

How to lift state up

We learned that if we need to pass data back up to a host component from an inner we can do that by lifting state up. Effectively, it is passing a function down into the inner and running it in the inner. Let's exercise that.

On our landing page we have a list of days that we created inside `<PickDate />`, remember? When the user taps on a day, we want to set it as the "current date". Let's implement that by lifting state up, then let's come back around and refactor it to be cleaner. Ready? Here we go!

1. Open `LandingPage.js`. If `currentDate` is still being hardcoded as a new `Date()`, change that to read the `currentDate` from state like so:

```
const currentDate = new Date(state.currentDate);
```

2. Create a new method called `set_current_date(date)`. It should dispatch an action to the store like this:

```
store.dispatch(actions.setCurrentDate(date))
```

3. Pass that method down to `PickDate` like this:

```
<PickDate onPicked={set_current_date} />
```

4. Edit `PickDate`. Make it work like this: when the user clicks/taps a date, we call the `onPicked` method passing the date chosen. (Hint: You'll need to use the arrow function form of event handling).
5. Run and test. When you tap a day, the current date should change. You'll see it on the big page header.

Got it working? Awesome. Now let's break it. ;-).

Seriously, the above steps were a valuable learning exercise for the situations where you need to lift state up. But when your team uses a state container like `Redux`, the need for lifting state up is greatly diminished. When you combine that with the fact that lifting state up increases coupling and complexity, the best practice would be to avoid it when it makes sense to do so. To prove my point, try tapping a day in `FilmBrief`. It fails because we haven't provided the `set_current_date` function in props.

So let's now get some practice in refactoring. Don't worry, it will only take a few minutes.

6. Pull the `set_current_date` function out of `LandingPage` and move it into `PickDate` so that `PickDate` is more encapsulated. Change the `onClick` event in `PickDate` to call its local function. Remove the `onPicked` prop.
7. Run and test. Is it still setting the day properly? And check out `FilmDetails` because it now works there also. As you click a day the date and the showingTimes are changing!