# Contributing

Thanks for considering contributing to `react-beautiful-dnd`! ??

There are a few categories of contribution so we'll go through them individually.

## Documentation

If you think the docs could be improved - please feel free to raise a pull request!

## Bug

If you spot a bug you are welcome to raise it on our issue page. You are also welcome to take a crack at fixing it if you like! When you create an issue you will be prompted with the details we would like you to provide.

## Feature request

If you would like to see a feature added to the library, here is what you do:

1. Have a read of `README.md` to understand the motivations of this library. It is fairly opinionated and is not intended to be a universal drag and drop library. As such, it will not support every drag and drop interaction.

2. Have a search through the [open and closed issues](https://github.com/atlassian/react-beautiful-dnd/issues?utf8=%E2%9C%93&q=is%3Aissue) to see if the feature you are requesting as already been requested.

3. Have a clear and general purpose keyboard story for any feature request

4. Please [create an issue](https://github.com/atlassian/react-beautiful-dnd/issues/new) to discuss it.

\*\*Please do not raise a pull request directly\*\*. There may be reasons why we will not add every feature to this library.

## Large contributions

If you are interested in making a large contribution to this library there is some recommended reading / training we suggest. There is a large amount of different libraries, techniques and tools used in `react-beautiful-dnd` and we have created a list with resources about them. Not everything in the list will be applicable to everyone. But it is a great reference and starting point for those who do not know where to start.

The online courses listed are no free - feel free to seek out alternatives if you want to. We recommend the egghead.io courses because they are quite comprehensive.

### General knowledge

- [You Don't Know JS](https://github.com/getify/You-Dont-Know-JS): This is an amazing resource that I recommend all the time. It is great for having a deeper understanding of the JavaScript language.

### Technologies

#### `React`

This is a `React` project so getting familiar with `React` is a must!

- [`react`](https://facebook.github.io/react/)

- [An intro to using React](https://egghead.io/courses/start-using-react-to-build-web-applications)

#### `Redux`

This project uses `redux` for its state management. If you have not used `redux` before it is worth getting familiar with it.

- [`redux`](http://redux.js.org/docs/introduction/): the library itself has really great docs

- [Getting Started with Redux](https://egghead.io/courses/getting-started-with-redux): the whole course

- [Building React Applications with Idiomatic Redux](https://egghead.io/courses/building-react-applications-with-idiomatic-redux): no need to do the lessons on react router or data fetching

- [`react-redux`](https://github.com/reactjs/react-redux): `react` bindings for `redux`

- [`reselect`](https://github.com/reactjs/reselect): we use `reselect` heavily to ensure that state selectors are as fast as they can be. Please have a read of its main page, especially the [sharing Selectors with Props Across Multiple Components](https://github.com/reactjs/reselect#sharing-selectors-with-props-across-multiple-components) section.

#### Testing

We test our application very thoroughly. Changes will not be accepted without tests

- [`jest`](https://facebook.github.io/jest/): We use the jest test runner. It is worth getting familiar with it

- [Test JavaScript with Jest](https://egghead.io/lessons/javascript-test-javascript-with-jest)

- [React Testing Cookbook](https://egghead.io/courses/react-testing-cookbook)

#### Performance

Performance is \*\*critical\*\* to this project. Please get familiar with React performance considerations. Changes that break core performance characteristics will not be accepted.

- [Performance optimisations for React applications](https://medium.com/@alexandereardon/performance-optimisations-for-react-applications-b453c597b191)

- [Performance optimisations for React applications round 2](https://medium.com/@alexandereardon/performance-optimisations-for-react-applications-round-2-2042e5c9af97)

- [React performance tools](https://facebook.github.io/react/docs/perf.html)

- [React performance documentation](https://facebook.github.io/react/docs/optimizing-performance.html)

- [React is slow, React is fast](https://marmelab.com/blog/2017/02/06/react-is-slow-react-is-fast.html)

#### Flow

This codebase is typed with [`flow`](https://flow.org/). Changes will not be merged without correct flow typing. If you are not sure about a particular use case let flow break the build and it can be discussed in the pull request.

- [`flow`](https://flow.org/en/docs/getting-started/): the `flow` docs are great

- [Up and Running with Facebook Flow for Typed JavaScript](https://egghead.io/lessons/javascript-up-and-running-with-facebook-flow-for-typed-javascript): a small primer for running flow

### Drag and drop problem space

#### HTML5 drag and drop

How this library performs dragging is an implementation detail. The api is what users interact with. That said, this library does not use the html5 drag and drop api. The main reason is that html5 drag and drop does not allow the level of control we need to create our powerful and beautiful experiences. I could go into detail but this is not the right forum.

Here is some general reading about html5 drag and drop. It is worth having a read to get familiar with its ideas and api

- [HTML5 drag and drop api](https://developer.mozilla.org/en-US/docs/Web/API/HTML\_Drag\_and\_Drop\_API)

- [HTML5 Rocks - dnd basics](https://www.html5rocks.com/en/tutorials/dnd/basics/)

- [The HTML5 drag and drop disaster](https://www.quirksmode.org/blog/archives/2009/09/the\_html5\_drag.html)

- [HTML5 drag and drop browser inconsistencies](http://mereskin.github.io/dnd/)

#### Prior work

It is worth looking at other libraries out there to see how they do drag and drop. Things to look at is their philosophy and api. `react-beautiful-dnd` is an opinionated, higher level abstraction than most drag and drop libraries. We do not need to support every use case. We need to find the right level of control while still maintaining a beautiful experience for the user, flexibility of use and a clean, powerful api.

- [`react-dnd`](https://react-dnd.github.io/react-dnd/) - `react-beautiful-dnd` draws a fair amount of inspiration from `react-dnd`. Something to keep in mind is that `react-dnd` is designed to provide a set of drag and drop primitives which is a different set of goals to this project.

- [`react-sortable-hoc`](https://github.com/clauderic/react-sortable-hoc/) - on the surface this library looks similar to `react-beautiful-dnd`. I created a [comparison blog](https://medium.com/@alexandereardon/thanks-for-reaching-out-dimitar-nestorov-8c0bf9abe19) that explains the differences

- [`jQuery sortable`](http://jqueryui.com/sortable/) - the king of drag and drop for a long time