

The minimal React + Webpack 4 + Babel Setup

[f Follow \(https://www.facebook.com/rwieruch\)](https://www.facebook.com/rwieruch)
[t Follow 6k \(https://twitter.com/intent/follow?screen_name=rwieruch\)](https://twitter.com/intent/follow?screen_name=rwieruch)
[g Follow 2038 \(https://github.com/rwieruch\)](https://github.com/rwieruch)

JANUARY 18, 2018

(https://www.pinterest.com/pin/create/button/?url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f&media=https://www.robinwieruch.de/img%2fposts%2fminimal-react-webpack-babel-setup%2fbanner.jpg&description=)



This tutorial is part 2 of 2 in the series.

Part 1: My development setup as a JavaScript web developer (<https://www.robinwieruch.de/developer-setup/>)

Personally I bootstrapped a lot of React projects in the recent time. I always had to setup the project from scratch. Eventually I have created my own boilerplate project on GitHub (<https://github.com/rwieruch/minimal-react-webpack-babel-setup>). As you might know, uncountable React boilerplate projects and repositories were created that way. But the article is not my attempt to advertise yet another React boilerplate project. I had several reasons why I extracted the setup process from another article of mine.

First, I can reuse it for all my other articles of my website whenever there is a React project setup involved. You might be reading this article because you are interested in another article of mine. [f \(https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f\)](https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f) [t \(https://twitter.com/rwieruch\)](https://twitter.com/rwieruch) [p \(https://www.pinterest.com/pin/create/button/?url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f&media=https://www.robinwieruch.de/img%2fposts%2fminimal-react-webpack-babel-setup%2fbanner.jpg&description=\)](https://www.pinterest.com/pin/create/button/?url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f&media=https://www.robinwieruch.de/img%2fposts%2fminimal-react-webpack-babel-setup%2fbanner.jpg&description=)

Second, it helps me to maintain the React setup at one place. It is my single source of truth. Whenever there are updates regarding React, Webpack, Babel or Hot Reloading, I can come back to this one article to keep all other articles updated.

Third, a single source of truth has to be well maintained. When several of my articles reference this one article to set up a React application with Webpack and Babel, I am enforced to maintain it well. People, who search about setting up their React, Webpack and Babel environment, will hopefully always find an up to date version of the article. I really appreciate any feedback, issue reports and improvements for the article.

Fourth, the article is not about the boilerplate project itself. The article is more about teaching people how to setup their own project without a third-party boilerplate project. At some point, you will start to use the tools around your library or framework of choice. In JavaScript you will have to deal with Webpack, Babel et al. and thus it makes sense to learn about them. I hope this article helps you with this adventure.

Last but not least, there is already a great official way introduced by Facebook to start a React project: create-react-app (<https://github.com/facebookincubator/create-react-app>) comes without any build configuration which I can only recommend for anyone who is getting started with React. If you are a beginner, you probably shouldn't bother with a setup of Webpack and Babel. I use create-react-app to teach plain React in my book the Road to learn React (<https://www.robinwieruch.de/the-road-to-learn-react/>). You should take the time to read it before you get started with the tooling around React.

That should be enough said about my motivation behind this article. Let's dive into my personal minimal setup for a React project. This tutorial is up to the recent React 16 and Webpack 4 versions. However, you should be able to make it work with Webpack 3 as well.



Table of Contents

- React Project Setup
- Webpack Setup
- Babel Setup
- React Setup in a Webpack + Babel Project
- Hot Module Replacement

React Project Setup

Before you can get started, you should make sure to have an installed editor and terminal (<https://www.robinwieruch.de/developer-setup/>) on your machine. In addition, you will need an installed version of node with npm (<https://nodejs.org/en/>). Make sure to have setup everything of it before you continue to read.

For any new project, there has to be a folder to allocate the project's configuration but most importantly its source code. This folder usually resides in another folder where all your other projects can be found. That's at least how I do it for my projects. In order to get started with your new project, create its folder on the command line or in your favorite folder/file explorer (e.g. MacOS finder, editor/IDE side bar) and navigate into it.

 (<https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fwww.robinwieruch.de%2Fminimal-react-webpack-babel-setup%2F>)  (<https://twitter.com/sharer/sharer.php?text=The%20minimal%20React%20%2B%20Babel%20Setup%20by%20rwieruch%23ReactJs&url=https%3A%2F%2Fwww.robinwieruch.de%2Fminimal-react-webpack-babel-setup%2F>)

Now you have got the project's folder. Next you can initialize it as npm (<https://docs.npmjs.com/cli/init>) project. By giving it the `-y` shorthand flag, you are telling npm that it should take all the defaults. If you leave the flag out, you are in charge to specify the information about your project manually.

```
npm init -y
```

You can checkout the `package.json` file after initializing your project as npm project. It should be filled with your defaults. If you want to change your defaults, you can see and change them with the following commands on the command line:

```
npm config list

npm set init.author.name "<Your Name>"
npm set init.author.email "you@example.com"
npm set init.author.url "example.com (http://example.com)"
npm set init.license "MIT"
```

After setting up your npm project, you can install node packages (libraries) to your project with npm itself. Once you install a new node package, it should show up in your `package.json` file.

