 opened on 6 Aug by b450mhdv

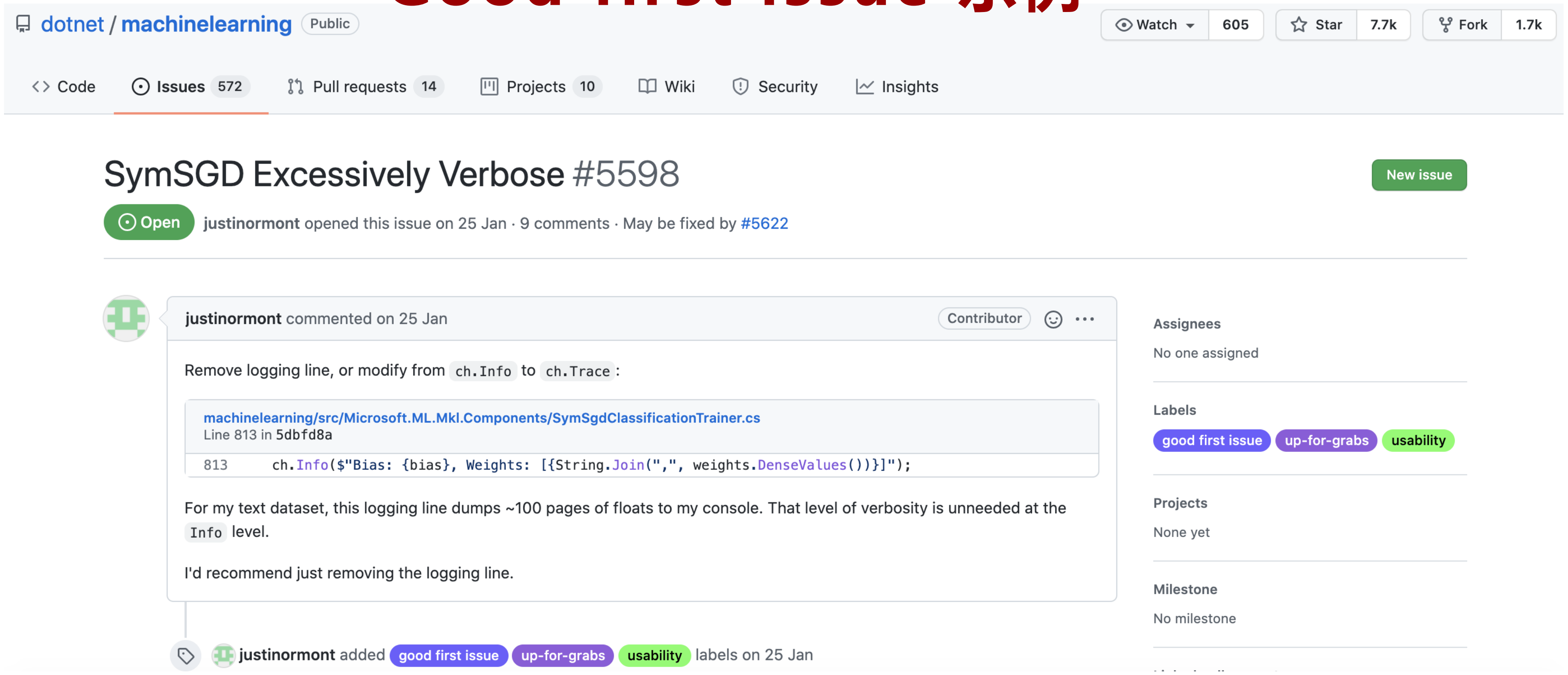
Add "Hindi" and "Marathi" translation for App strings
#1014 opened on 6 Jul by MSid01

Filter by label
Filter labels
Unlabeled
Beginner
bug
done
duplicate
easy
enhancement
good first issue

## 3、挑选出准备解决的issue，明确需要解决的任务

- 从issue标题和描述明确该issue希望解决的问题

**Good first issue 示例一**



dotnet / machinelearning Public

Watch 605 · Star 7.7k · Fork 1.7k

Code · Issues 572 · Pull requests 14 · Projects 10 · Wiki · Security · Insights

SymSGD Excessively Verbose #5598

Open · justinormont opened this issue on 25 Jan · 9 comments · May be fixed by #5622

justinormont commented on 25 Jan · Contributor

Remove logging line, or modify from `ch.Info` to `ch.Trace`:

machinelearning/src/Microsoft.ML.Mkl.Components/SymSgdClassificationTrainer.cs
Line 813 in 5dbfd8a

```
ch.Info($"Bias: {bias}, Weights: [{String.Join(",", weights.DenseValues())}]");
```

For my text dataset, this logging line dumps ~100 pages of floats to my console. That level of verbosity is unneeded at the `Info` level.

I'd recommend just removing the logging line.

