

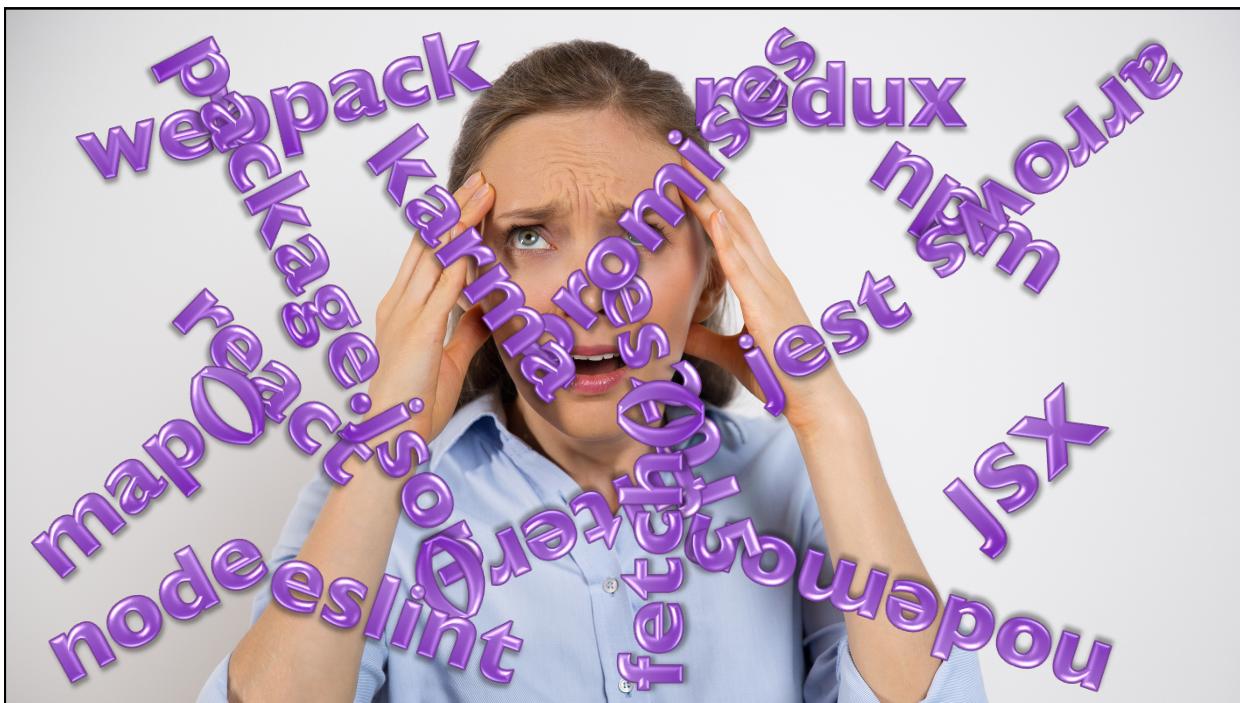
# Create React App

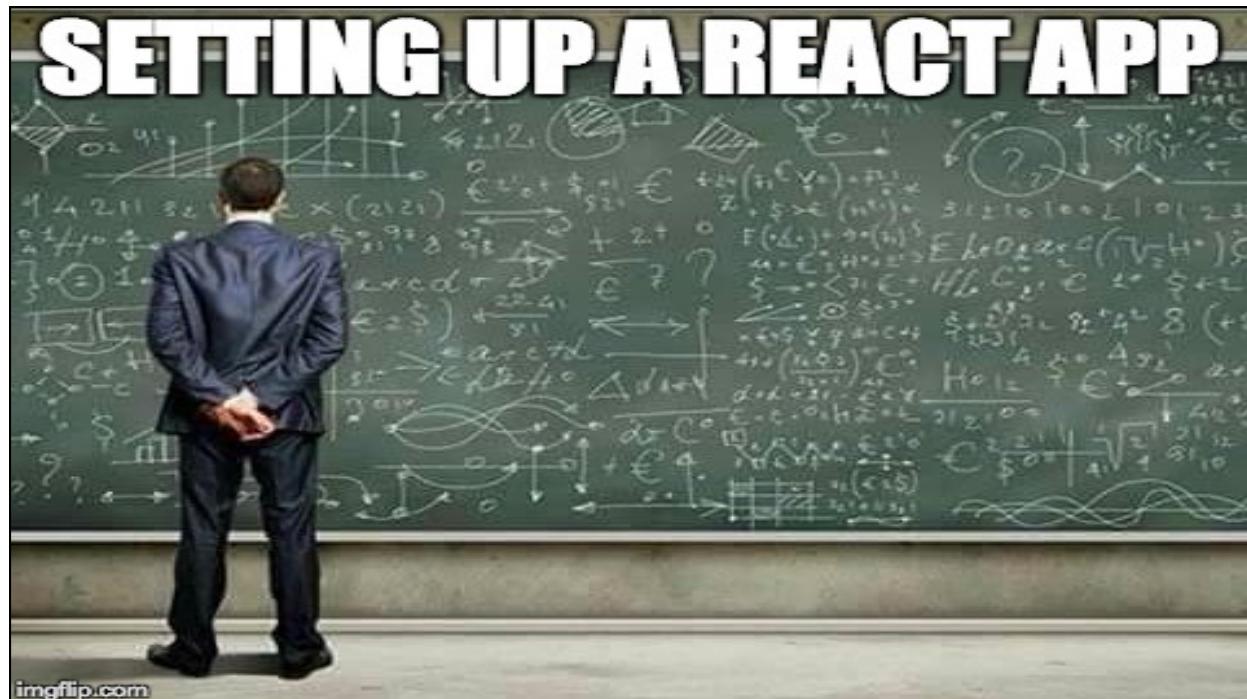
## tl;dr

- React is very difficult to configure and brittle.
- So the team has provided us an incredibly simple tool called `create-react-app` that scaffolds the project
- Compile and run in watch mode with `npm run dev`
- To transpile, bundle, minify, lint and run tests for production, you `npm run build`

# React has a lot of moving parts

React uses JSX, which isn't JavaScript ...  
So it has to be transpiled by Babel ...  
which should be automated by webpack ...  
which also minifies, bundles and tree-shakes ...  
the CSS, HTML, and hundreds of libraries ...  
which must be managed by npm ...  
thus carefully configured in package.json ...  
linted with eslint and tested with jest ...  
using ES2015 modules and AMD requires ...  
run through node processes ...





