

## Custom Webpack Config

Before continuing to add custom webpack configuration to your application make sure Next.js doesn't already support your use-case:

- CSS imports
- CSS modules
- Sass/SCSS imports
- Sass/SCSS modules
- preact
- Customizing babel configuration

Some commonly asked for features are available as plugins:

- @next/mdx
- @next/bundle-analyzer

In order to extend our usage of **webpack**, you can define a function that extends its config inside **next.config.js**, like so:

```
module.exports = {  
  webpack: (config, { buildId, dev, isServer, defaultLoaders, webpack }) => {  
    // Important: return the modified config  
    return config  
  },  
}
```

The **webpack** function is executed twice, once for the server and once for the client. This allows you to distinguish between client and server configuration using the **isServer** property.

The second argument to the **webpack** function is an object with the following properties:

- **buildId**: String - The build id, used as a unique identifier between builds
- **dev**: Boolean - Indicates if the compilation will be done in development
- **isServer**: Boolean - It's **true** for server-side compilation, and **false** for client-side compilation
- **defaultLoaders**: Object - Default loaders used internally by Next.js:
  - **babel**: Object - Default **babel-loader** configuration

Example usage of **defaultLoaders.babel**:

```
// Example config for adding a loader that depends on babel-loader  
// This source was taken from the @next/mdx plugin source:  
// https://github.com/vercel/next.js/tree/canary/packages/next-mdx  
module.exports = {  
  webpack: (config, options) => {  
    config.module.rules.push({  
      test: /\.mdx/,  
      use: [  

```

```
    options.defaultLoaders.babel,  
    {  
      loader: '@mdx-js/loader',  
      options: pluginOptions.options,  
    },  
  ],  
})  
  
  return config  
},  
}
```

## Related

Introduction to next.config.js: Learn more about the configuration file used by Next.js.