# How to contribute to billboard.js

billboard.js is open to everyone, and we welcome any kinds of contribution.

We believe that our project can grow with your interests in helping others' necessities.

## Style Guide

billboard.js has several style guidelines that you must follow.

Before your start, please read the below instructions carefully.

### Linting and Code Convention

To maintain the code style and quality, we adopted [ESLint](http://eslint.org/).

The [rules](https://github.com/naver/eslint-config-naver/tree/master/rules) are based on the [Airbnb JavaScript Style Guide](https://github.com/airbnb/javascript) with some modifications.

### Commit Log Guidelines

billboard.js follows formatted commit logs based on [Conventional Commits](https://www.conventionalcommits.org/) for many different purposes (like creating CHANGELOG, ease history searching, etc.).

To not break the current format, you'll be forced to follow our commit log guidelines.

Before your commit/push, make sure follow our commit log guidelines.

The outline is as below:

```bash

<type>[optional scope]: <description>

[optional body]

[optional footer]

```

- #### Types

- \*\*feat\*\*: A new feature

- \*\*fix\*\*: A bug fix

- \*\*docs\*\*: Documentation only changes

- \*\*style\*\*: Changes that do not affect the meaning of the code. Such as white-space, formatting, missing semi-colons, etc.

- \*\*refactor\*\*: A code change that neither fixes a bug nor adds a feature

- \*\*test\*\*: Adding missing tests. Changing tests.

- \*\*chore\*\*: Changes to the build process or tools and libraries such as documentation generation

- \*\*skip\*\*: For commits made by after the 1st commit. Usually for applying code review changes.

- #### Description

A short description of the commit.

```bash

fix(Axis): Correct tick rendering

```

Example commit applying code review (after the 1st commit)

> \*\*Note:\*\* Commit log starting with `skip:` type will be ignored by commit hook constraint.

```bash

skip: Applied the review

```

- #### Body

A short descriptive message part of the commit.

Example commit of fixing a bug:

```bash

<type>(<module>): <subject>

Update condition of tick to avoid unnecessary tick rendering

<footer>

```

> \*\*BREAKING CHANGE:\*\* a commit that has the text 'BREAKING CHANGE:' at the beginning of its optional body or footer section introduces a breaking API change.

- #### Footer

Related github issue number referenced by `Ref #ISSUE-NO`.

ex) When the commit is about issue number 20, then

```bash

Ref #20

```

- #### Example

```bash

fix(Axis): Correct tick rendering

Update condition of tick to avoid unnecessary tick rendering

Ref #20

```

```bash

feat(tooltip): Intent to ship tooltip.contents.template

- Implementation of tooltip.contents.template

- Update legend template processing with util's .tplProcess()

Ref #813

```

```bash

feat(browser): Drop support for IE 11

Drop support for IE 11.

Ref #31

BREAKING CHANGE: Internet Explorer 11 is a burden to support and at EOL.

```

## How to submit Pull Requests

Steps to submit your pull request:

1. Fork `billboard.js` on your repository

2. Create a new branch from your billboard.js `master` branch (and be sure to be always up-to-date)

3. Do your work

4. Create test code for your work (when is possible)

5. Run `npm run lint` for linting and code style check. (update until without any error or warnings)

6. Run test code by `npm test` or `npm test:chrome` for chrome browser.

Make sure all tests pass at least on the latest version of Chrome(mobile/desktop).

7. Write a commit log following convention and push to your repository branch.

8. Create a new PR from your branch to `billboard.js/master` branch.

9. Wait for reviews.

When your contribution is well enough to be accepted, then it will be merged to our branch.

10. All done!

## License

By contributing to billboard.js, you're agreeing that your contributions will be licensed under its [MIT](https://opensource.org/licenses/MIT) license.