justinormont added good first issue up-for-grabs usability labels on 25 Jan

Assignees: No one assigned

Labels: good first issue, up-for-grabs, usability

Projects: None yet

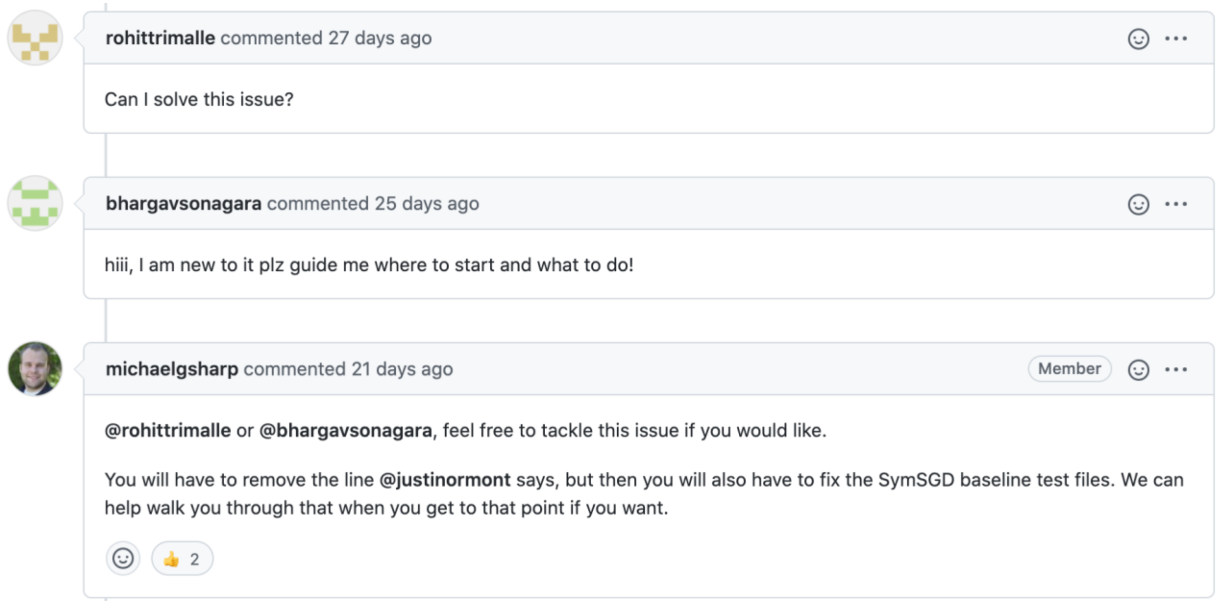
Milestone: No milestone

根据issue描述，该issue对应的任务是删除一行代码。

https://github.com/dotnet/machinelearning/issues/5598

## Good first issue 示例一

- 从issue标题和描述明确该issue希望解决的问题；
- 可以通过评论向项目成员寻求帮助。


**rohittrimalle** commented 27 days ago

Can I solve this issue?

**bhargavsonagara** commented 25 days ago

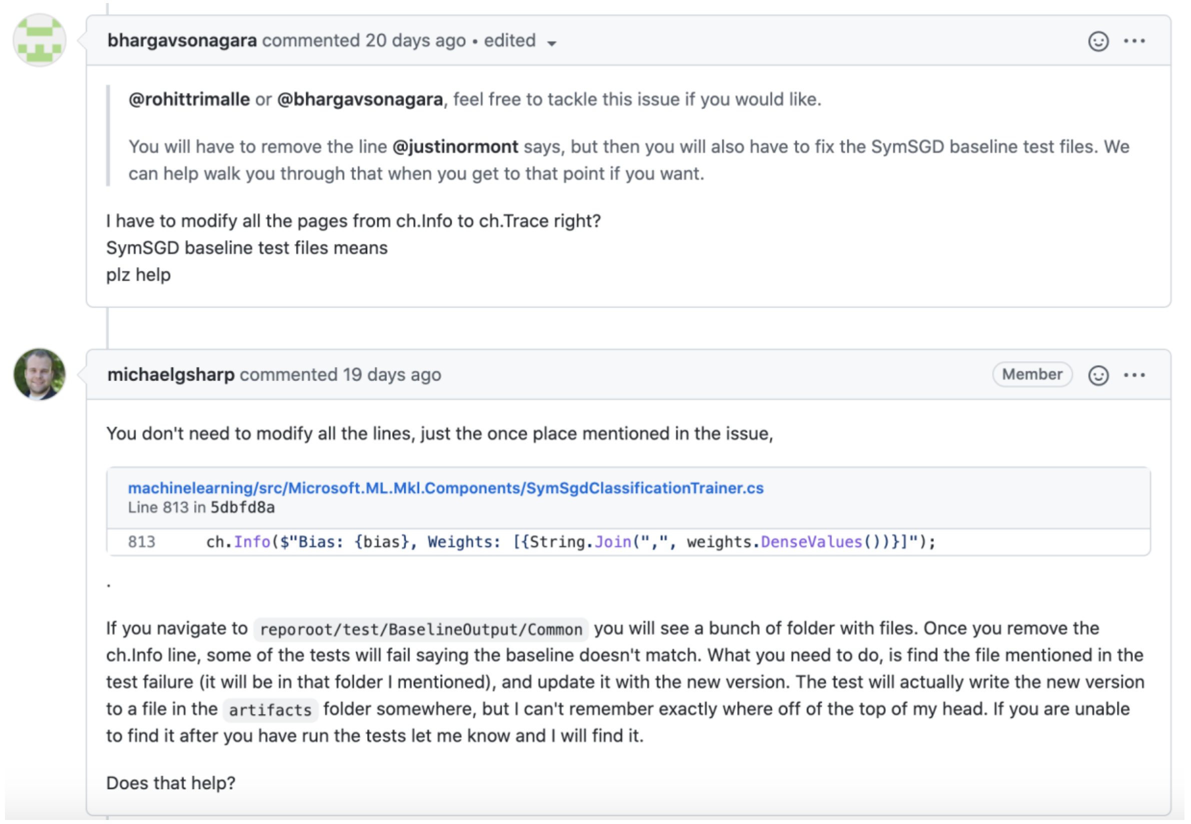
hiii, I am new to it plz guide me where to start and what to do!

**michaelgsharp** commented 21 days ago Member

**@rohittrimalle** or **@bhargavsonagara**, feel free to tackle this issue if you would like.

You will have to remove the line **@justinormont** says, but then you will also have to fix the SymSGD baseline test files. We can help walk you through that when you get to that point if you want.

👍 2


**bhargavsonagara** commented 20 days ago • edited

> **@rohittrimalle** or **@bhargavsonagara**, feel free to tackle this issue if you would like.
>
> You will have to remove the line **@justinormont** says, but then you will also have to fix the SymSGD baseline test files. We can help walk you through that when you get to that point if you want.