The next step is to create a distribution folder. The folder will be used to serve your application. Serving the app makes it possible to view it in the browser or host it on an external server to make it accessible for everyone.

The whole served application contains only of two files: a `.html` and a `.js` file. While the `.js` file will be generated automatically from all of your JavaScript source files (via Webpack) later, you can already create the `.html` file manually as an entry point for your application.

From root folder (*minimal-react-boilerplate*):

```
mkdir dist
cd dist
touch index.html
```

As a side note: The distribution folder will be everything you need to publish your web app to a hosting server. It will only need the HTML and JS file to serve your app.

The created file should have the following content:

dist/index.html

```
<!DOCTYPE html>
<html>
  <head>
    <title>The Minimal React Webpack Babel Setup</title>
  </head>
  <body>
    <div id="app"></div>
    <script src="/bundle.js"></script>
  </body>
</html>
```

Two important facts about the content:

react-webpack-babel-setup%2f) text=The%20minimal%20React%20%2b%20 by %40rwieruch %23ReactJs&url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f) (https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f) (https://twitter.com/rwieruch/status/971111111111111111)

- the bundle.js file will be a generated file by Webpack (1)
- the id="app" attribute will help our root React component to find its entry point (2)

Therefore our next possible steps are:

- (1) setup Webpack to bundle our source files in one file as bundle.js
- (2) build our first React root component which uses the entry point id="app"

Let's continue with the former step followed by the latter one.

Webpack Setup

You will use Webpack (<https://github.com/webpack/webpack>) as module bundler and build tool. Moreover you will use webpack-dev-server (<https://github.com/webpack/webpack-dev-server>) to serve your bundled app in a local environment. Otherwise you couldn't see it in the browser to develop it. Last but not least, you need the webpack-cli (<https://github.com/webpack/webpack-cli>) node package to configure your Webpack setup in a configuration file later on. Let's install all three node packages by using npm.

From root folder:

```
npm install --save-dev webpack webpack-dev-server webpack-cli
```

Now you should have a *node_modules* folder where you can find your third party dependencies. The dependencies will be listed in the *package.json* file as well, since you used the *--save-dev* flag. Your folder structure should look like the following by now:

Folder structure:

```
- dist
-- index.html
- node_modules
- package.json
```

In the *package.json* file, you can add a start script additionally to the default given scripts to run the webpack-dev-server.

package.json

```
...
"scripts": {
  "start": "webpack-dev-server --config ./webpack.config.js --mode development",
  ...
},
...
```

The script defines that you want to use the webpack-dev-server with a configuration file called *webpack.config.js*. The *mode development* flag just adds default Webpack configurations which came with Webpack 4. You wouldn't need the flag for Webpack 3.

react-webpack-babel-setup%2f)

by %40rwieruch %23ReactJs&url=https%3

Let's create the required *webpack.config.js* file.

react-webpack-

From root folder:

```
touch webpack.config.js
```

You can continue by providing the following content:

webpack.config.js

```
module.exports = {
  entry: './src/index.js',
  output: {
    path: __dirname + '/dist',
    publicPath: '/',
    filename: 'bundle.js'
  },
  devServer: {
    contentBase: './dist'
  }
};
```

Roughly the configuration file says that (1) we want to use the *src/index.js* file as entry point to bundle all of its imported files. (2) The bundled files will result in a *bundle.js* file which (3) will be generated in our already set up */dist* folder. The */dist* folder will be used to serve our app.

What is missing in our project is the *src/index.js* file.

From root folder:

```
mkdir src
cd src
touch index.js
```


src/index.js

```
console.log('My Minimal React Webpack Babel Setup');
```

Folder structure:

```
- dist
-- index.html
- node_modules
- src
-- index.js
- package.json
- webpack.config.js
```

Now you should be able to start your webpack-dev-server.

 ([https://www.facebook.com/sharer/sharer.php?](https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f)

 ([https://twi](https://twitter.com/rwieruch/status/1000000000000000000)

From root folder:
<https://www.robinwieruch.de/minimal-react-webpack-babel-setup/>
 npm start

text=The%20minimal%20React%20%2b%20by%20rwieruch%20%23ReactJs&url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f

You can open the app in a browser (<http://localhost:8080/>). Additionally you should see the `console.log()` in the developer console.

You are serving your app via Webpack now. You bundle your entry point file `src/index.js` as `bundle.js`, use it in `dist/index.html` and can see the `console.log()` in the developer console. For now it is only the `src/index.js` file. But you will import more JS files later on in that file, which will get bundled automatically by Webpack in the `bundle.js` file.

