# Contributing

## Pull requests are welcome

Feel free to open a pull request. Just be aware that it might take some time for me to review your request.

#### How to make a good pull request

- Make sure you do not break the API.

- Make sure the tests succeed.

- If you added functionality, add a test case if possible.

- If you added a function or modified the API, change the documentation accordingly.

- Don't change the `package.json` or `package-lock.json` files to bump the version number of this library.

- Feel free to change the `CHANGELOG.md` to include your change. You don't need to add a date though. I'll do it, when I publish the next version.

Thank you very much for contributing :)

## Release Process

\*\*There is no need to read the following guide if you want to make a pull request. This is basically my TODO list for publishing.\*\*

Releases are automatically published to npm by Travis CI. To successfully create a release the following preconditions have to be met:

- The commit has to be on the master.

- The commit has to be tagged.

- The build has to pass the tests.

To create a new release, follow these steps (my publishing TODO list).

#### Prepare Release

1. Make sure all tests pass: `npm run test`

2. Make sure [CHANGELOG.md](./CHANGELOG.md) contains the changes and the current date next to the version.

#### Release

1. `npm version [patch|minor|major]`

2. `git push --follow-tags`

After 2-30min, a new version should be published on npm. To check which files are being published, check Travis log or [unpkg.com](https://unpkg.com/puppeteer-cluster/).

#### Failed release

Sometimes a test fails on Travis and a new version is not published. In that case do the following:

1. Delete the tag from the local repository

- `git tag -d v0.X.X`

2. Delete the tag from the remote repository

- `git push --delete origin v0.X.X`

3. Fix the problem

4. Commit and push the (untagged) changes and make sure the tests succeed.

5. Manually tag the commit which fixes the problem

- `git tag v0.X.X`

6. Push the tag

- `git push --tags`

Now Travis will run again, hopefully succeeding this time.

## Dependency Upgrades

The repository has [dependabot](https://github.com/dependabot) enabled. At the start of each month, the bot will open one pull request for each dependency which needs to be upgraded. Most dependencies are uncritical and used for testing. If the tests succeed they can be merged.

### Manually upgrading dependencies

If there are a lot of dependencies with updates available it might be easier to upgrade them all at once manually:

1. Check which dependencies are outdated (this will return nothing if all dependencies are up-to-date)

- `npm outdated`

2. Update all dependencies to their latest version (according to `package.json`). This should do all minor updates.

- `npm update`

3. In case there are any vulnerabilities (`found ... high severity vulnerabilities`), fix them so that people installing this library, don't get warnings.

- `npm audit fix`

4. Test if everything went okay, all tests should pass.

- `npm run test`

5. Check again if there are any outdated dependencies. This might be major upgrades which are not performed automatically.

- `npm outdated`

6. Manually decide for each package if it should be upgraded and upgrade them:

- `npm install PACKAGE\_NAME@latest --save`

7. Rerun tests.

- `npm run test`

8. If everything worked, the changes can be pushed, probably followed by a new release.