

---

**LizardByte**

**ReenigneArcher**

**Feb 04, 2023**



# ABOUT

<b>1</b>	<b>Overview</b>	<b>1</b>
1.1	About . . . . .	1
<b>2</b>	<b>Support</b>	<b>3</b>
<b>3</b>	<b>Donate</b>	<b>5</b>
<b>4</b>	<b>Contributing</b>	<b>7</b>
4.1	Legal . . . . .	8
<b>5</b>	<b>Automations</b>	<b>9</b>
5.1	Code Style . . . . .	9
5.2	Tests . . . . .	9
5.3	CI . . . . .	9



## OVERVIEW

LizardByte has the full documentation hosted on [Read the Docs](#).

### 1.1 About

LizardByte is developing self hosted cloud game streaming solutions. These applications are developed by volunteers in their free time. If you like our products please consider donating!



## SUPPORT

**Warning:** We use GitHub issues exclusively for actionable bug reports. Low effort issues will be immediately closed. Spamming the GitHub repo or organization will result in a ban.

Our available support methods are listed in our [Support Center](#).





---

CHAPTER  
**THREE**

---

**DONATE**

Like our projects and want to support our development work? These are the ways you can donate to our projects.

---

**Note:** MEE6 and Patreon provide Discord benefits.

---



## CONTRIBUTING

---

**Tip:** If this is your first time contributing to an open source project, it is a good idea to read MDN's [Basic etiquette for open source projects](#) first. There are a few best practices to adopt that will help ensure that you and the other project contributors feel valued and safe, and stay productive.

---

1. Fork the repo on GitHub
2. Create a new branch for the feature you are adding or the issue you are fixing

---

**Tip:** Base the new branch off the *nightly* branch. It will make your life easier when you submit the PR!

---

3. Make changes, push commits, etc.
4. Files should contain an empty line at the end.
5. Document your code!
6. Test your code!
7. When ready create a PR against the *nightly* branch.

---

**Hint:** If you accidentally make your PR against a different branch, a bot will comment letting you know it's on the wrong branch. Don't worry. You can edit the PR to change the target branch. There is no reason to close the PR!

---

---

**Note:** Draft PRs are also welcome as you work through issues. The benefit of creating a draft PR is that an automated build can run in a github runner.

---

**Attention:** Do not expect partially complete PRs to be merged. These topics will be considered before merging.

- **Does the code follow the style guidelines of the project?**

---

**Tip:** Look at examples of existing code in the project!

---

- Is the change value added?
- Is the code well commented?

- Were documentation blocks updated for new or modified components?

---

**Note:** Developers and maintainers will attempt to assist with challenging issues.

---

## 4.1 Legal

You must sign our CLA (Contributor License Agreement) before we merge your PR. A bot will comment on your PR if action is necessary.

- [CLA - individual agreement](#)
- [CLA - entity agreement](#)

**Attention:** Follow the instructions provided by the bot to complete signing the contributor license agreement.

**Instructions for if you do not own the Copyright in the entire work of authorship Submitted.**

1. Add the owner as a *co-author* to a commit in the PR. See [Creating a commit with multiple authors](#).
2. All authors must sign the CLA before it can be merged.

## AUTOMATIONS

Our repositories have many automations that help us have consistent and high quality code.

---

**Note:** Some of our first repos are lacking in this, but we are working to improve them.

---

Have suggestions for additional checks for a project? Reach out to us and let us know.

### 5.1 Code Style

We enforce code style guidelines on our projects. The checks that apply depend on the labels of the repository. These are standard checks which are copied regularly from our `.github` repo to each repo.

For example, repositories with the `python` label will have `flake8` tests.

### 5.2 Tests

We aim to have additional tests for each project, but not every project has this available. We would welcome contributions that add to or improve tests.

### 5.3 CI

Generally, our repositories have continuous integrations setup. These take care of building the project as needed, as well as automatically publishing releases.