Babel Setup

Babel (<https://babeljs.io/>) enables you writing your code in ES6 (ES2015) (<https://babeljs.io/docs/learn-es2015/>). With Babel the code will get transpiled back to ES5 so that every browser, without having all ES6 features implemented, can interpret it. Babel even takes it one step further. You can not only use ES6 features, but also the next generations of ES.

From root folder:

```
npm install --save-dev babel-core babel-loader babel-preset-env
```

Additionally you might want to use some more experimental features in ES6 (e.g. object spread (<https://github.com/sebmarkbage/ecmascript-rest-spread>)) which can get activated via stages (<https://babeljs.io/docs/plugins/preset-stage-0/>). No worries, even though it is experimental, it is already used in create-react-app by Facebook too.

From root folder:

```
npm install --save-dev babel-preset-stage-2
```

As last step, since you want to use React, you need one more configuration to transform the natural React `.jsx` files to `.js` files. It is for the sake of convenience.

From root folder:

```
npm install --save-dev babel-preset-react
```

Now, with all node packages in place, you need to adjust your `package.json` and `webpack.config.js` to respect the Babel changes. These changes include all packages you have installed.

package.json

f (<https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f>) **🐦** (<https://twitter.com/rwieruch/status/1000000000000000000?text=The%20minimal%20React%20%2b%20babel%20setup%20by%20rwieruch%20%23ReactJs&url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f>)

```

...
"keywords": [],
"author": "",
"license": "ISC",
"babel": {
  "presets": [
    "env",
    "react",
    "stage-2"
  ]
},
"devDependencies": {
...

```

webpack.config.js

```

module.exports = {
  entry: './src/index.js',
  module: {
    rules: [
      {
        test: /\.js$/,
        exclude: /node_modules/,
        use: ['babel-loader']
      }
    ]
  },
  resolve: {
    extensions: ['*', '.js', '.jsx']
  },
  output: {
    path: __dirname + '/dist',
    publicPath: '/',
    filename: 'bundle.js'
  },
  devServer: {
    contentBase: './dist'
  }
};

```

You can start your application again. Nothing should have changed except for that you can use upcoming ECMAScript functionalities for JavaScript now. An optional step would be to extract your Babel configuration in a separate *.babelrc* configuration file.

From root folder:

```
touch .babelrc
```



Now you can add the configuration for Babel, which you have previously added in your *package.json*, in the *.babelrc* file. Don't forget to remove the configuration in the *package.json* afterward. It should be configured at only one place.

.babelrc

```

{
  "presets": [
    "env",
    "react",
    "stage-2"
  ]
}

```

 (https://www.facebook.com/sharer/sharer.php?u=https%3A%2F%2Fwww.robinwieruch.de%2Fminimal-react-webpack-babel-setup%2F)
  (https://twitter.com/rwieruch/status/1000000000000000000)

You are set up to build your first React component now.

React Setup in a Webpack + Babel Project

In order to use React, you need two more node packages. The react and react-dom packages should fix your npm start.

From root folder:

```
npm install --save react react-dom
```

In your `src/index.js` you can implement your first hook into the React world.

`src/index.js`

```
import React from 'react';
import ReactDOM from 'react-dom';

const title = 'My Minimal React Webpack Babel Setup';

ReactDOM.render(
  <div>{title}</div>,
  document.getElementById('app')
);
```

You should be able to see the output in your browser rather than in a developer console now.

`ReactDOM.render` needs two parameters. The first parameter is your JSX. It has to have always one root node. The second parameter is the node where your output should be appended. Remember when we used `<div id="app"></div>` in the `dist/index.html` file? The same id is your entry point for React now.

Hot Module Replacement in React

A huge development boost will give you react-hot-loader (<https://github.com/gaearon/react-hot-loader>) (Hot Module Replacement). It will shorten your feedback loop during development. Basically whenever you change something in your source code, the change will apply in your app running in the browser without reloading the entire page (<https://www.youtube.com/watch?v=xsSnOQynTHs>).

From root folder:

```
npm install --save-dev react-hot-loader
```

You have to add some more configuration to your webpack configuration file.

[f \(https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f\)](https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f)
[t \(https://twitter.com/rwieruch/status/1018111111111111111\)](https://twitter.com/rwieruch/status/1018111111111111111)


```

const webpack = require('webpack');

module.exports = {
  entry: './src/index.js',
  module: {
    rules: [
      {
        test: /\.?(js|jsx)$/,
        exclude: /node_modules/,
        use: ['babel-loader']
      }
    ]
  },
  resolve: {
    extensions: ['*', '.js', '.jsx']
  },
  output: {
    path: __dirname + '/dist',
    publicPath: '/',
    filename: 'bundle.js'
  },
  plugins: [
    new webpack.HotModuleReplacementPlugin()
  ],
  devServer: {
    contentBase: './dist',
    hot: true
  }
};

```

Additionally in the `src/index.js` file you have to define that hot reloading is available and should be used.

src/index.js

```

import React from 'react';
import ReactDOM from 'react-dom';

const title = 'My Minimal React Webpack Babel Setup';

ReactDOM.render(
  <div>{title}</div>,
  document.getElementById('app')
);

module.hot.accept();

```

Now you can start your app again.