I have to modify all the pages from ch.Info to ch.Trace right?
SymSGD baseline test files means
plz help

**michaelgsharp** commented 19 days ago Member

You don't need to modify all the lines, just the once place mentioned in the issue,

machinelearning/src/Microsoft.ML.Mkl.Components/SymSgdClassificationTrainer.cs
Line 813 in 5dbfd8a

```
813        ch.Info($"Bias: {bias}, Weights: [{String.Join(",", weights.DenseValues())}]");
```

.

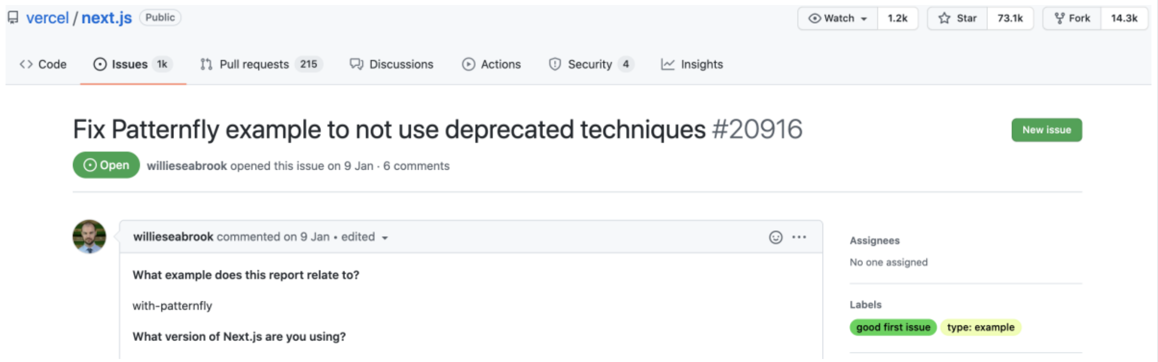
If you navigate to `reporoot/test/BaselineOutput/Common` you will see a bunch of folder with files. Once you remove the ch.Info line, some of the tests will fail saying the baseline doesn't match. What you need to do, is find the file mentioned in the test failure (it will be in that folder I mentioned), and update it with the new version. The test will actually write the new version to a file in the `artifacts` folder somewhere, but I can't remember exactly where off of the top of my head. If you are unable to find it after you have run the tests let me know and I will find it.

Does that help?

