# Contributing

balenaSound was created as a [hack friday](https://www.balena.io/blog/hack-friday-september/) project and has since grown to be a fully featured product with multiple community contributions. If you are interested in contributing to balenaSound this document will help you get started.

## Overview

Make sure you checkout our [architecture][ARCHITECTURE] guide, which aims to explain how all pieces fit together. It's a good starting point for understanding how we designed and built balenaSound.

## Commit guidelines

We enforce certain rules on commits with the following goals in mind:

- Be able to reliably auto-generate the `CHANGELOG.md` \*without\* any human intervention.

- Be able to automatically and correctly increment the semver version number based on what was done since the last release.

- Be able to get a quick overview of what happened to the project by glancing over the commit history.

- Be able to automatically reference relevant changes from a dependency upgrade.

Our CI will run checks to ensure this guidelines are followed and won't allow merging contributions that don't adhere to them. Version number and changelog are automatically handled by the CI build flow after a pull request is merged. You only need to worry about the commit itself.

### Commit structure

Each commit message should consist of a header a body and a footer, structured in the following format:

```

<scope (optional)>: <subject (mandatory)>

--BLANK LINE--

(optional) <body>

--BLANK LINE--

(optional) Connects-to: #issue-number

(mandatory) Change-type: major | minor | patch

(optional) Signed-off-by: Foo Bar <foobar@balena.io>

```

Note that:

- Blank lines are required to separate header from body and body from footer. You don't need to add two blank lines if you don't add a body.

- `scope`: If your commit touches a well defined component/part/service please addthe scope tag to clarify. Some examples: `docs`, `airplay`, `multi-room`.

- `subject`: The subject should contain a short description of the change. Use the imperative, present tense.

- `body`: A detailed description of changes being made and reasoning if necessary. This may contain several paragraphs.

- `Connects-to`: If your commit fixes or is connected to an existing issue, link it by adding this tag with `#issue-number`. Example: `Connects-to: #123`

- `Change-type`: At least one of your commits on a PR needs to have this tag. You have the flexibility, and it's good practise, to use this tag in as many commits as you see fit; in the end, the resulting change type for the scope of the PR will be folded down to the biggest one as marked in the commits (`major>minor>patch`). Our version numbering adheres to [Semantic Versioning](http://semver.org/).

- `Signed-off-by`: Sign your commits by providing your full name and email address in the format: `Name Surname <email@something.com>`. \*This is an optional tag.\*

### Commit examples

Here are some examples of valid commits:

\*\*Big new feature\*\*

```

multi-room: Add multi-room feature

This feature adds multi-room audio streaming to balenaSound.

No breaking changes were made, but considering this a major version bump since it's a big feature and all services were affected.

New services added:

- snapcast-server

- snapcast-client

- fleet-supervisor

Other changes:

- By default, all audio services now stream to a fifo pipe file instead of alsa backend.

- Multi-room can be disabled via env var DISABLE\_MULTI\_ROOM.

Change-type: major

Signed-off-by: Tomás Migone <tomas@balena.io>

```

\*\*Simple change\*\*

```

Remove mplayer, use WAV notification sounds

Change-type: patch

Signed-off-by: Chris Crocker-White <chriscw@balena.io>

```

\*\*Fix an issue\*\*

```

Fix spotify password error if it has spaces

Change-type: patch

Connects-to: #90

```

## PR guidelines

Pull requests are the only way of pushing your code to the `master` branch. When creating a PR make sure you choose a short but sensical PR title and add a few lines describing your contribution.

### PR approval

The PR will only be able to be merged only after:

- It has been approved at least by one [codeowner](https://github.com/balenalabs/balena-sound/blob/master/.github/CODEOWNERS)

- All the checks and tests carried out by our CI are passed

### Commit squashing

If your PR contains multiple commits you might be asked to rebase your PR branch on top of the latest `master` and squash your commits before merging. This can be achieved with the following steps, assuming that the current branch is that to be merged to master in your local file system:

```

git checkout master

git pull

git checkout your-branch

git rebase -i master

```

At this point `git rebase` will prompt you to choose an action for each commit and resolve any conflicts. You should `pick` one commit, `reword` it if necessary and `squash` the rest. The reason behind using rebase is that it makes for tidier git branching history. Push the rebased PR branch to the remote and you should be good to go:

```

git push --force-with-lease origin your-branch

```

We strongly encourage using the `--force-with-lease` option instead of `--force` when performing git push to a repository. The reason is that `git push --force` can accidentally overwrite work that has been pushed by a team member in the meantime.

## balena-ci bot

If for some reason it looks like the checks and tests for the PR have failed, add a comment to the PR with `@balena-ci retest`. This should force balenaCI to re-run all the tests again.