From root folder:

```
npm start
```

When you change your title for the React component in the `src/index.js` file, you should see the updated output in the browser without any browser reloading. If you would remove the `module.hot.accept();` line, the browser would perform a reload if something has changed in the code.




<https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f>
<https://twitter.com/rwieruch/status/1000000000000000000>

That's it for a minimal React setup with Babel and Webpack. Let me know your thoughts. Again, you can find the repository on GitHub (<https://github.com/rwieruch/minimal-react-webpack-babel-setup>). You can contribute by creating issues when new versions introduce breaking changes. Even more you can have a direct impact on this article on GitHub as well.

Read more: React Code Style with ESLint + Babel + Webpack (<https://www.robinwieruch.de/react-eslint-webpack-babel/>)

Read more: The minimal Node.js with Babel Setup (<https://www.robinwieruch.de/minimal-node-js-babel-setup/>)

I would like to hear your thoughts :-). Find me on  Twitter (<https://twitter.com/rwieruch>) and  GitHub (<https://github.com/rwieruch>)

Did the article help you? Share it with your friends on social media , support me on  Patreon (<https://www.patreon.com/rwieruch>), become a React developer with my books 

The Road to learn React



Build a Hacker News App along the way. No setup configuration. No tooling. No Redux. Plain React in 190+ pages of learning material. Learn React like **33.000+ readers**.

GET THE BOOK FOR FREE > ([HTTPS://WWW.GETREVUE.CO/PROFILE/RWIERUCH](https://www.getrevue.co/profile/rwieruch))

Or get it from Amazon (<https://amzn.to/2LHjxRB>)

 ([https://www.facebook.com/sharer/sharer.php?](https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f)

 ([https://twi](https://twitter.com/rwieruch/status/1031111111111111111)

[u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f](https://www.robinwieruch.de/minimal-react-webpack-babel-setup/)
73 Comments

text=The%20minimal%20React%20%2b%20ReactJs%20by%20rwieruch%23ReactJs&url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f

 Recommend 18  Tweet  Share

react-webpack-
Sort by Best



Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

**Shad Holland** • a year ago

If you do this today, you will get an error with ReactDOM due to the react-hot-loader. I personally don't like hot reloading. I know it's the cool new thing to do. But I like to make a bunch of changes and save as I go. Then when I am ready to see my work in action, I just type webpack in the console and it builds. Then I spent a huge amount of energy reaching over to my mouse and clicking on the refresh button in my browser.

If you do want to do this without the hot reloading, make sure you install webpack globally.

```
>npm install -g webpack
```

Oh, and great tutorial. I love it when people keep it simple!!!!!!

4 ^ | v • Reply • Share ›

**Robin Wieruch** Mod → Shad Holland • a year ago

Thank you for taking the time to write down your workaround! Often when people run into problems when reading an article, they have a look into the comment section. So thanks for providing this solution! :)

3 ^ | v • Reply • Share ›

**Max K** → Robin Wieruch • 10 months ago

Thanks for the great tutorial. Like Shad Holland mentioned the react-hot-loader isn't working anymore in that configuration. If you don't use the package you can still use the npm start command as presented. -> The app is rendering new after saving the changes. -> Which works great for me. Unfortunately it took me a while to figure out the problem. Would you mind updating your article, so other readers will not run into the same problem when they follow your tutorial step by step.

The changes I made:

- SKIP "npm install --save-dev react-hot-loader"
- SKIP "module.hot.accept();" in index.js
- CHANGE in webpack.config.js:
--> loader: 'react-hot-loader!babel-loader' INTO loader: 'babel-loader'

Thanks and keep up the great work!

1 ^ | v • Reply • Share ›

**samson Ssali** • 7 months ago

After your update to Webpack 4 I took some time to compare it with `create-react-app` cli output and discovered that you missed out `service worker` implementation. Do you mind explaining why? [Seems like this minimalistic set up won't work for Progressive Web Apps]

5 ^ | v • Reply • Share ›

**teoman shipahi** → samson Ssali • 6 months ago

That's why it's called "minimalistic".

5 ^ | v • Reply • Share ›

f (https://www.facebook.com/sharer/sharer.php?

t (https://twi

u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f&text=The%20minimal%20React%20%2b%20Babel%20Setup%20by%20Robin%20Wieruch%20ReactJs&url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f



Great article Robin. Thanks a lot for making things simple.

Just used this with the new version of webpack (4.0.1). After a few tweaks, its now working!