由项目成员的补充说明可知，该issue的任务不仅是issue描述中所述的删除一行代码，还包括修改对应的测试代码。

# Good first issue 示例二

- 根据issue的标题和描述，该bug为项目给出的example中使用了被弃用的模块。



Fix Patternfly example to not use deprecated techniques #20916

Open willieseabrook opened this issue on 9 Jan · 6 comments

willieseabrook commented on 9 Jan • edited

**What example does this report relate to?**

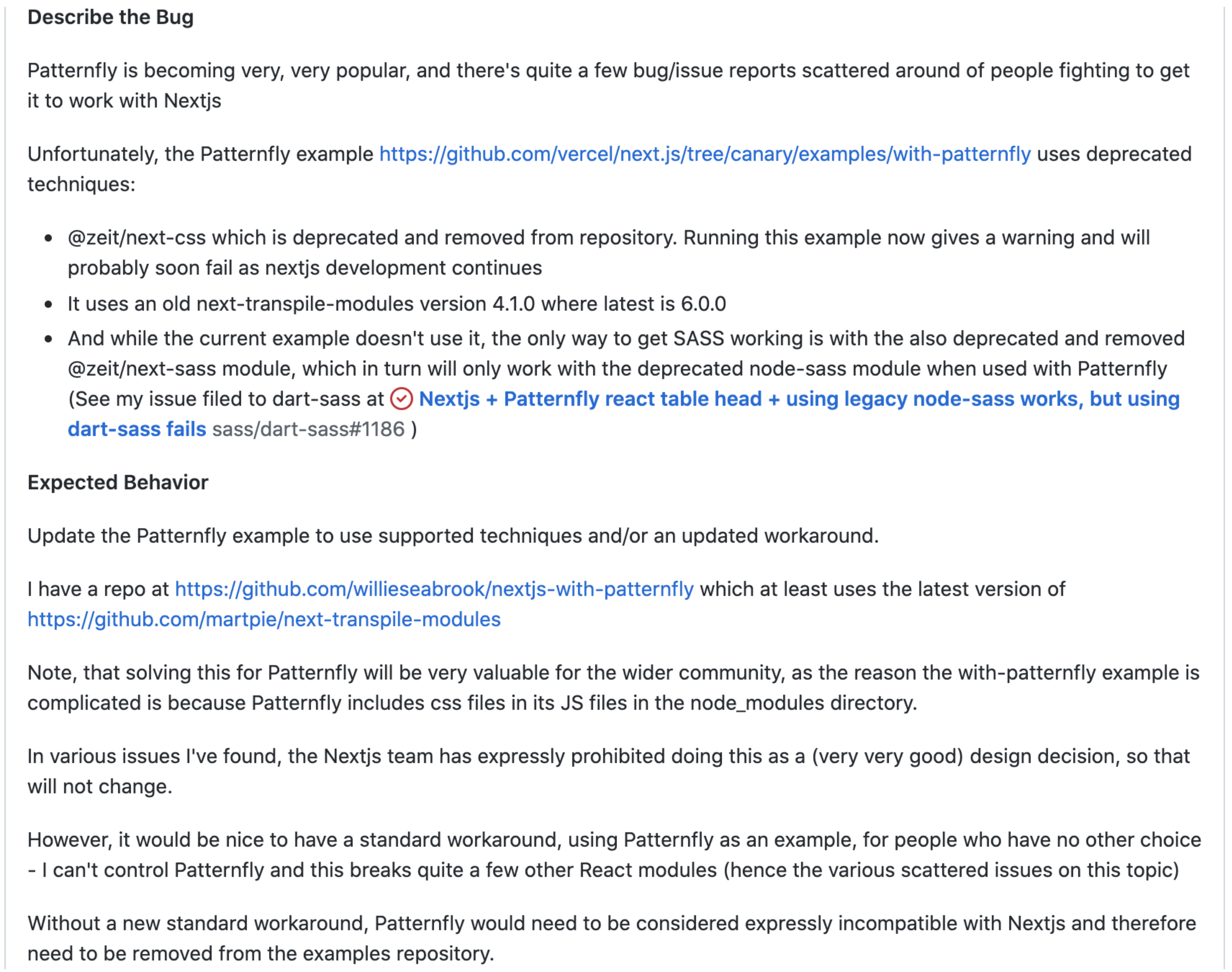
with-patternfly

**What version of Next.js are you using?**

Assignees: No one assigned

Labels: good first issue, type: example

https://github.com/vercel/next.js/issues/20916



**Describe the Bug**

Patternfly is becoming very, very popular, and there's quite a few bug/issue reports scattered around of people fighting to get it to work with Nextjs

Unfortunately, the Patternfly example https://github.com/vercel/next.js/tree/canary/examples/with-patternfly uses deprecated techniques:

- @zeit/next-css which is deprecated and removed from repository. Running this example now gives a warning and will probably soon fail as nextjs development continues
- It uses an old next-transpile-modules version 4.1.0 where latest is 6.0.0
- And while the current example doesn't use it, the only way to get SASS working is with the also deprecated and removed @zeit/next-sass module, which in turn will only work with the deprecated node-sass module when used with Patternfly (See my issue filed to dart-sass at **Nextjs + Patternfly react table head + using legacy node-sass works, but using dart-sass fails** sass/dart-sass#1186 )

**Expected Behavior**

Update the Patternfly example to use supported techniques and/or an updated workaround.

I have a repo at https://github.com/willieseabrook/nextjs-with-patternfly which at least uses the latest version of https://github.com/martpie/next-transpile-modules

Note, that solving this for Patternfly will be very valuable for the wider community, as the reason the with-patternfly example is complicated is because Patternfly includes css files in its JS files in the node_modules directory.

In various issues I've found, the Nextjs team has expressly prohibited doing this as a (very very good) design decision, so that will not change.

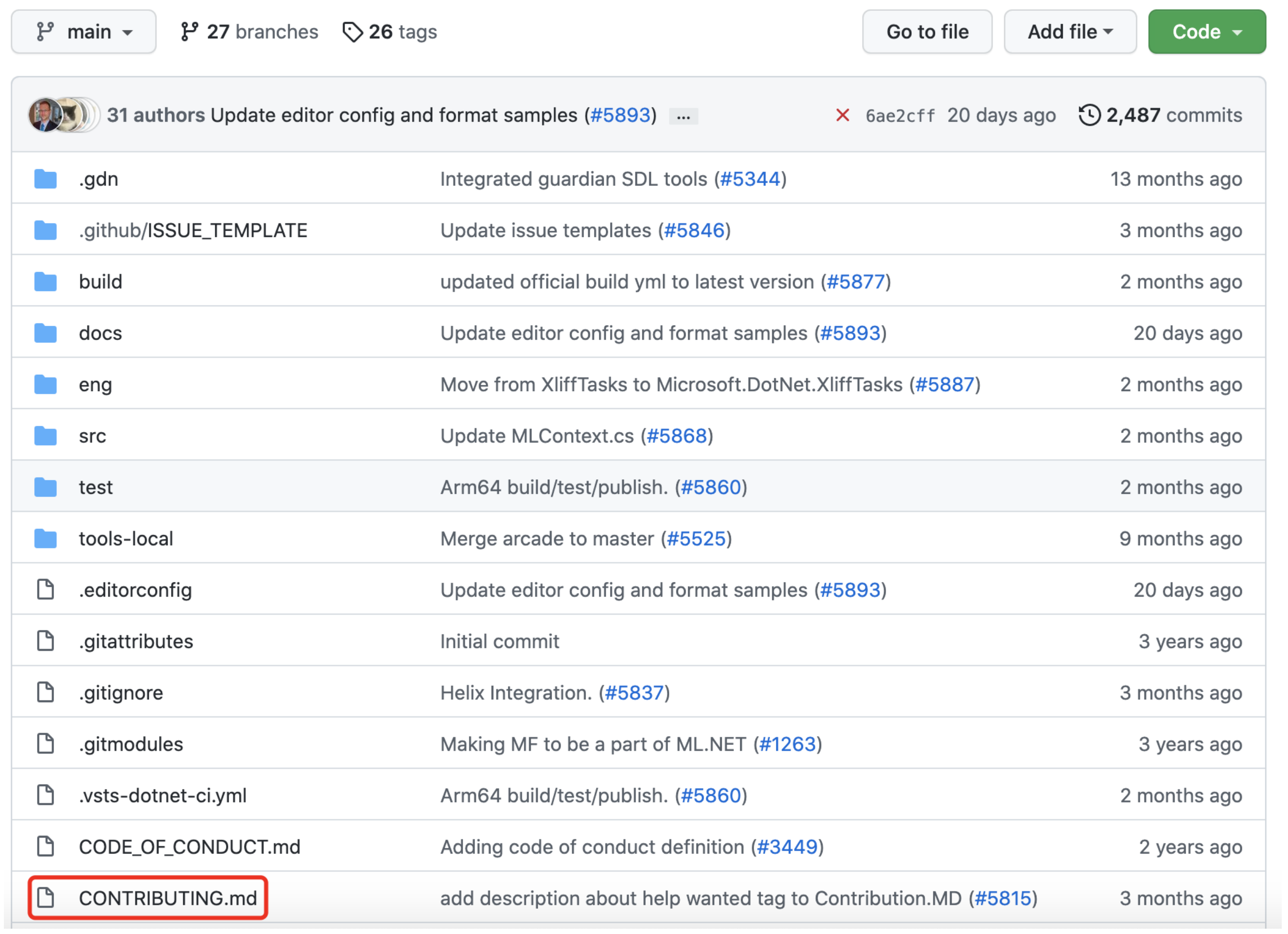
However, it would be nice to have a standard workaround, using Patternfly as an example, for people who have no other choice - I can't control Patternfly and this breaks quite a few other React modules (hence the various scattered issues on this topic)

