

# Bootstrap with Angular

## Prerequisites

To follow this tutorial, you will need:

- [Node.js](#) installed.
- [npm](#) installed.
- An IDE. This demonstration uses [Visual Studio Code](#).
- [Angular CLI](#) installed on your machine. To install the Angular CLI, you need to open a console window. It's recommended to install the CLI globally so that you can start new projects anywhere. Then, run **npm install -g @angular/cli@latest**.

## Create a Project

The next step is to build the project. Move to the directory that you wish to create it in, and type the following command: **ng new hello-world**

You can replace **hello-world** with your own project ID.

Running this command will load the CLI and prompt you to answer the following questions:

1. **“Add Angular Routing?”**: Here, you'll choose if you're building an app with multiple pages.
2. **“Which stylesheet format would you like to use?”**: Here, select your preferred CSS platform.

Then, navigate to the project folder. Switch into it using the **cd** command to access your project.

When the project is ready and you're in the correct folder, start it by running the following: **ng serve --open**

## Get Bootstrap Ready and Installed

If you're running Angular already, you'll need to open a new console window to manage the installation. Then, enter the following: **npm install --save bootstrap**

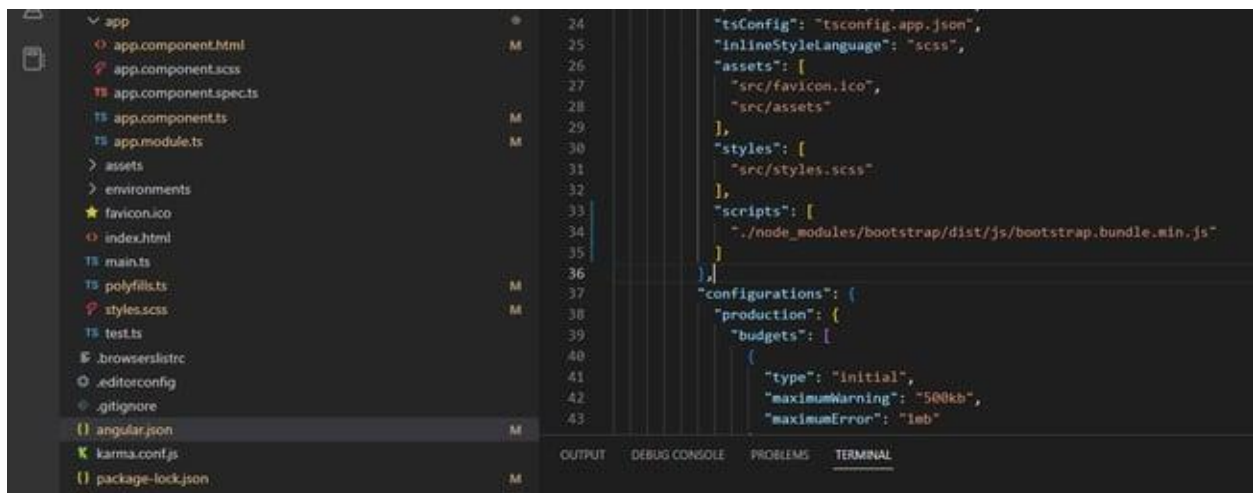
Wait for the installation to finish, which should only take a few minutes.

Next, you need to import Bootstrap CSS.

In your application's global **scss** file — which will most likely be **styles.scss** — import the Bootstrap file using this command: **@import "~bootstrap/dist/css/bootstrap.css";**

This will include the Bootstrap 5 styling, allowing you to use all the CSS features that Bootstrap provides.

If you don't want to import the **Bootstrap.css** file, you can also include it here under **styles**, below **src/styles.scss**:



Angular won't recognize it right away if it's running. So, close the page and stop **ng serve --open** by typing **Ctrl+C**. This will cause the node to terminate the Angular application.

Then, type **ng serve --open** to get it started again.

## Build a Modal

Now that you've installed Bootstrap, let's see how it works with a small modal. Angular generates an example app with a single main component under the app folder.

Click on the app.component.html file. You'll see a default template. Erase the code that's there and add the following code to create a button and modal:

```
<div role="main">

  <button (click)="trigger()" data-toggle="modal" data-target="#helloModal"
type="button" class="btn btn-primary">Click Me!</button>

</div>

<div #helloModal class="modal fade" id="helloModal" role="dialog" aria-
labelledby="helloModalLabel" aria-hidden="true">

  <div class="modal-dialog" role="document">

    <div class="modal-content">

      <div class="modal-header">

        <h5 class="modal-title" id="helloModalLabel">Greetings!</h5>

        <button (click)="trigger()" type="button" class="close" data-dismiss="modal"
aria-label="Close">

          <span aria-hidden="true">&times;</span>

        </button>

      </div>

      <div class="modal-body">

        Hello, World! I'm a modal!

      </div>

      <div class="modal-footer">

        <button (click)="trigger()" type="button" class="btn btn-secondary" data-
dismiss="modal">Close</button>
```

```
</div>
</div>
</div>
</div>
```

Open the **app.component.html** file in the same folder, and add the following additional imports after **Component**:

```
import { Component, ElementRef, ViewChild, AfterViewInit } from
'@angular/core';
import * as bootstrap from 'bootstrap';
```

Now you're ready to create a functional modal. In **app.component.ts**, add these methods and variables to the class:

```
export class AppComponent implements AfterViewInit {

  @ViewChild('helloModal') helloEl?: ElementRef;

  modal?: bootstrap.Modal;

  ngAfterViewInit() {

    this.modal = new bootstrap.Modal(this.helloEl?.nativeElement, { });

  }

  trigger() {

    this.modal?.toggle();

  }
```

}