Here they are:

1. Installed webpack-cli (a prerequisite) in addition to the dependencies you installed here

react-webpack-

..... webpack on (pre)require, in addition to the dependencies you installed here

2. version comes with entry point and output preconfigured, so I removed those from webpack.config.js

3. included build script in package.json as "webpack"

4. renamed script src attribute value from bundle.js to main.js (it's what the authors named their bundled file) and its was produced in the dist/

then built (yarn build) -> build was successful but with a warning that I did specify the mode

5. To configure for production and development, I just edited my script section in package.json to:

```
"scripts": {
  "dev": "webpack --mode development",
  "build": "webpack --mode production",
  "start": "webpack-dev-server --progress --colors --config ./webpack.config.js --mode development",
  .....,
},
```

Ran again and everything compiled with success both for production and development.

2 ^ | v • Reply • Share ›



Robin Wieruch Mod → samson Ssali • 7 months ago

Thanks for your comment Samson. The tutorial got updated to Webpack 4 now!

1 ^ | v • Reply • Share ›



samson Ssali → samson Ssali • 7 months ago

Not to mention,

1. in order to use hot-module-reload; i had to configure entypoint as you did here in my webpack.config.js file.

2. Incase anyone is interested in using the `html-webpack-plugin`, it's still work under progress, so it's recommended to install this dependency through a parched fork for now like:

`yarn add html-webpack-plugin@webpack-contrib/html-webpack-plugin`

^ | v • Reply • Share ›



Robin Wieruch Mod → samson Ssali • 7 months ago

I think I saw yesterday a release of it for Webpack 4 on Twitter.

^ | v • Reply • Share ›



samson Ssali → Robin Wieruch • 7 months ago

Could you please provide a link?

^ | v • Reply • Share ›



Himanshu Pandey → samson Ssali • 6 months ago

How to add loaders for image and css file in webbpack

Getting below error when run: npm start

ERROR in ./src/Images/bible.png

Module parse failed: Unexpected character '◆' (1:0)

You may need an appropriate loader to handle this file type.

(Source code omitted for this binary file)

@ ./src/HomePage.js 170:69-98

@ ./src/index.js

@ multi (webpack)-dev-server/client?http://localhost:8080 webpack/hot/dev-server react-hot-loader/patch ./src/index.js

ERROR in ./src/App.css

Module parse failed: Unexpected token (2:5)

You may need an appropriate loader to handle this file type.

```
body {
```

```
| margin: 0;
```

@ ./src/SignInForm.js 15:0-20

f (<https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f>)

tw (<https://twitter.com/rwieruch/status/1000000000000000000>)

text=The%20minimal%20React%20%2b%20react-webpack-babel-setup%20by%20rwieruch%23ReactJs&url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f

react-webpack-

@ ./src/HomePage.js

@ ./src/index.js

@ multi (webpack)-dev-server/client?http://localhost:8080 webpack/hot/dev-server react-hot-loader/patch ./src/index.js

^ | v • Reply • Share ›



Roman Volkov • 7 months ago

Awesome tutorial, but looks like it's time to update it to Webpack 4 :)

I had to add "webpack-cli" as dependency as well as change the webpack config a little (particularly, "mode": "development" to get rid of a warning in the console). Other than that, everything should work fine.

2 ^ | v • Reply • Share ›



Robin Wieruch Mod → **Roman Volkov** • 7 months ago

That sounds great! I will update it :) Can you paste your new start script?

^ | v • Reply • Share ›



Roman Volkov → **Robin Wieruch** • 6 months ago

I submitted a PR with all necessary changes, please take a look when you've got time.

1 ^ | v • Reply • Share ›



Robin Wieruch Mod → **Roman Volkov** • 6 months ago

Perfect :)

^ | v • Reply • Share ›



Roman Volkov → **Robin Wieruch** • 7 months ago

The start script is exactly the same as yours, but it fails unless I add webpack-cli to devDependencies:
"start": "webpack-dev-server --config ./webpack.config.js"

1 ^ | v • Reply • Share ›



Sai Ram • a year ago

I am seeing mostly blog using webpack-dev-server but my question is that it is not a backend framework something like express . I want to use webpack with express as backend but could not find a single blog for that atleast that is good enough to follow.It would be great if you add a article on that too.Also if there is any link to something I mentioned that you know then please share it here Thanks. Finally I also appreciate for your tremendous effort in making react to understand in such easy way.

2 ^ | v • Reply • Share ›



Pankaj Agham → **Sai Ram** • 7 months ago

hey, If you want to use express server, just add server.js file in root folder as shown in img



f (<https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f>) **t** (<https://twitter.com/rwieruch/status/1023123123123123123>)