Without a new standard workaround, Patternfly would need to be considered expressly incompatible with Nextjs and therefore need to be removed from the examples repository.

## 4. 以符合项目贡献协议的方式提交贡献

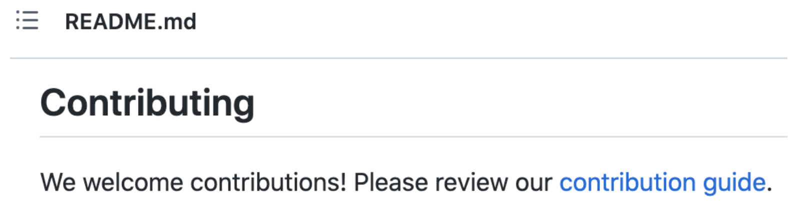
- 规范的项目通常会对贡献的形式有所要求(一般在contributing.md中)，提交贡献时务必遵守这些要求。

## Good first issue 示例一



31 authors Update editor config and format samples (#5893) ... 6ae2cff 20 days ago 2,487 commits

| | | |
|---|---|---|
| .gdn | Integrated guardian SDL tools (#5344) | 13 months ago |
| .github/ISSUE_TEMPLATE | Update issue templates (#5846) | 3 months ago |
| build | updated official build yml to latest version (#5877) | 2 months ago |
| docs | Update editor config and format samples (#5893) | 20 days ago |
| eng | Move from XliffTasks to Microsoft.DotNet.XliffTasks (#5887) | 2 months ago |
| src | Update MLContext.cs (#5868) | 2 months ago |
| test | Arm64 build/test/publish. (#5860) | 2 months ago |
| tools-local | Merge arcade to master (#5525) | 9 months ago |
| .editorconfig | Update editor config and format samples (#5893) | 20 days ago |
| .gitattributes | Initial commit | 3 years ago |
| .gitignore | Helix Integration. (#5837) | 3 months ago |
| .gitmodules | Making MF to be a part of ML.NET (#1263) | 3 years ago |
| .vsts-dotnet-ci.yml | Arm64 build/test/publish. (#5860) | 2 months ago |
| CODE_OF_CONDUCT.md | Adding code of conduct definition (#3449) | 2 years ago |
| CONTRIBUTING.md | add description about help wanted tag to Contribution.MD (#5815) | 3 months ago |

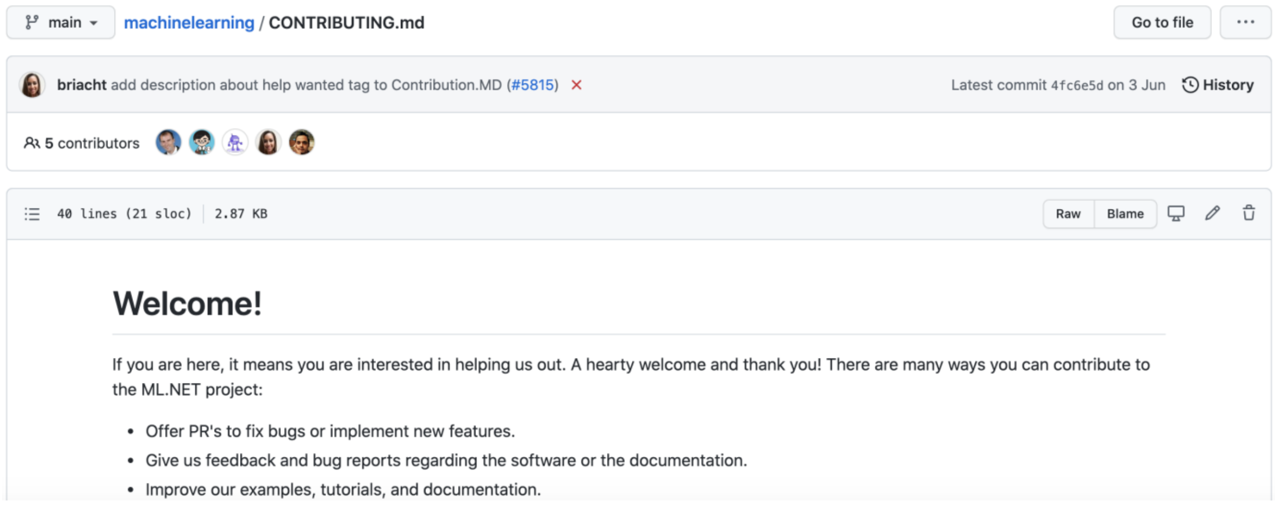

README.md

### Contributing

We welcome contributions! Please review our contribution guide.