imgflip.com

We knew it was frustrating to spend days setting up a project when all you wanted was to learn React. We wanted to fix this.

We loved Ember CLI and Elm Reactor which help users to get started.

So Christopher Chedeau, Kevin Lacker, and I wrote create-react-app



## Installation is optional with **npx**



## create-react-app makes your project

**npx create-react-app hello-world**

```

● ● ●
└ react-dom@15.2.1

Success! Created hello-world at /Users/dan/hello-world.

Inside that directory, you can run several commands:
* npm start: Starts the development server.
* npm run build: Bundles the app into static files for production.
* npm run eject: Removes this tool. If you do this, you can't go back!

We suggest that you begin by typing:
cd /Users/dan/hello-world
npm start

Happy hacking!
create-react-app)

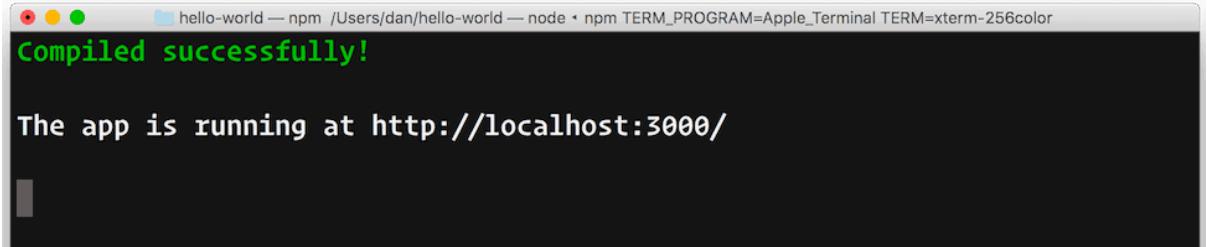
```

