

Version 12

To upgrade to version 12, run the following command:

>_ Terminal



```

1  npm install next@12
2
3  yarn add next@12
4
5  pnpm update next@12

```

Upgrading to 12.2

[Middleware](#) - If you were using Middleware prior to `12.2`, please see the [upgrade guide](#) for more information.

Upgrading to 12.0

[Minimum Node.js Version](#) [↗] - The minimum Node.js version has been bumped from `12.0.0` to `12.22.0` which is the first version of Node.js with native ES Modules support.

[Minimum React Version](#) [↗] - The minimum required React version is `17.0.2`. To upgrade you can run the following command in the terminal:

>_ Terminal



```

1  npm install react@latest react-dom@latest
2
3  yarn add react@latest react-dom@latest
4
5  pnpm update react@latest react-dom@latest

```

SWC replacing Babel

Next.js now uses the Rust-based compiler [SWC](#) to compile JavaScript/TypeScript. This new compiler is up to 17x faster than Babel when compiling individual files and up to 5x faster Fast Refresh.

Next.js provides full backward compatibility with applications that have [custom Babel configuration](#). All transformations that Next.js handles by default like styled-jsx and tree-shaking of `getStaticProps` / `getStaticPaths` / `getServerSideProps` have been ported to Rust.

When an application has a custom Babel configuration, Next.js will automatically opt-out of using SWC for compiling JavaScript/Typescript and will fall back to using Babel in the same way that it was used in Next.js 11.

Many of the integrations with external libraries that currently require custom Babel transformations will be ported to Rust-based SWC transforms in the near future. These include but are not limited to:

- Styled Components
- Emotion
- Relay

In order to prioritize transforms that will help you adopt SWC, please provide your `.babelrc` on [this feedback thread](#).

SWC replacing Terser for minification

You can opt-in to replacing Terser with SWC for minifying JavaScript up to 7x faster using a flag in `next.config.js`:

JS next.config.js



```
1 module.exports = {  
2   swcMinify: true,  
3 };
```

Minification using SWC is an opt-in flag to ensure it can be tested against more real-world Next.js applications before it becomes the default in Next.js 12.1. If you have feedback about minification, please leave it on [this feedback thread](#).

Improvements to styled-jsx CSS parsing

On top of the Rust-based compiler we've implemented a new CSS parser based on the one used for the styled-jsx Babel transform. This new parser has improved handling of CSS and now errors when invalid CSS is used that would previously slip through and cause unexpected behavior.

Because of this change invalid CSS will throw an error during development and `next build`. This change only affects styled-jsx usage.

`next/image` changed wrapping element

`next/image` now renders the `` inside a `` instead of `<div>`.

If your application has specific CSS targeting span such as `.container span`, upgrading to Next.js 12 might incorrectly match the wrapping element inside the `<Image>` component. You can avoid this by restricting the selector to a specific class such as `.container span.item` and updating the relevant component with that className, such as ``.

If your application has specific CSS targeting the `next/image <div>` tag, for example `.container div`, it may not match anymore. You can update the selector `.container span`, or preferably, add a new `<div className="wrapper">` wrapping the `<Image>` component and target that instead such as `.container .wrapper`.

The `className` prop is unchanged and will still be passed to the underlying `` element.

See the [documentation](#) for more info.

HMR connection now uses a WebSocket

Previously, Next.js used a [server-sent events](#) connection to receive HMR events. Next.js 12 now uses a WebSocket connection.

In some cases when proxying requests to the Next.js dev server, you will need to ensure the upgrade request is handled correctly. For example, in `nginx` you would need to add the following configuration:

```
1 location /_next/webpack-hmr {
2     proxy_pass http://localhost:3000/_next/webpack-hmr;
3     proxy_http_version 1.1;
4     proxy_set_header Upgrade $http_upgrade;
5     proxy_set_header Connection "upgrade";
6 }
```

If you are using Apache (2.x), you can add the following configuration to enable web sockets to the server. Review the port, host name and server names.

```
<VirtualHost *:443>
# ServerName yourwebsite.local
ServerName "${WEBSITE_SERVER_NAME}"
ProxyPass / http://localhost:3000/
ProxyPassReverse / http://localhost:3000/
# Next.js 12 uses websocket
<Location /_next/webpack-hmr>
```

```
RewriteEngine On
RewriteCond %{QUERY_STRING} transport=websocket [NC]
RewriteCond %{HTTP:Upgrade} websocket [NC]
RewriteCond %{HTTP:Connection} upgrade [NC]
RewriteRule /(.*) ws://localhost:3000/_next/webpack-hmr/$1 [P,L]
ProxyPass ws://localhost:3000/_next/webpack-hmr retry=0 timeout=30
ProxyPassReverse ws://localhost:3000/_next/webpack-hmr
</Location>
</VirtualHost>
```

For custom servers, such as `express`, you may need to use `app.all` to ensure the request is passed correctly, for example:

```
1 app.all('/_next/webpack-hmr', (req, res) => {
2   nextjsRequestHandler(req, res);
3 });
```

Webpack 4 support has been removed

If you are already using webpack 5 you can skip this section.

Next.js has adopted webpack 5 as the default for compilation in Next.js 11. As communicated in the [webpack 5 upgrading documentation](#) Next.js 12 removes support for webpack 4.

If your application is still using webpack 4 using the opt-out flag, you will now see an error linking to the [webpack 5 upgrading documentation](#).

`target` option deprecated

If you do not have `target` in `next.config.js` you can skip this section.

The `target` option has been deprecated in favor of built-in support for tracing what dependencies are needed to run a page.

During `next build`, Next.js will automatically trace each page and its dependencies to determine all of the files that are needed for deploying a production version of your application.

If you are currently using the `target` option set to `serverless`, please read the [documentation on how to leverage the new output](#).