# Good first issue 示例一

Issue所在项目的CONTRIBUTING.md



main · machinelearning / CONTRIBUTING.md · Go to file

briacht add description about help wanted tag to Contribution.MD (#5815) ✗ Latest commit 4fc6e5d on 3 Jun · History

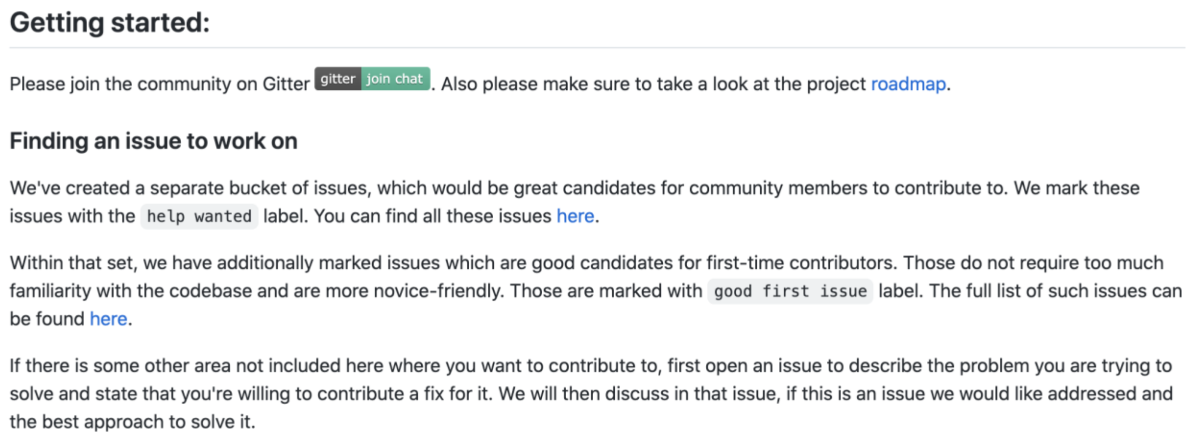
5 contributors

40 lines (21 sloc) | 2.87 KB · Raw · Blame

## Welcome!

If you are here, it means you are interested in helping us out. A hearty welcome and thank you! There are many ways you can contribute to the ML.NET project:

- Offer PR's to fix bugs or implement new features.
- Give us feedback and bug reports regarding the software or the documentation.
- Improve our examples, tutorials, and documentation.



## Getting started:

Please join the community on Gitter gitter join chat. Also please make sure to take a look at the project roadmap.

### Finding an issue to work on

We've created a separate bucket of issues, which would be great candidates for community members to contribute to. We mark these issues with the `help wanted` label. You can find all these issues here.

Within that set, we have additionally marked issues which are good candidates for first-time contributors. Those do not require too much familiarity with the codebase and are more novice-friendly. Those are marked with `good first issue` label. The full list of such issues can be found here.

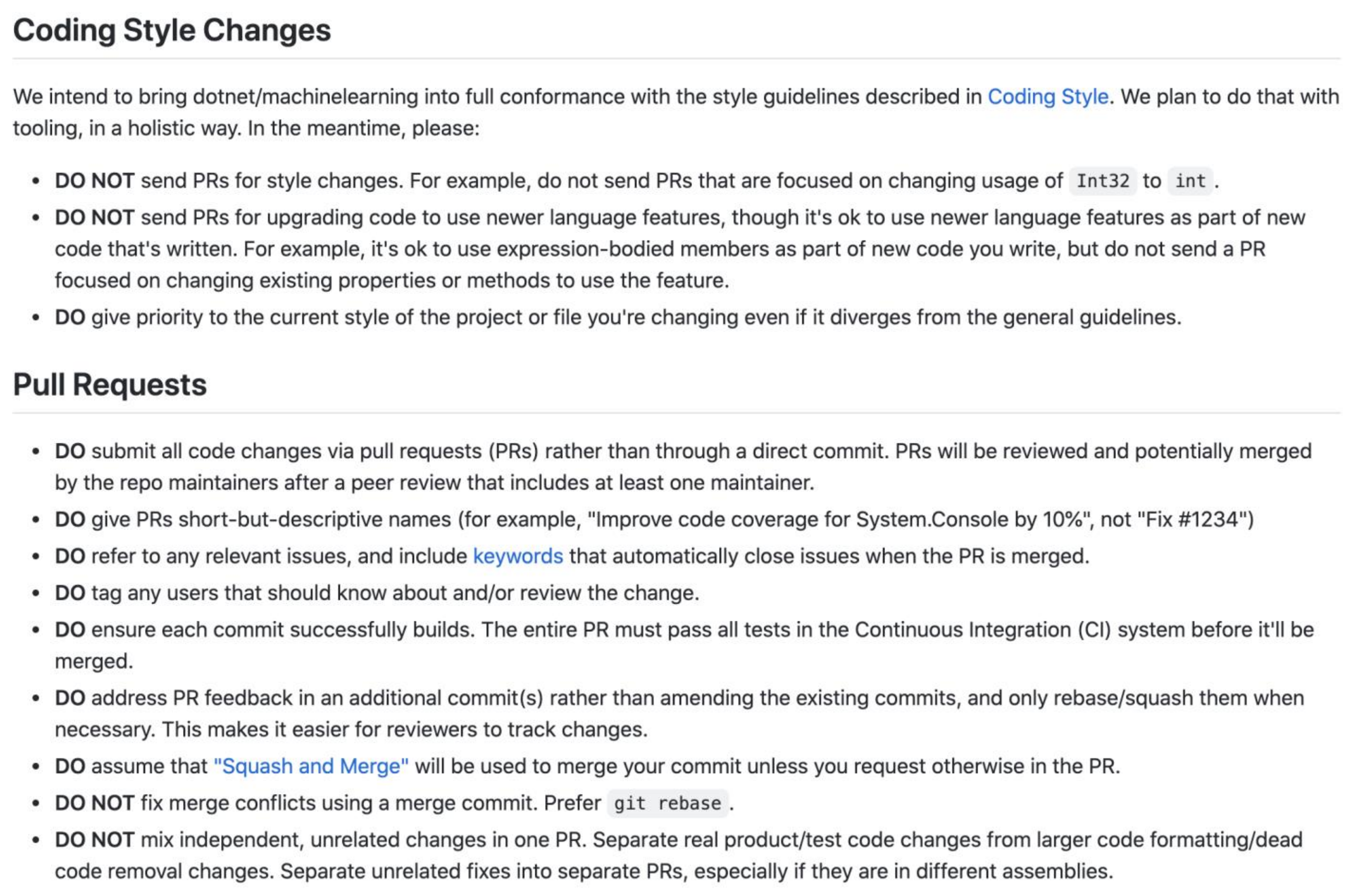
If there is some other area not included here where you want to contribute to, first open an issue to describe the problem you are trying to solve and state that you're willing to contribute a fix for it. We will then discuss in that issue, if this is an issue we would like addressed and the best approach to solve it.