and just run as

node server

and open new terminal window and run
webpack

1 ^ | v · Reply · Share ›



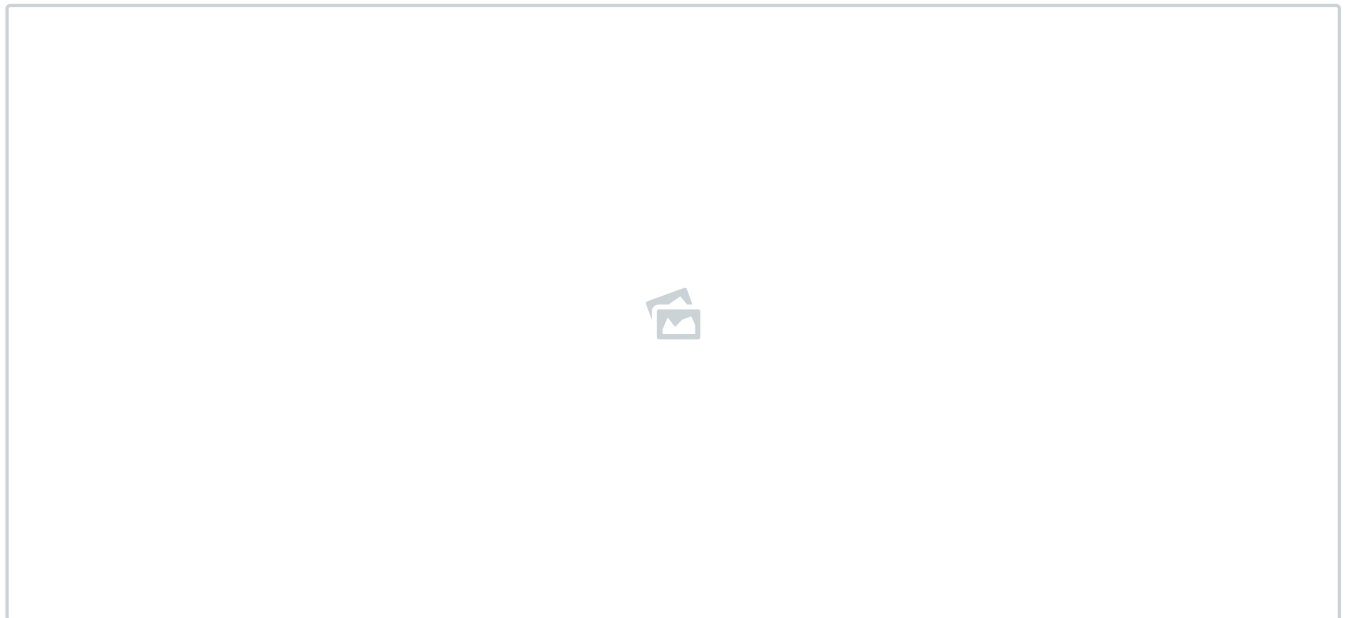
WebX → Sai Ram · 6 months ago

As pointed by [@Pankaj Agham](#) you can simply ditch the webpack-dev-server and use your backend. As long as you are serving the webpack build folder from your backend it shall be fine.

^ | v · Reply · Share ›



Toyosi Akinoso · 3 months ago



Hi, thanks for the article Robin, but for some reason I run into the error as shown above...

1 ^ | v · Reply · Share ›



Robin Wieruch Mod → Toyosi Akinoso · 2 months ago

Check again whether your Babel Setup is alright. Seems like the build cannot identify your JSX syntax of React. Babel is in charge for transpiling it to vanilla JavaScript.

^ | v · Reply · Share ›



Fred Lunjevich · 5 months ago

Great tut Robin. The only problem I get (and I followed the code exactly) is that bundle.js fails to load because the server is outputting a file with MIME type text/html. I can't figure out where to start to fix this.

1 ^ | v · Reply · Share ›



Robin Wieruch Mod → Fred Lunjevich · 2 months ago

You mean for development with webpack-dev-server? Hm, I never ran into this issue. Did you fix it?

1 ^ | v · Reply · Share ›



Fred Lunjevich → Robin Wieruch · 2 months ago

I can't recall now. I think I solved it but may have been using someone else's setup to do that (Maximillian Schwarzmüller has a nice minimalist setup too).



Dominik Kretschmer → Fred Lunjevich · 2 months ago

Had the same problem, try the following:

1. add new npm script "build": "webpack --mode development"

2. npm run build
3. npm start
4. open <http://localhost:8080/>



aguidis • 7 months ago

Very nice article thanks a lot!

I just wanted to ask why does the command "webpack-dev-server --progress --colors --config ./webpack.config.js --mode development" does not generate any bundle.js file in my dist directory ?

But when I check the url "<http://localhost:8080/bundle.js>" my browser seems to display the compiled javascript

1 ^ | v • Reply • Share ›



Jürgen Gutsch • 8 months ago

Hey Robin, this is really a great tutorial. Easy to follow. Many Thanks :-)

1 ^ | v • Reply • Share ›



Robin Wieruch Mod ➔ **Jürgen Gutsch** • 8 months ago

Thank you! :)

^ | v • Reply • Share ›



Adriano Lemos • a year ago

Very simple and to the point. I would put yarn CLI commands along side npm, tough. And I agree that webpack 2 would be nice to see also.

Another thing that bothered me was why use react-hot-loader at all? On a minimal setup this seemed over to me. Nothing serious tough. Great post! Thanks!

