# Contributing to the Reactive Streams Project

The Reactive Streams project welcomes contributions from anybody who wants to participate in moving this initiative forward. All code or documentation that is contributed will have to be covered by a waiver of all copyrights and other rights as detailed by the LICENSE and COPYING files at each repository root, the rationale for this is that the APIs defined by this project shall be freely implementable and usable by everyone.

## Copyright Statement

The aforementioned waiver of copyrights and other rights is represented by the addition of a line to the file [CopyrightWaivers.txt](https://github.com/reactive-streams/reactive-streams-jvm/blob/master/CopyrightWaivers.txt). For a pull request to be considered every contributor must have signed the copyright statement in this way; this may be included within that same pull request.

## Gatekeepers

To ensure consistent development of Reactive Streams towards their goal, a group of gatekeepers is defined:

\* Kaazing Corp., currently represented by Todd Montgomery (@tmontgomery)

\* Netflix Inc., currently represented by Ben Christensen (@benjchristensen)

\* Pivotal Software Inc., currently represented by Jon Brisbin (@jbrisbin) and Stéphane Maldini (@smaldini)

\* Red Hat Inc., currently represented by Tim Fox (@purplefox) and Norman Maurer (@normanmaurer)

\* Twitter Inc., currently represented by Marius Eriksen (@mariusaeriksen)

\* Typesafe Inc., currently represented by Viktor Klang (@viktorklang) and Roland Kuhn (@rkuhn)

The role of this group is detailed in the following, additions to this list are made by pull request as defined below, removals require the consent of the entity to be removed or unanimous consent of all other Gatekeepers. Changing a representative of one of the gatekeeper entities can be done by a member of that entity without requiring consent from the other Gatekeepers.

Gatekeepers commit to the following:

1. 1-week SLA on :+1: or :-1: Pull Requests

\* If a Gatekeeper will be unavailable for a period of time, notify @reactive-streams/contributors and appoint who will vote in his/her place in the mean time

2. tag @reactive-streams/contributors with a deadline when there needs to be a vote on an Issue,

with at least 1 week of notice (see rule 1 above)

## General Workflow

1. Make sure you have signed the Copyright Statement, see above.

2. Before starting to work on a change, make sure that:

1. There is a ticket for your work in the project's issue tracker. If not, create it first. It can help accelerating the pull request acceptance process if the change is agreed upon beforehand within the ticket, but in some cases it may be preferable to discuss the necessity of the change in consideration of a concrete proposal.

2. The ticket has been scheduled for the current milestone.

3. You should always perform your work in a Git feature branch within your own fork of the repository your are targeting (even if you should have push rights to the target repository).

4. When the change is completed you should open a [Pull Request](https://help.github.com/articles/using-pull-requests) on GitHub.

5. Anyone can comment on the pull request while it is open, and you are expected to answer questions or incorporate feedback.

6. Once at least two thirds of the gatekeepers have signaled their consent, the pull request is merged by one of the gatekeepers into the branch and repository it is targeting. Consent is signaled by commenting on the pull request with the text “LGTM”, and it suffices for one representative of a gatekeeper to signal consent for that gatekeeper to be counted towards the two thirds quorum.

7. It is not allowed to force-push to the branch on which the pull request is based. Replacing or adding to the commits on that branch will invalidate all previous consenting comments and consent needs to be re-established.

8. Before merging a branch that contains multiple commits, it is recommended that these are squashed into a single commit before performing the merge. To aid in verification that no new changes are introduced, a new pull request should be opened in this case, targeting the same branch and repository and containing just one commit which encompasses the full change set of the original pull request.

## Pull Request Requirements

For a Pull Request to be considered at all it has to meet these requirements:

1. If applicable, the new or fixed features must be accompanied by comprehensive tests.

2. If applicable, the pull request must contain all necessary documentation updates required by the changes introduced.

3. The pull request must not contain changes that are unrelated to the ticket that it corresponds to. One pull request is meant to introduce only one logically contiguous change.

## Creating Commits And Writing Commit Messages

Follow these guidelines when creating public commits and writing commit messages.

1. If your work spans multiple local commits (for example; if you do safe point commits while working in a feature branch or work in a branch for long time doing merges/rebases etc.) then please do not commit it all but rewrite the history by squashing the commits into a single big commit which you write a good commit message for (like discussed in the following sections). For more info read this article: [Git Workflow](http://sandofsky.com/blog/git-workflow.html). Every commit should be able to be used in isolation, cherry picked etc.

2. First line should be a descriptive sentence what the commit is doing. It should be possible to fully understand what the commit does—but not necessarily how it does it—by just reading this single line. We follow the “imperative present tense” style for commit messages ([more info here](http://tbaggery.com/2008/04/19/a-note-about-git-commit-messages.html)).

It is \*\*not ok\*\* to only list the ticket number, type "minor fix" or similar. In order to help with automatic filtering of the commit history (generating ChangeLogs, writing the migration guide, code archaeology) we use the following encoding:

3. Following the single line description should be a blank line followed by an enumerated list with the details of the commit. For very simple commits this may be empty.

4. Add keywords for your commit:

\* ``Review by @gituser`` - if you want to notify someone specifically for review; this has no influence on the acceptance process described above

Example:

add CONTRIBUTING.md

\* clarify how pull requests should look like

\* describe the acceptance process

\* define the Copyright Statement signing process

## Performing Official Releases

Creating binary artifacts, uploading them to central repositories and declaring these to be an official release of the Reactive Streams project requires the consent of all gatekeepers. The process is initiated by creating a ticket in the `reactive-streams` repository for this purpose and consent is signaled in the same way as for pull requests. The actual work of updating version numbers and publishing the artifacts will typically involve pull requests targeting the affected repositories.