列出为项目做贡献的方式

介绍如何从issues中寻找任务

# Good first issue 示例一

Issue所在项目的CONTRIBUTING.md



## Coding Style Changes

We intend to bring dotnet/machinelearning into full conformance with the style guidelines described in Coding Style. We plan to do that with tooling, in a holistic way. In the meantime, please:

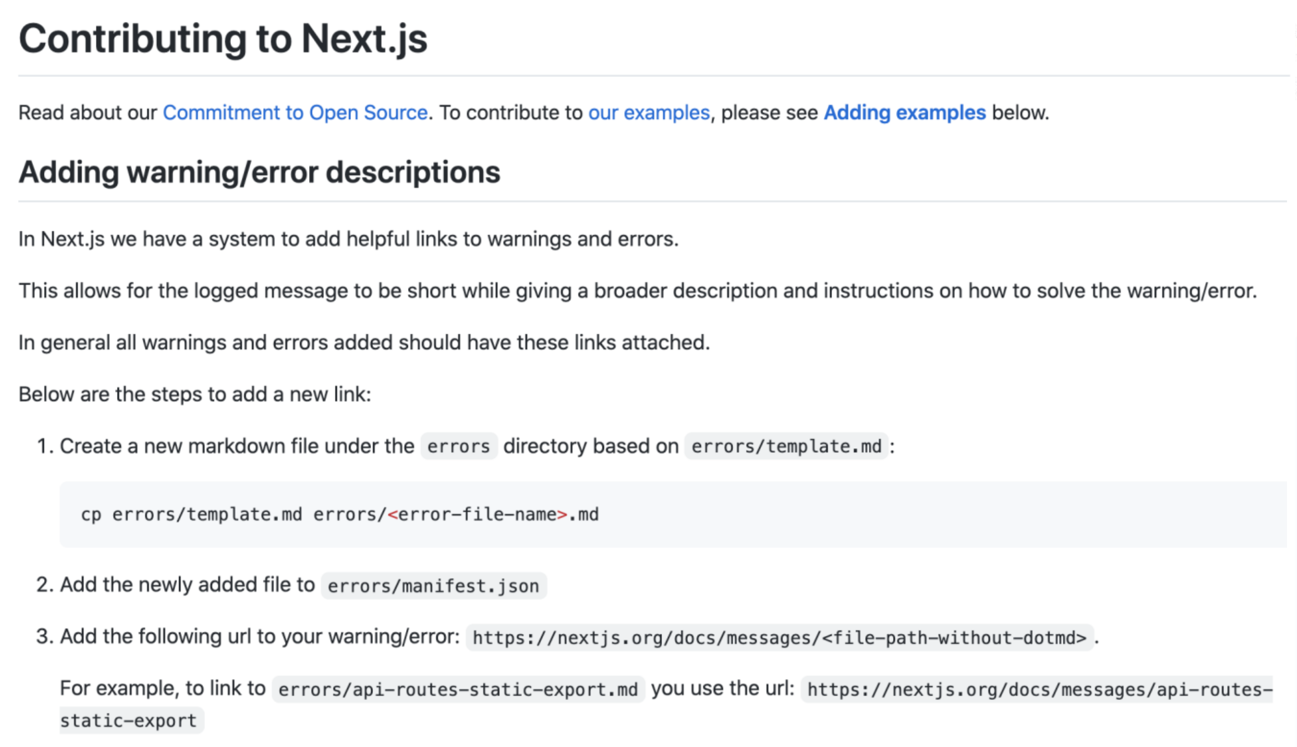
- **DO NOT** send PRs for style changes. For example, do not send PRs that are focused on changing usage of `Int32` to `int`.
- **DO NOT** send PRs for upgrading code to use newer language features, though it's ok to use newer language features as part of new code that's written. For example, it's ok to use expression-bodied members as part of new code you write, but do not send a PR focused on changing existing properties or methods to use the feature.
- **DO** give priority to the current style of the project or file you're changing even if it diverges from the general guidelines.

## Pull Requests

- **DO** submit all code changes via pull requests (PRs) rather than through a direct commit. PRs will be reviewed and potentially merged by the repo maintainers after a peer review that includes at least one maintainer.
- **DO** give PRs short-but-descriptive names (for example, "Improve code coverage for System.Console by 10%", not "Fix #1234")
- **DO** refer to any relevant issues, and include keywords that automatically close issues when the PR is merged.
- **DO** tag any users that should know about and/or review the change.
- **DO** ensure each commit successfully builds. The entire PR must pass all tests in the Continuous Integration (CI) system before it'll be merged.
- **DO** address PR feedback in an additional commit(s) rather than amending the existing commits, and only rebase/squash them when necessary. This makes it easier for reviewers to track changes.
- **DO** assume that "Squash and Merge" will be used to merge your commit unless you request otherwise in the PR.
- **DO NOT** fix merge conflicts using a merge commit. Prefer `git rebase`.
- **DO NOT** mix independent, unrelated changes in one PR. Separate real product/test code changes from larger code formatting/dead code removal changes. Separate unrelated fixes into separate PRs, especially if they are in different assemblies.