1 ^ | v • Reply • Share ›



Robin Wieruch Mod ➔ Adriano Lemos • 8 months ago

Updated to Webpack 3 :)

^ | v • Reply • Share ›



Adriano Lemos → Robin Wieruch · 8 months ago

Great! Thanks for letting me know!

^ | v • Reply • Share ›



Robin Wieruch Mod ➔ Adriano Lemos • a year ago

Agreed, thanks for your feedback. It is definitely on my todo list to cover the next generations of Webpack. The hint for yarn is great as well, I would only have to replace 'npm' correct? Never have used yarn actively myself.

^ | v • Reply • Share ›



Computer Carl → Robin Wieruch • a year ago

'yarn add' will replace 'npm install'. I don't think it's necessary to add both to the tutorial. Noobs (including myself) should already be familiar with NPM and Yarn.

^ | v • Reply • Share ›



Marc Cueto • a year ago

Very nice. Thank you so much. I've been searching around and some samples are too old and it won't work or using
https://www.facebook.com/sharer/sharer.php? (https

u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f) text=The%20minimal%20React%20%2b%20webpack%20tutorial%20by%20robinwieruch%23ReactJs&url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f)



React-Webpack-Babel-Setup%2f)
Robin Wieruch Mod ↗ Marc Cueto · a year ago

That was my intention. Glad that you like it!

^ | v . Reply . Share ,

^ | v · Reply · Share ›



Stephen Wright → Robin Wieruch · 9 months ago

In package.json, you have

```
"babel": {
```

```
"presets": [
```

```
"es2015",
```

```
"react",
```

```
"stage-2"
```

```
]
```

```
},
```

should the last 2 entries be changed into:

```
"babel-preset-stage-2"
```

Good job, enjoyed it.

^ | v · Reply · Share ›



jonni3 · a year ago

Would of been nice to see with webpack 2, I would consider this out of date otherwise

1 ^ | v · Reply · Share ›



Robin Wieruch Mod → jonni3 · 8 months ago

Updated to Webpack 3 (:

^ | v · Reply · Share ›



Robin Wieruch Mod → Robin Wieruch · 7 months ago

Updated to Webpack 4.

^ | v · Reply · Share ›



Jonathan Agoot · 2 years ago

This is a good react-webpack-babel starter. There are a lot of starters out there but are not geared towards keeping lean for newbies.

1 ^ | v · Reply · Share ›



Robin Wieruch Mod → Jonathan Agoot · a year ago

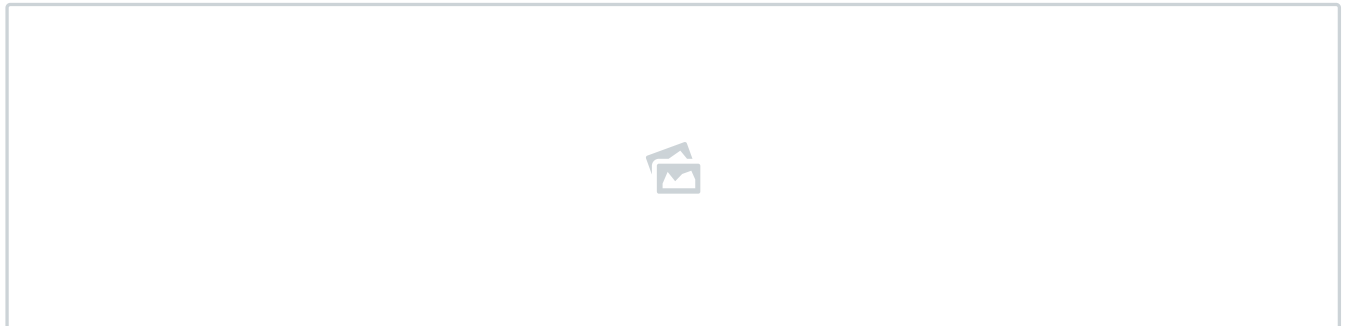
Great that it helps newcomers to get up and started!

^ | v · Reply · Share ›



Altangerel Du · 13 days ago

Hi, i did exactly the same way as this instruction, but having an error on final npm start:



f (https://www.facebook.com/sharer/sharer.php?

t (https://twi

u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f)

text=The%20minimal%20React%20%2b%20babel%20setup%20by%20%40rwieruch%20%23ReactJs&url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-



Roza Jay · 16 days ago

Exactly what I needed!! Thank you!!!

react-webpack-

^ | v · Reply · Share ›



rainecc · 25 days ago

Really great tutorial and very helpful for folks like me who need to understand WHY something is there.

Just one question. When I start up my minimalist app, I see an xhr request from sockjs and then a websocket is opened (which remains in "pending" state). Why are these two requests occurring? I saw these before I started installing the hot loading elements. If not needed I would like to eliminate these connections.

The only change I have made from the tutorial is my webpack-dev-server listens on a different IP address/port (is not localhost).