Your project has a single build dependency and some node scripts.

```
package.json
{
  "name": "whatever-you-want",
  "dependencies": {
    "react": "^17.2.1",
    "react-dom": "^17.2.1"
  },
  "devDependencies": {
    "react-scripts": "1.1.0"
  },
  "scripts": {
    "start": "react-scripts start",
    "build": "react-scripts build",
    "eject": "react-scripts eject"
  }
}
```

## Starting the Server

- Run npm start to launch the development server. The browser will open automatically with the created app's URL.



A screenshot of a Mac OS X terminal window titled 'hello-world — npm /Users/dan/hello-world — node • npm TERM\_PROGRAM=Apple\_Terminal TERM=xterm-256color'. The window shows the command 'npm start' being run, followed by the message 'Compiled successfully!' in green text, and the instruction 'The app is running at http://localhost:3000/'.

```
hello-world — npm /Users/dan/hello-world — node • npm TERM_PROGRAM=Apple_Terminal TERM=xterm-256color
Compiled successfully!

The app is running at http://localhost:3000/
```

## Then you're in watch mode

- When your code changes, webpack lints with eslint and transpiles with babel.

```
● ● ● hello-world — npm /Users/dan/hello-world — node • npm TERM_PROGRAM=Apple_Terminal TERM=xterm-256color
Failed to compile.

Error in ./src/App.js
Syntax error: /Users/dan/hello-world/src/App.js: Unexpected token (8:7)
  6 |   render() {
  7 |     return (
> 8 |       <><div className="App">
  9 |         ^
  10 |         <div className="App-header">
  11 |           <img src={logo} className="App-logo" alt="logo" />
    11 |           <h2>Welcome to React</h2>
@ ./src/index.js 11:11-27
```

## ESLint output shows in the console

- The lint rules are tuned for a React app.

```
● ● ● hello-world — npm /Users/dan/hello-world — node • npm TERM_PROGRAM=Apple_Terminal TERM=xterm-256color
Compiled with warnings.

Warning in ./src/App.js

/Users/dan/hello-world/src/App.js
  7:9  warning  'hello' is defined but never used  no-unused-vars

✖ 1 problem (0 errors, 1 warning)

You may use special comments to disable some warnings.
Use // eslint-disable-next-line to ignore the next line.
Use /* eslint-disable */ to ignore all warnings in a file.
```

## "npm run build" to go to production

It is minified, bundled, content hashed (for caching).

```
● ● ● fish /Users/dan/hello-world — -fish
hello-world) npm run build

> hello-world@0.0.1 build /Users/dan/hello-world
> react-scripts build

Successfully generated a bundle in the build folder!

You can now serve it with any static server, for example:
  cd build
  npm install -g http-server
  hs
  open http://localhost:8080

The bundle is optimized and ready to be deployed to production.
hello-world)
```

But what if I don't like  
the default settings?

- To create the config files and move the dependencies into your package.json, you simply ...

npm run eject

## tl;dr

- React is very difficult to configure and brittle.
- So the team has provided us an incredibly simple tool called `create-react-app` that scaffolds the project
- Compile and run in watch mode with `npm run dev`
- To transpile, bundle, minify, lint and run tests for production, you `npm run build`

## Resources for further study

- Dan Abramov's blog post where these screen shots and quotes came from:
  - <http://bit.ly/create-react-app-article>