
No-Sweat™ Eslint and Prettier Setup

These are my settings for ESLint and Prettier

You might like them - or you might not. Don't worry you can always change them.

What it does

- Lints JavaScript and TypeScript based on the latest standards
- Fixes issues and formatting errors with Prettier
- Lints + Fixes inside of html script tags
- Lints + Fixes React via eslint-config-airbnb
- You can see all the rules here - these generally abide by the code written in my courses. You are very welcome to overwrite any of these settings, or just fork the entire thing to create your own.

Project Install

It's recommended you install this once per every project. ESLint used to have global configs, but no longer.

1. If you don't already have a **package.json** file, create one with `npm init -y`.
2. Then we need to install this config

```
1 npm install eslint-config-wesbos
```

4. We need to put our eslint settings in a file in the root of your project. I prefer to use our existing **package.json**, and add an `eslintConfig` property. You can also create a new `.eslintrc` or `.eslintrc.js` file that lives where `package.json` does:

in package.json, add this anywhere top level. Like right under your "scripts" object.

```
1 "eslintConfig": {  
2   "extends": ["wesbos"]  
3 }
```

Or put this in a `.eslintrc` file

```
1 {  
2   "extends": ["wesbos"]  
3 }
```

For TypeScript projects, use `wesbos/typescript`.

```
1 {
2   "extends": ["wesbos/typescript"]
3 }
```

TypeScript users will also need a `tsconfig.json` file in their project. An empty object (`{}`) or my base will do!

5. You can add two scripts to your `package.json` to lint and/or fix:

```
1 "scripts": {
2   "lint": "eslint .",
3   "lint:fix": "eslint . --fix"
4 },
```

6. Now you can manually lint your code by running `npm run lint` and fix all fixable issues with `npm run lint:fix`. You probably want your editor to do this though.

Settings

If you'd like to overwrite eslint or prettier settings, you can add the rules in your `.eslintrc` file. The ESLint rules go directly under `"rules"`.

```
1 {
2   "extends": [
3     "wesbos"
4   ],
5   "rules": {
6     "no-console": 2,
7   }
8 }
```

Prettier Rules

There are only 2 prettier rules included in my config - `singleQuote: true` and `endOfLine: 'auto'`.

If you want custom prettier options, it's recommended to create a `.prettierrc` file in your root directory like so:

```
1 {
2   "singleQuote": true,
3   "endOfLine": "auto",
4   "tabWidth": 4
5 }
```

You can also put this in your ESLint config as a rule like so:

```
1 {
2   "extends": ["wesbos"],
3   "rules": {
4     ... any eslint rules here
5     "prettier/prettier": [
6       "error",
7       {
8         "singleQuote": true,
9         "endOfLine": "auto",
10        "tabWidth": 4
11      },
12    ],
13  }
14 }
```

Note if you are switching to double quotes, you'll also need to add this eslint rule, or they will fight to the death!

```
1 quotes: ["error", "double"];
```

With VS Code

You should read this entire thing. Serious!

Once you have done one, or both, of the above installs. You probably want your editor to lint and fix for you. Here are the instructions for VS Code:

1. Install the ESLint package
2. Now we need to setup some VS Code settings via [Code/File](#) → [Preferences](#) → [Settings](#). It's easier to enter these settings while editing the `settings.json` file, so click the Open (Open Settings) icon in the top right corner:

```
1 // These are all my auto-save configs
2 "editor.formatOnSave": true,
3 // turn it off for JS and JSX, we will do this via eslint
4 "[javascript][javascriptreact][typescript][typescriptreact]": {
5   "editor.formatOnSave": false
6 },
7 // tell the ESLint plugin to run on save
8 "editor.codeActionsOnSave": {
9   "source.fixAll.eslint": true
10 },
```

After attempting to lint your file for the first time, you may need to click on 'ESLint' in the bottom right and select 'Allow Everywhere' in the alert window.

Finally you'll usually need to restart VS code. They say you don't need to, but it's never worked for me until I restart.

With Create React App

1. Run `npx install-peerdeps --dev eslint-config-wesbos`
2. Crack open your `package.json` and replace `"extends": "react-app"` with `"extends": "wesbos"`

With Gatsby

1. Run `npx install-peerdeps --dev eslint-config-wesbos`
2. follow the `Local / Per Project Install` steps above

With WSL

It should work as above.

With JetBrains Products (IntelliJ IDEA, WebStorm, RubyMine, PyCharm, PhpStorm, etc)

If you have previously configured ESLint to run via a File Watcher, turn that off.

If you choose Local / Per Project Install Above

1. Open ESLint configuration by going to File > Settings (Edit > Preferences on Mac) > Languages & Frameworks > Code Quality Tools > ESLint (optionally just search settings for "eslint")
2. Select **Automatic ESLint Configuration**
3. Check **Run eslint -fix on save**

If you choose Global Install

The following steps are for a typical Node / ESLint global installation. If you have a customized setup, refer to JetBrains docs for more ESLint Configuration Options.

1. Open ESLint configuration by going to File > Settings (Edit > Preferences on Mac) > Languages & Frameworks > Code Quality Tools > ESLint (optionally just search settings for "eslint")

-
2. Select **Manual ESLint configuration**
 3. Choose your **Node interpreter** from the detected installations
 4. Select the global **ESLint package** from the dropdown
 5. Leave Configuration File as **Automatic Search**
 6. Check **Run eslint –fix on save**

Ensure the Prettier plugin is disabled if installed.

1. Open Prettier configuration by going to File > Settings (Edit > Preferences on Mac) > Languages & Frameworks > Code Quality Tools > Prettier (optionally just search settings for “prettier”)
2. Uncheck both **On code reformat** and **On save**
3. *Optional BUT IMPORTANT:* If you have the Prettier extension enabled for other languages like CSS and HTML, turn it off for JS since we are doing it through ESLint already.
 1. Make sure the **Run for files** glob does not include `js,ts,jsx,tsx`.
 2. An example glob for styles, config, and markdown. `{**/*,*}. {yaml,css,sass,md}`

With Typescript

Same instructions as above, just make sure you extend `wesbos/typescript` instead of just `wesbos`.

With Yarn

It should just work! Open an issue if not.

With pnpm

It should just work! Open an issue if not.

Issues with ESLint not formatting code

If you experience issues with ESLint not formatting the code or you receive a `Parsing error: Cannot find module '@babel/preset-react'` error message then you need to check that you opened the folder where you installed and configured ESLint directly in VS Code. The correct folder to open will be the one where you installed the `eslint-config-wesbos` npm package and where you created the `.eslintrc` file.

Opening a parent folder or child folder in your code editor will cause ESLint to fail in finding the ESLint npm packages and the formatting won't work.

```
1 your-username
2 |
3 projects
4 |
5 beginner-javascript # <- Open this folder directly in your code
  editor
6 .eslintrc
7 package.json
8 node_modules/
9 exercises/
10 playground/
```