# Contributing

We welcome community support with both pull requests and reporting bugs. Please

don't hesitate to jump in.

## Review others' work

Check out the list of outstanding pull requests if there is something you might

be interested in. Maybe somebody is trying to fix that stupid bug that bothers

you. Review the PR. Do you have any better ideas how to fix this problem? Let us

know.

## Issues

The issue tracker is the preferred channel for bug reports, features requests

and submitting pull requests, but please respect the following restrictions:

- Please do not use the issue tracker for personal support requests. Stack Overflow ([react-bootstrap](http://stackoverflow.com/questions/tagged/react-bootstrap) tag), [Discord](https://discord.gg/0ZcbPKXt5bXLs9XK), or [Thinkful](http://start.thinkful.com/react/?utm\_source=github&utm\_medium=badge&utm\_campaign=react-bootstrap) are better places to get help.

- Please do not open issues or pull requests regarding the code in React or Bootstrap (open them in their respective repositories).

\_Note: Occasionally issues are opened that are unclear, or we cannot verify them. When the issue author has not responded to our questions for verification within 7 days then we will close the issue.\_

## Tests

All commits that fix bugs or add features need a test. You can run `npm run tdd MyComponentName` for component specific tests.

## API Design

Try and be consistent with the overall style and API of the library as a whole. Generally, we avoid monolithic or very high level component APIs. React bootstrap is a toolbox! Prefer to split components out into "sub components" as they make sense. This is usually indicated by the bootstrap CSS classes, e.g. `.nav`, `.nav-item`, and `nav-link` translate into `<Nav>`, `<NavItem>`, and `<NavLink>` components.

Avoid unnecessary Higher Order Components (HOCs), unless they add a significant amount of value or abstract way something that would otherwise complicate many components (like `uncontrollable`). It's not that HOCs are bad, but we want to try and keep these low level UI blocks as flat and straightforward as possible. Prefer to work explicitly in the component and avoid over optimization up front.

Components should not be function components by default. Folks often add `refs` to them so class components are a better default for `react-bootstrap` components. Components should also \*\*not\*\* use `PureComponent` by default. For a variety of reasons the sort of components these are don't generally benefit from that optimization, and may cause bugs.

### Accessible by Default

React-bootstrap takes web accessibility (a11y) seriously and we take advantage of the React component model to add better defaults that plain bootstrap can (being mostly CSS). Often this means, making sure the a11y details present in the [bootstrap docs](https://getbootstrap.com/) are added as defaults to components where possible. Usually this means handling `aria-selected`/`aria-controls` for tab like components or having a default label for an icon only button, or making it easier to apply `htmlFor` and `id` to form controls.

\*\*There are plenty of cases where the correct a11y is only possible from the user's code and that's okay!\*\* We can't handle every case.

## Visual Changes

When making a visual change, please provide screenshots

and/or screencasts of the proposed change. This will help us to understand the

desired change easier.

## Documentation

Please update the docs with any API changes, the code and docs should always be

in sync.

Component prop documentation is generated automatically from the React components

and their leading comments. Please make sure to provide comments for any `propTypes` you add

or change in a Component.

```js

propTypes: {

/\*\*

\* Sets the visibility of the Component

\*/

show: PropTypes.bool,

/\*\*

\* A callback fired when the visibility changes

\* @type {func}

\* @required

\*/

onHide: myCustomPropType

}

```

There are a few caveats to this format that differ from conventional JSDoc comments.

- Only specific doclets (the @ things) should be used, and only when the data cannot be parsed from the component itself

- `@type`: Override the "type", use the same names as the default React PropTypes: string, func, bool, number, object. You can express enum and oneOfType types, Like `{("optionA"|"optionB")}`.

- `@required`: to mark a prop as required (use the normal React isRequired if possible)

- `@private`: Will hide the prop in the documentation

- All description text should be above the doclets.

## Implement additional components and features

This project is seeking parity with the core Bootstrap library.

Component by component to the extent it is possible.

Also Bootstrap mentions http://getbootstrap.com/getting-started/#examples

as examples of things you can do, but they are not part of the core library,

therefore this project is the wrong place to implement them.

## Breaking changes

Breaking changes should be accompanied with deprecations of removed functionality. Prior to the 1.0.0 release, we aim to follow React's example of taking two minor releases to break old functionality. As such, changes that intend to remove or change public APIs should be be submitted against the `next` branch, and should be accompanied with deprecation warnings on the old APIs. The deprecated APIs themselves should not be removed until the minor release after that.

## Notes for lodash functions usage in the code

You can use `lodash` but keep it to things where it actually needs it, i.e. don't use `lodash`'s `forEach` when `Array.prototype.forEach` is fine.

The full discussion about it at [#889](https://github.com/react-bootstrap/react-bootstrap/issues/889)

## Collaborators

Please see the [Maintaining](./MAINTAINING.md) documentation.

[huboard-badge]: https://img.shields.io/badge/Hu-Board-7965cc.svg

[huboard]: https://huboard.com/react-bootstrap/react-bootstrap

[issues]: https://github.com/react-bootstrap/react-bootstrap/issues

[editorconfig]: http://editorconfig.org

[eslint]: http://eslint.org

[commit-message]: http://robots.thoughtbot.com/5-useful-tips-for-a-better-commit-message