# How to contribute

Polaris React is Shopify’s most active open source project. It’s the only project that’s both under very active development and is used to build Shopify and Shopify apps every day. We want to make it as easy and transparent as possible to contribute. If we are missing anything or can make the process easier in any way, [please let us know](mailto:polaris@shopify.com).

## Code of conduct

We expect all participants to read our [code of conduct](https://github.com/Shopify/polaris-react/blob/master/.github/CODE\_OF\_CONDUCT.md) to understand which actions are and aren’t tolerated.

## Open development

All work on Polaris React happens directly on GitHub. Both team members and external contributors send pull requests which go through the same review process.

## Semantic versioning

Polaris React follows semantic versioning. We release [patch versions for bug fixes](https://github.com/Shopify/polaris-react/blob/master/documentation/Versioning%20and%20changelog.md#patch), [minor versions for new features](https://github.com/Shopify/polaris-react/blob/master/documentation/Versioning%20and%20changelog.md#minor), and [major versions for breaking changes](https://github.com/Shopify/polaris-react/blob/master/documentation/Versioning%20and%20changelog.md#major). When we make breaking changes, we introduce deprecation warnings in a minor version along with the upgrade path so that our users learn about the upcoming changes and migrate their code in advance.

We tag every pull request with a label marking whether the change should go in the next patch, minor, or a major version. We release new patch versions most weeks, minor versions every couple weeks, and major versions one or two times a year.

Every significant change is documented in the [CHANGELOG](https://github.com/Shopify/polaris-react/blob/master/CHANGELOG.md). We have a Probot ? to remind you to include a CHANGELOG entry.

## Branch organization

We do our best to keep `master` releasable at all times, with work for major releases happening in separate branches. [Breaking changes](https://github.com/Shopify/polaris-react/blob/master/documentation/Versioning%20and%20changelog.md#major) should never be merged directly to `master`. Otherwise, if you send a pull request please do it against the `master` branch. Continue reading for more about pull requests and breaking changes.

## Bugs

### Where to find known issues

We track all of our issues in GitHub and [bugs](https://github.com/Shopify/polaris-react/labels/Bug) are labeled accordingly. If you are planning to work on an issue, avoid ones which already have an assignee, where someone has commented within the last two weeks they are working on it, or the issue is labeled with [fix in progress](https://github.com/Shopify/polaris-react/labels/fix%20in%20progress). We will do our best to communicate when an issue is being worked on internally.

### Reporting new issues

To reduce duplicates, look through open issues before filing one. When [opening an issue](https://github.com/Shopify/polaris-react/issues/new?template=ISSUE.md), complete as much of the template as possible. The best way to get your bug fixed is to provide a reduced test case. [This CodeSandbox template](https://codesandbox.io/s/q82mlq0m26) is a great starting point.

## Feature requests

Before requesting a feature, search the [existing feature requests](https://github.com/Shopify/polaris-react/issues?utf8=%E2%9C%93&q=is%3Aissue+label%3A%22feature+request%22+sort%3Areactions-%2B1-desc). You can [? upvote](https://help.github.com/articles/about-conversations-on-github/) feature requests to help the Polaris team set priorities. If a feature request is closed, you can still upvote! A closed feature request means it’s not something we’re currently working on, but we take all your input into account when planning what to work on next.

Otherwise, [request a feature](https://github.com/Shopify/polaris-react/issues/new?template=FEATURE\_REQUEST.md).

## Proposing a change

If you intend to build a new component, change a public API, make design improvements, or any other non-trivial changes, [we recommend filing an issue](https://github.com/Shopify/polaris-react/issues/new?template=FEATURE\_REQUEST.md). This lets us all discuss and reach an agreement on the proposal before you put in significant time and effort.

Please use this [template for proposing new components or changes to existing components](https://github.com/Shopify/polaris-react/issues/new?template=NEW\_COMPONENT.md).

If you’re only fixing a bug, it’s okay to submit a pull request right away but we still recommend you file an issue detailing what you’re fixing. This is helpful in case we don’t accept that specific fix but want to keep track of the issue.

## Your first pull request

Working on your first pull request? You can learn how from this free video series:

[How to Contribute to an Open Source Project on GitHub](https://egghead.io/series/how-to-contribute-to-an-open-source-project-on-github)

To help you get familiar with our contribution process, we have a list of [good first issues](https://github.com/Shopify/polaris-react/labels/good%20first%20issue) that contain bugs with limited scope. This is a great place to get started.

