

Easy Install

```
create  npm create-react-app my-app
app
```

```
delete del src/*
src
```

```
creating type nul > index.css
files    type nul > index.js
```

```
index.js import React from
         'react';
import ReactDOM from
         'react-dom';
import './index.css';
```

Build

```
run server    npm start
```

```
create bundle npm run build
```

Deploy

Setup from Scratch

```
Initia-  npm init
lizing
```

```
Install  npm install react react-dom --
React    save
```

```
Install  npm install webpack webpack-d-
webpack ev-server webpack-cli --save
```

```
Install  npm install babel-core babel--
babel    loader babel-preset-env babel--
         preset-react html-webpack-
         plugin --save-dev
```

Structure

```
index.html type nul > index.html
```

```
App.js     type nul > App.js
```

```
main.js    type nul > main.js
```

```
webpack.conf- type nul > webpackco-
ig.js       nfig.js
```

```
.babelrc   type nul > .babelrc
```

React



Set Compiler, Server & Loader

```
File : webpack.config.js
const path = require('path');
const HtmlWebpackPlugin =
require('html-webpack-plugin');
module.exports = {
  entry: './main.js',
  output: {
    path: path.join(__dirname,
'/bundle'),
    filename: 'index_bundl-
e.js'
  },
  devServer: {
    inline: true,
    port: 8001
  },
  module: {
    rules: [
      {
        test: /\.jsx?$/,
        exclude: /node_mod-
ules/,
        loader: 'babel-lo-
ader',
        query: {
          presets:
['es2015', 'react']
        }
      }
    ]
  },
  plugins: [
```

Set Compiler, Server & Loader (cont)

```
    new HtmlWebpackPlugin({
      template: './index.h-
tml'
    })
  ]
}
```

package.json

```
{
  "name": "reactApp",
  "version": "1.0.0",
  "description": "First React
App",
  "main": "index.js",
  "scripts": {
    /*test: "echo \"Error: no test
specified\" && exit 1"
    REMOVE THIS AND ADD FOLLOWING 2
    LINES */
    "start": "webpack-dev-server
--mode development --open --
hot",
    "build": "webpack --mode
production"
  },
  "keywords": [],
  "author": "KNIGHTblood",
  "license": "ISC",
  "dependencies": {
    "react": "^16.12.0",
    "react-dom": "^16.12.0",
    "webpack": "^4.41.5",
    "webpack-cli": "^3.3.10",
    "webpack-dev-server": "^3.1-
0.1"
  },
  "devDependencies": {
    "babel-core": "^6.26.3",
    "babel-loader": "^8.0.6",
    "babel-preset-env": "-
^1.7.0",
```



By **KNIGHT BLOOD**

(spgusess)

cheatography.com/spgusess/blackvisionlab.com

Not published yet.

Last updated 17th January, 2020.

Page 1 of 2.

Sponsored by **Readable.com**

Measure your website readability!

<https://readable.com>

package.json (cont)

```
"babel-preset-react": "^6.2-4.1",
"html-webpack-plugin": "~3.2.0"
}
```

index.html

```
<!DOCTYPE html>
<html lang = "en">
  <head>
    <meta charset = "UTF-8">
    <title>React App</title>
  </head>
  <body>
    <div id = "app"></div>
    <script src = 'index_bundle.js'></script>
  </body>
</html>
```

App.js

```
import React, { Component } from 'react';
class App extends Component{
  render(){
    return(
      <div>
        <h1>Hello
World</h1>
      </div>
    );
  }
}
export default App;
```

main.js

```
import React from 'react';
import ReactDOM from 'react-dom';
import App from './App.js';
ReactDOM.render(<App />, document.getElementById('app'));
```

.babelrc

```
{
  "presets": ["env", "react"]
}
```

Components are made with jsx

HTML	<div> {'Hello'.toUpperCase()} </div>
JS Like	<div className="hello-upper"> {'Hello'.toUpperCase()} </div>



By **KNIGHT BLOOD**
(spegusess)
cheatography.com/spegusess/blackvisionlab.com

Not published yet.
Last updated 17th January, 2020.
Page 2 of 2.

Sponsored by **Readable.com**
Measure your website readability!
<https://readable.com>