

turbo (experimental)

Warning: These features are experimental and will only work with `next --turbo`.

webpack loaders

Currently, Turbopack supports a subset of webpack's loader API, allowing you to use some webpack loaders to transform code in Turbopack.

To configure loaders, add the names of the loaders you've installed and any options in `next.config.js`, mapping file extensions to a list of loaders:

JS next.config.js



```
1  module.exports = {
2    experimental: {
3      turbo: {
4        loaders: {
5          // Option format
6          '.md': [
7            {
8              loader: '@mdx-js/loader',
9              options: {
10                format: 'md',
11              },
12            },
13          ],
14          // Option-less format
15          '.mdx': ['@mdx-js/loader'],
16        },
17      },
18    },
19  };
```

Then, given the above configuration, you can use transformed code from your app:

```
1 import MyDoc from './my-doc.mdx';
2
3 export default function Home() {
4   return <MyDoc />;
5 }
```

Resolve Alias

Through `next.config.js`, Turbopack can be configured to modify module resolution through aliases, similar to webpack's [resolve.alias](#) configuration.

To configure resolve aliases, map imported patterns to their new destination in `next.config.js`:

`JS` next.config.js



```
1 module.exports = {
2   experimental: {
3     turbo: {
4       resolveAlias: {
5         underscore: 'lodash',
6         mocha: { browser: 'mocha/browser-entry.js' },
7       },
8     },
9   },
10  };
```

This aliases imports of the `underscore` package to the `lodash` package. In other words, `import underscore from 'underscore'` will load the `lodash` module instead of `underscore`.

Turbopack also supports conditional aliasing through this field, similar to Node.js's [conditional exports](#). At the moment only the `browser` condition is supported. In the case above, imports of the `mocha` module will be aliased to `mocha/browser-entry.js` when Turbopack targets browser environments.

For more information and guidance for how to migrate your app to Turbopack from webpack, see [Turbopack's documentation on webpack compatibility](#).