

You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

# What's New in Webpack 5

And should you rush to upgrade?



Malcolm L [Follow](#)

Oct 11, 2020 · 4 min read ★



Photo by [Luca Bravo](#) on [Unsplash](#)

The fifth major webpack release was released yesterday. Almost two years since the last major release (4), this release brings a lot of changes to the most used module bundler in the JavaScript ecosystem. If like me, you started your frontend career prior to the rise of webpack, you remember the pain and frustration of working with tools like gulp and grunt.

Let's take a look at the breaking changes and the improvements that come with the new release of this incredibly popular library.

## What's new in Webpack 5

This new version concentrates on five key areas.

### Faster builds with persistent caching

Slow builds are one of the most common complaints from developers about webpack. The module bundler now offers an opt-in filesystem cache. This should improve our productivity as developers by speeding up our development builds.

### Smaller bundle sizes

Improvements have been made to tree shaking (also known as dead code elimination). While previous versions of webpack were able to remove unused code, version 5 takes it even further. webpack is now able to remove code inside of modules, leading to even smaller bundle sizes. To read more about all of the optimization features of webpack 5, check out the [official documentation](#).

### Better long term caching

After bundle size, the thing that can improve your app loading time the most is caching. With caching, returning visitors to your application experience an almost instantaneous loading experience. With webpack 5, changes made to your code that don't change the minimized version (eg, comments or variable names), do not result in cache invalidation. This means that your users will be able to experience the performance improvements of caching for longer.

### Breaking changes now to allow further improvements later

Some of the changes introduced in this version will not have any visible impact on your application's performance today. Instead, they are meant to allow for new features and improvements in later versions of webpack 5.

These future features include using `http(s)` imports as `module` externals. This will help with the development of micro frontends. To read more about these new and exciting features, check out the official documentation [here](#).

Another breaking change is bumping the minimum Node.js version from `6` to `10.13.0`. Dropping support for older Node.js versions will allow the team to simplify their code, and remove workarounds to support these older versions.

webpack 5 also brings a new `experiments` configuration option with support for WebAssembly, Async Web Assembly, Top Level Await, and outputting your bundle as a

module (previously only possible with rollup).

## Module Federation

This new feature, in short, allows multiple webpack builds to work together. It allows your application to dynamically load code from another application (aka, a different webpack build). The most popular application of module federation is to enable micro-frontend architecture.

Going over module federation in detail is beyond the scope of this article. If you are interested in learning more, be sure to read the official webpack post [here](#).

## Breaking changes

Here's a rundown of the breaking changes that made it into this version and corresponding migration advice.

## Removed Deprecated Items

All items that were marked as deprecated in version 4 have been removed. If your webpack 4 build prints deprecation warnings, be sure to address those before upgrading.

## Changes to built-in plugins

The plugins `IgnorePlugin` and `BannerPlugin` accept different arguments. Read more [here](#).

## Node.JS polyfills removed

In previous versions of webpack, polyfills for native NodeJS libraries like `crypto` were included. These have been removed. Instead, you should use frontend focused libraries, or install the polyfills yourself.

## Should you upgrade now?

This is a personal question, and it really depends on how you use webpack in your project. Most developers who use webpack, use a lot of [plugins](#). You need to make sure that the plugins you use, support this new version.

## What about NextJS?

If you are using [NextJS](#), you can upgrade to webpack 5 by setting the version as a yarn resolution in your package.json. But again, if you have a custom webpack config, you will need to ensure that your config works with webpack 5.

## What about Create-React-App?

The big advantage (and disadvantage for some) of create-react-app, is that there is no official way to customize your webpack config. For those of you using CRA, you will need to wait until react-scripts is upgraded to support webpack 5. According to a contributor, this should happen in Create-React-App version 4.1 ([source](#)).

For more information about migrating from version 4 to 5, be sure to check out the official [migration guide](#).

## Summing it up

This new release of webpack makes me even more excited for the future of frontend development. It's so refreshing to see new features and improvements to a tool that I use every day. We should see it's improvements driving innovation in the community for the next few years.

Most frontend developers don't end up touching webpack very much, and just assume that it "just works". I've said it before, and I'll say it again: I think this is a mistake. Understanding how your build tools work makes you a stronger developer and is invaluable in debugging errors.

While the webpack team will continue to support version 4, by fixing bugs and adding features, for the foreseeable future, they suggest that you upgrade to version 5. With almost any library (except perhaps React), there comes a time when you need to make breaking changes and architectural improvements that rely on those breaking changes.

In short — while dealing with breaking changes is annoying, I don't think it's too much of an ask to make some changes to your application's configuration every two years in exchange for a better and faster build system.

Thanks to Dominik Ferber.

## Sign up for Highlights

By Frontend Digest

The latest and greatest in frontend development [Take a look.](#)

Your email



Get this newsletter

By signing up, you will create a Medium account if you don't already have one. Review our [Privacy Policy](#) for more information about our privacy practices.



Get the Medium app