If you decide to fix an issue, please check the comment thread in case somebody is already working on a fix. If nobody is working on it, leave a comment stating that you intend to work on it.

If somebody claims an issue but doesn’t follow up for more than two weeks, it’s fine to take it over but still leave a comment stating that you intend to work on it.

### Sending a pull request

We’ll review your pull request and either merge it, request changes to it, or close it with an explanation. We’ll do our best to provide updates and feedback throughout the process.

\*\*Before submitting a pull request\*\*, please:

1. Fork the repository and create your branch from `master`

1. Run `yarn` in the repository root

1. If you’ve fixed a bug or added code that [should be tested](https://github.com/Shopify/web-foundation/blob/master/Best%20practices/Testing.md), add tests

1. Ensure the test suite passes with `yarn test` (protip: `yarn test:watch TestName` is helpful in development)

1. If your pull request modifies any SVG files run `yarn run sewing-kit optimize`

1. Format your code with `yarn format`

1. Make sure your code lints with `yarn lint`

1. Run the TypeScript compiler with `yarn type-check`

1. [Tophat ?](https://github.com/Shopify/polaris-react/blob/master/documentation/Tophatting.md) your changes locally with `yarn dev`

1. If you haven’t already, [sign a CLA](https://cla.shopify.com/)

### Contributor License Agreement (CLA)

Each contributor is required to [sign a CLA](https://cla.shopify.com/). This process is automated as part of your first pull request and is only required once. If any contributor has not signed or does not have an associated GitHub account, the CLA check will fail and the pull request is unable to be merged.

### Contribution prerequisites

- You have Node installed at v10.13.0+ and Yarn at v1.10.1+

### Best practices

Take a look at [our best practices](https://github.com/Shopify/web-foundation/tree/master/handbook/Best%20practices) which include [testing](https://github.com/Shopify/web-foundation/blob/master/handbook/Best%20practices/Testing.md), [React testing](https://github.com/Shopify/web-foundation/blob/master/handbook/Best%20practices/React/Testing.md), [Jest](https://github.com/Shopify/web-foundation/blob/master/handbook/Best%20practices/Jest.md), and [Enzyme](https://github.com/Shopify/web-foundation/blob/master/handbook/Best%20practices/Enzyme.md). We will continue to add best practices here.

### Development workflow

After cloning Polaris React, run `yarn` to fetch its dependencies. Then you can run several commands:

- `yarn dev` runs a Storybook server which includes a playground editable at `playground/Playground.tsx`

- `yarn test` runs the complete test suite

- `yarn test:coverage` runs tests and generates a test coverage report

- `yarn test <pattern>` runs tests with matching filenames

- `yarn tophat` runs a local development server and mounts all of the component examples

We recommend running `yarn test` (or its variations above), as well as trying your build of Polaris React in a real project, to make sure you don’t introduce any regressions as you work on your change.

Run `yarn run build-consumer <project>` where `<project>` is the directory the build will be copied. \*\*The `<project>` directory must be a sibling of the `polaris-react` directory\*\*.

```sh

# Example

yarn run build-consumer polaris-styleguide

```

In the example above, the build is copied to `polaris-styleguide/node\_modules/@shopify/polaris`. And in this case, a rebuild of `polaris-styleguide` is required after copying the `polaris-react` build, but may not be the case for all consuming projects.

### Code style

We use an automatic code formatter called [Prettier](https://prettier.io/). Run `yarn format` after making any changes to the code.

Linting will catch common issues that may exist in your code. You can check the status of your code styling by running `yarn lint`.

Our code editor of choice is [VS Code](https://code.visualstudio.com/) which has [integrations with Prettier](https://github.com/prettier/prettier-vscode) and our linting tools which make this automatic. If you choose to use VS Code, these integrations will be listed as recommended extensions (or search for `@recommended`) in the extensions panel.

However, there are still some styles that the linter cannot pick up. If you are unsure about something, looking at [our JavaScript style guide](https://github.com/Shopify/javascript) will guide you in the right direction.

## Breaking changes

If your pull request contains breaking changes, please target the branch for the next major release and open a pull request against `master` that introduces the deprecation warnings and upgrade path. If you are unsure if the changes are considered breaking or not, open your pull request against the `master` branch and let us know. We understand it can be uncomfortable asking for help and this is why we have a [code of conduct](https://github.com/Shopify/polaris-react/blob/master/.github/CODE\_OF\_CONDUCT.md) to ensure the community is positive, encouraging, and helpful.