

# Running our Weather app on your machine

We'll look at how to run our app on your machine so that you can run tests against the app

## WE'LL COVER THE FOLLOWING ^

- Prerequisites
- First steps
- Install npm modules

## Prerequisites #

Node.js is a JavaScript runtime for your terminal. Don't worry too much about it, but it's used by some tools we'll be using to build our application. If you don't have it installed already (check by running `node -v` in your terminal, which should print a version number) head over to [nodejs.org](https://nodejs.org), follow the instructions there to install the latest version (v6 at the time of this writing) and you're good to go!

## First steps #

Facebook recently open sourced a neat little tool called `create-react-app` that allows us to very easily get started with our React app! It includes all the necessary build tools and transpilation steps to just get stuff done.

Let's install it with `npm`:

```
npm install -g create-react-app
```

As soon as that's finished you now have access to the `create-react-app` command in your terminal! Let's create our barebones weather app:

```
create-react-app weather-app
```

The argument to `create-react-app`, in our case `weather-app`, tells the utility what to name the folder it'll create. Since we're creating a weather app, `weather-app` seems like a solid choice!

## Install npm modules #

During the lessons, you might have noticed that we have used a few npm modules. Let's install those modules in your app directory

```
npm install --save redux react-redux
npm install --save redux-thunk
npm install --save immutable
```

Now comes the hardest part (*just kidding* :)).

Copy the code from the files below and put it into the respective files.

`index.js` is already there and you should create the other files e.g. `App.js`, `Plot.js`, `reducers.js`, `actions.js` etc.

```
import React from 'react';

class Plot extends React.Component {
  shouldComponentUpdate(nextProps) {
    const xDataChanged = !this.props.xData.equals(nextProps.xData);
    const yDataChanged = !this.props.yData.equals(nextProps.yData);

    return xDataChanged || yDataChanged;
  }

  drawPlot = () => {
    Plotly.newPlot('plot', [{
      x: this.props.xData.toJS(),
      y: this.props.yData.toJS(),
      type: this.props.type
    }], {
      margin: {
        t: 0, r: 0, l: 30
      },
      xaxis: {
        gridcolor: 'transparent'
      }
    }, {
      displayModeBar: false
    });

    document.getElementById('plot').on(
      'plotly_click', this.props.onPlotClick);
  };
}
```

```
        plotly_click, this.props.onPlotClick),
    }

    componentDidMount() {
        this.drawPlot();
    }

    componentDidUpdate() {
        this.drawPlot();
    }

    render() {
        console.log('RENDER PLOT');
        return (
            <div id="plot"></div>
        );
    }
}

export default Plot;
```

Thankfully, create-react-app includes a simple server so instead of having to open the index.html file manually we can simply run `npm run start` in the weather-app directory and see our application at localhost:3000!

Now that we have our app running on your machine, lets jump into testing it.