

Ctrl + /v5.0 ▾[View on GitHub](#)

# Webpack and bundlers

Learn how to include Bootstrap in your project using Webpack or other bundlers.



Quadruple your VPS speed  
for the same price! 4CPU,  
8GB RAM, 200GB SSD,  
\$5.50 Boost your project  
now!

ads via Carbon

## On this page

Installing Bootstrap

Importing JavaScript

Importing Styles

Importing Precompiled Sass

Importing Compiled CSS

## Installing Bootstrap

[Install bootstrap](#) as a Node.js module using npm.

## Importing JavaScript

Import [Bootstrap's JavaScript](#) by adding this line to your app's entry point (usually `index.js` or `app.js`):

```
// You can specify which plugins you need
import { Tooltip, Toast, Popover } from 'bootstrap';
```

Alternatively, if you only need just a few of our plugins, you may **import plugins individually** as needed:

```
import Alert from 'bootstrap/js/dist/alert';  
...
```

Bootstrap depends on [Popper](#), which is specified in the `peerDependencies` property. This means that you will have to make sure to add it to your `package.json` using `npm install @popperjs/core`.

## Importing Styles

### Importing Precompiled Sass

To enjoy the full potential of Bootstrap and customize it to your needs, use the source files as a part of your project's bundling process.

First, create your own `_custom.scss` and use it to override the [built-in custom variables](#). Then, use your main Sass file to import your custom variables, followed by Bootstrap:

```
@import "custom";  
@import "~bootstrap/scss/bootstrap";
```

For Bootstrap to compile, make sure you install and use the required loaders: [sass-loader](#), [postcss-loader](#) with [Autoprefixer](#). With minimal setup, your webpack config should include this rule or similar:

```
// ...  
{  
  test: /\.scss$/,  
  use: [{  
    // inject CSS to page  
    loader: 'style-loader'  
  }, {  
    // translates CSS into CommonJS modules  
    loader: 'css-loader'  
  }, {  
    // Run postcss actions  
    loader: 'postcss-loader',  
    options: {
```

```

// `postcssOptions` is needed for postcss 8.x;
// if you use postcss 7.x skip the key
postcssOptions: {
  // postcss plugins, can be exported to postcss.config.js
  plugins: function () {
    return [
      require('autoprefixer')
    ];
  }
}, {
  // compiles Sass to CSS
  loader: 'sass-loader'
}]
}
// ...

```

## Importing Compiled CSS

Alternatively, you may use Bootstrap's ready-to-use CSS by simply adding this line to your project's entry point:

```
import 'bootstrap/dist/css/bootstrap.min.css';
```

In this case you may use your existing rule for `css` without any special modifications to webpack config, except you don't need `sass-loader` just [style-loader](#) and [css-loader](#).

```

// ...
module: {
  rules: [
    {
      test: /\.css$/,
      use: [
        'style-loader',
        'css-loader'
      ]
    }
  ]
}
// ...

```



Designed and built with all the love in the world by the Bootstrap team with the help of our contributors.

Code licensed MIT, docs CC BY 3.0.

Currently v5.0.2.

Analytics by Fathom.

## Links

[Home](#)

[Docs](#)

[Examples](#)

[Themes](#)

[Blog](#)

[Swag Store](#)

## Guides

[Getting started](#)

[Starter template](#)

[Webpack](#)

[Parcel](#)

## Projects

[Bootstrap 5](#)

[Bootstrap 4](#)

[Icons](#)

[RFS](#)

[npm starter](#)

## Community

[Issues](#)

[Discussions](#)

[Corporate sponsors](#)

[Open Collective](#)

[Stack Overflow](#)