"start": "webpack-dev-server --port 8080 --host 10.0.2.4 --config ./webpack.config.js --mode development",

^ | v · Reply · Share ›



Eric Coleman · 2 months ago

Great article, this is exactly what I needed to get bootstrapped. Who wants to pull in 2000 libraries with create-react-app? I went even a bit more minimalistic and left out the hot reloading stuff - now I have a nice little skeleton repo for building React apps, and a configuration that I can understand without reading for 6 hours. Thanks!

^ | v · Reply · Share ›



Robin Wieruch Mod → **Eric Coleman** · a month ago

Perfect! I've found that HMR is essential to my workflow. Even more than having a production build. That's why I've included it as default :)

^ | v · Reply · Share ›



Albert221 · 3 months ago

Hey, thanks for this awesome quick start guide!

^ | v · Reply · Share ›



Robin Wieruch Mod → **Albert221** · 3 months ago

Glad that it helped you :)

^ | v · Reply · Share ›



bressonnemesis · 3 months ago

Great tutorial! I'm relatively new to Webpack and am curious that if the "dist" directory is used to serve development code, what directory should be used for production code that gets deployed? Sorry if my question is naive but I'm transitioning from Gulp to WP. Thank you

^ | v · Reply · Share ›



Robin Wieruch Mod → **bressonnemesis** · 3 months ago

Once you integrate a build step, it is okay to generate the build files into the dist folder. Then you just serve this folder for production. But as you have noticed, you can separate those two things in a more elaborate setup: build folder and maybe a public folder (where you have your index.html).

^ | v · Reply · Share ›

[Load more comments](#)

[Subscribe](#) [Add Disqus to your site](#) [Add Disqus](#) [Disqus' Privacy Policy](#) [Privacy Policy](#) [Privacy](#)

[f \(https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f\)](https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f) [twitter \(https://twitter.com/rwieruch/status/971111111111111111\)](https://twitter.com/rwieruch/status/971111111111111111)

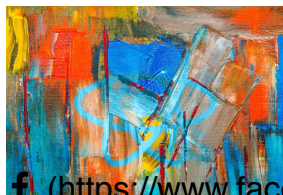
Never miss an article about web development, JavaScript and self-growth.

TAKE PART

- ✓ Join 18.000+ Developers
- ✓ Learn Web Development with JavaScript
- ✓ Tips and Tricks
- ✓ Access Tutorials, eBooks and Courses
- ✓ Personal Development as a Software Engineer

SUBSCRIBE

RECENT POSTS

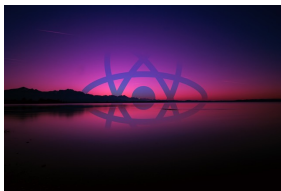


HOW TO PREVENT A RERENDER IN REACT ([HTTPS://WWW.ROBINWIERUCH.DE/REACT-PREVENT-RERENDER-COMPONENT/](https://www.robinwieruch.de/react-prevent-rerender-component/))

[f](https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f) (<https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f>)

[t](https://twitter.com/rwieruch) ([https://twi](https://twitter.com/rwieruch))
text=The%20minimal%20React%20%2b%20react-webpack-babel-setup%20by%20rwieruch%23ReactJs&url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f

INTERSECTION OBSERVER API IN REACT ([HTTPS://WWW.ROBINWIERUCH.DE/REACT-](https://www.robinwieruch.de/react-intersection-observer-api/)

**INTERSECTION-OBSERVER-API/**

(<https://www.robinwieruch.de/react-intersection-observer-api/>)



**GRAPHQL SERVER TUTORIAL WITH APOLLO SERVER AND EXPRESS
([HTTPS://WWW.ROBINWIERUCH.DE/GRAPHQL-APOLLO-SERVER-TUTORIAL/](https://www.robinwieruch.de/graphql-apollo-server-tutorial/))**

(<https://www.robinwieruch.de/graphql-apollo-server-tutorial/>)

ABOUT ME

Hi, I'm Robin - a German and Remote/Berlin based consulting software engineer. Find out more about me (<https://www.robinwieruch.de/about>), work with me (<https://www.robinwieruch.de/work-with-me>), or have a look at my social media profiles below.

© Robin Wieruch

 (<https://www.twitter.com/rwieruch>)  (<https://www.github.com/rwieruch>)

 (<https://www.linkedin.com/in/robin-wieruch-971933a6>)

 (<https://www.producthunt.com/@rwieruch>)  (<https://www.patreon.com/rwieruch>)

 (<https://www.paypal.me/rwieruch>)  (<https://www.facebook.com/rwieruch>)

 (<https://www.instagram.com/rwieruch/>)

[Legal Notice \(/legal\)](#)

 ([https://www.facebook.com/sharer/sharer.php?](https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f)

[u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f](https://www.facebook.com/sharer/sharer.php?u=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f))

 (<https://twitter.com/rwieruch>)

[text=The%20minimal%20React%20%2b%20ReactJs&url=https%3a%2f%2fwww.robinwieruch.de%2fminimal-react-webpack-babel-setup%2f](https://twitter.com/rwieruch))