# Contributing

Our vision for Backstage is for it to become the trusted standard toolbox (read: UX layer) for the open source infrastructure landscape. Think of it like Kubernetes for developer experience. We realize this is an ambitious goal. We can’t do it alone.

Therefore we want to create strong community of contributors -- all working together to create the kind of delightful experience that our developers deserve.

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given. ??

Backstage is released under the Apache2.0 License, and original creations contributed to this repo are accepted under the same license.

# Types of Contributions

## Report bugs

No one likes bugs. Report bugs as an issue [here](https://github.com/spotify/backstage/issues/new?template=bug\_template.md).

## Fix bugs or build new features

Look through the GitHub issues for [bugs](https://github.com/spotify/backstage/labels/bugs), [good first issues](https://github.com/spotify/backstage/labels/good%20first%20issue) or [help wanted](https://github.com/spotify/backstage/labels/help%20wanted).

## Build a plugin

The value of Backstage grows with every new plugin that gets added. Wouldn't it be fantastic if there was a plugin for every infrastructure project out there? We think so. And we would love your help.

A great reference example of a plugin can be found on [our blog](https://backstage.io/blog/2020/04/06/lighthouse-plugin) (thanks [@fastfrwrd](https://github.com/fastfrwrd)!)

What kind of plugins should/could be created? Some inspiration from the 120+ plugins that we have developed inside Spotify can be found [here](https://backstage.io/demos), but we will keep a running list of suggestions labeled with [[plugin]](https://github.com/spotify/backstage/labels/plugin).

## Suggesting a plugin

If you start developing a plugin that you aim to release as open source, we suggest that you create a new [new Issue](https://github.com/spotify/backstage/issues/new?template=plugin\_template.md). This helps the community know what plugins are in development.

You can also use this process if you have an idea for a good plugin but you hope that someone else will pick up the work.

## Write Documentation

The current documentation is very limited. Help us make the `/docs` folder come alive.

## Contribute to Storybook

We think the best way to ensure different plugins provide a consistent experience is through a solid set of reusable UI/UX components. Backstage uses [Storybook](http://storybook.backstage.io).

Either help us [create new components](https://github.com/spotify/backstage/labels/help%20wanted) or improve stories for the existing ones (look for files with `\*.stories.tsx`).

## Submit Feedback

The best way to send feedback is to file [an issue](https://github.com/spotify/backstage/issues).

If you are proposing a feature:

- Explain in detail how it would work.

- Keep the scope as narrow as possible, to make it easier to implement.

- Use appropriate labels

- Remember that this is a volunteer-driven project, and that contributions

are welcome :)

## Add your company to ADOPTERS

Have you started using Backstage? Adding your company to [ADOPTERS](ADOPTERS.md) really helps the project.

# Get Started!

So...feel ready to jump in? Let's do this. Head over to the [Getting Started guide](https://github.com/spotify/backstage#getting-started) ???

If you need help, just jump into our [Discord chatroom](https://discord.gg/MUpMjP2).

# Coding Guidelines

All code is formatted with `prettier` using the configuration in the repo. If possible we recommend configuring your editor to format automatically, but you can also use the `yarn prettier --write <file>` command to format files.

If you're contributing to the backend or CLI tooling, be mindful of cross-platform support. [This](https://shapeshed.com/writing-cross-platform-node/) blog post is a good guide of what to keep in mind when writing cross-platform NodeJS.

Also be sure to skim through our [ADRs](https://github.com/spotify/backstage/tree/master/docs/architecture-decisions) to see if they cover what you're working on. In particular [ADR006: Avoid React.FC and React.SFC](https://github.com/spotify/backstage/blob/master/docs/architecture-decisions/adr006-avoid-react-fc.md) is one to look out for.

# Code of Conduct

This project adheres to the [Spotify FOSS Code of Conduct][code-of-conduct]. By participating, you are expected to honor this code.

[code-of-conduct]: https://github.com/spotify/backstage/blob/master/CODE\_OF\_CONDUCT.md

# Security Issues?

See [SECURITY](SECURITY.md).