关于代码风格和pull requests的规范

# Good first issue 示例二



## Contributing to Next.js

Read about our Commitment to Open Source. To contribute to our examples, please see **Adding examples** below.

### Adding warning/error descriptions

In Next.js we have a system to add helpful links to warnings and errors.

This allows for the logged message to be short while giving a broader description and instructions on how to solve the warning/error.

In general all warnings and errors added should have these links attached.

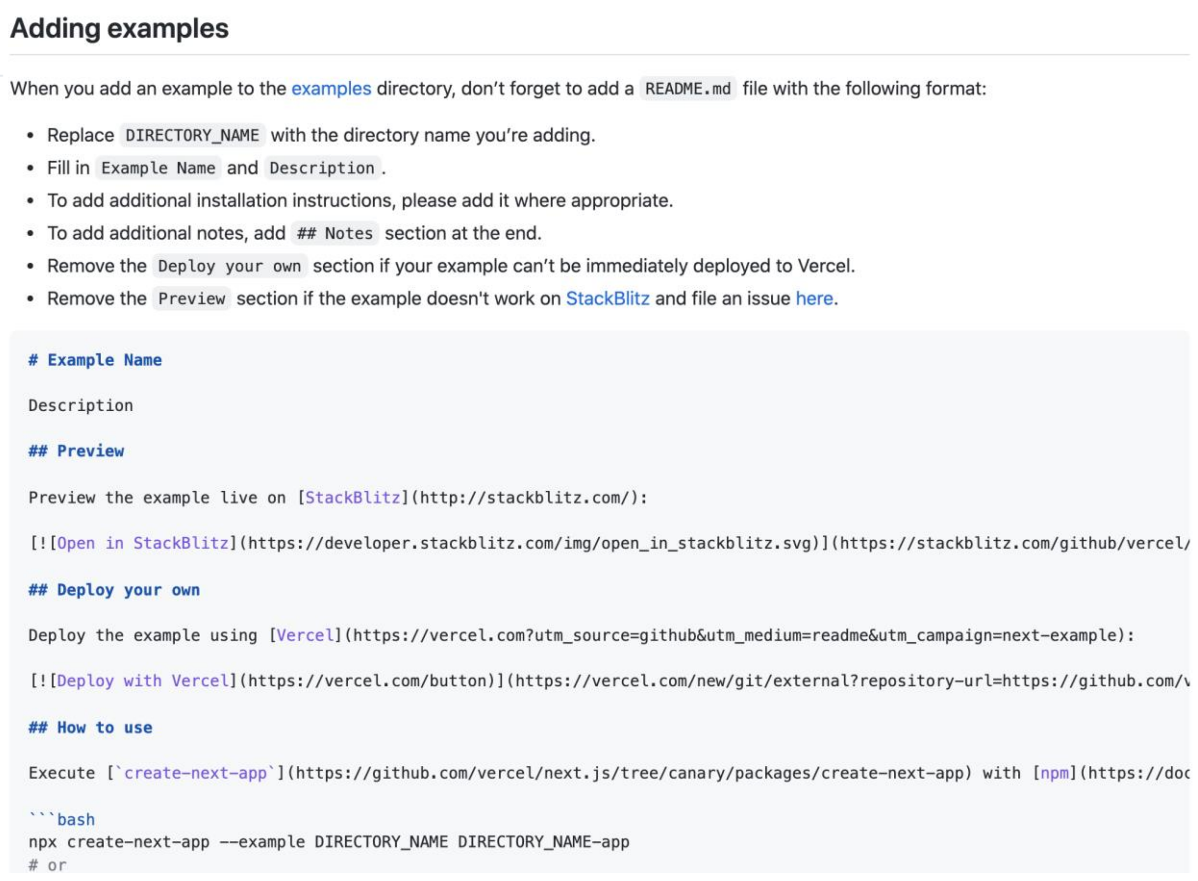
Below are the steps to add a new link:

1. Create a new markdown file under the `errors` directory based on `errors/template.md`:

   ```
   cp errors/template.md errors/<error-file-name>.md
   ```

2. Add the newly added file to `errors/manifest.json`
3. Add the following url to your warning/error: `https://nextjs.org/docs/messages/<file-path-without-dotmd>`.

   For example, to link to `errors/api-routes-static-export.md` you use the url: `https://nextjs.org/docs/messages/api-routes-static-export`



## Adding examples

When you add an example to the examples directory, don't forget to add a `README.md` file with the following format:

- Replace `DIRECTORY_NAME` with the directory name you're adding.
- Fill in `Example Name` and `Description`.
- To add additional installation instructions, please add it where appropriate.
- To add additional notes, add `## Notes` section at the end.
- Remove the `Deploy your own` section if your example can't be immediately deployed to Vercel.
- Remove the `Preview` section if the example doesn't work on StackBlitz and file an issue here.

~~~
# Example Name

Description

## Preview

Preview the example live on [StackBlitz](http://stackblitz.com/):

[![Open in StackBlitz](https://developer.stackblitz.com/img/open_in_stackblitz.svg)](https://stackblitz.com/github/vercel/

## Deploy your own

Deploy the example using [Vercel](https://vercel.com?utm_source=github&utm_medium=readme&utm_campaign=next-example):

[![Deploy with Vercel](https://vercel.com/button)](https://vercel.com/new/git/external?repository-url=https://github.com/v

## How to use

Execute [`create-next-app`](https://github.com/vercel/next.js/tree/canary/packages/create-next-app) with [npm](https://doc

```bash
npx create-next-app --example DIRECTORY_NAME DIRECTORY_NAME-app
# or
~~~

项目的contributing.md文档具体说明了提交的贡献需遵循的格式

# End