

1 Background

This document was composed in early August 2017 for the purpose of explaining how to incorporate Bootstrap version 4 into an Angular 4.x application that was initially scaffolded using Angular-CLI.

A lot of the information in this document is available in bits and pieces on the internet, but, as far as I can tell, has never been gathered into one location for the purposes explaining (in sufficient detail) what steps to perform and in what order to insure that Bootstrap V4 can be used in an Angular 4.x application that was initially built using Angular-CLI.

2 Assumptions

- You performed the initial scaffolding of your Angular 4.x applications using Angular-CLI with the following options:
 - `--style=scss`
 - `--routing`
- You run the application using the following Angular-CLI command:
 - `ng serve --open --verbose --progress --extract-css --vendor-chunk --prod`
- You want to style your application using Bootstrap 4
- (optionally) You want to use Bootstrap navigation
- (optionally) You want to use one of the Bootstrap widgets (e.g., accordion)

3 Tools Used

- Angular-CLI 1.2.6
- Angular 4.3.2
- Bootstrap 4.0.0-alpha.6
- jQuery 3.2.1
- tether 1.4.0
- @ng-bootstrap/ng-bootstrap 1.0.0-alpha.29

4 Installing

The following commands were used *after* the initial Angular application was created via the usual Angular-CLI `ng new` command.

```
npm install -S bootstrap@next
npm install -S jquery
npm install -S tether
npm install -S @ng-bootstrap/ng-bootstrap
```

The above set of commands is one of things I found to be incomplete in most of the other places I searched on the internet on how to use Bootstrap V4 with Angular 4. In particular, the lack of any mention of the need to install *tether* was most puzzling to me. Without it, my navigation bar did not function correctly.

5 package.json

After the necessary additional packages have been installed, your package.json file should include the following (NOTE: this is just a small subset of the full file):

```
"@angular/animations": "^4.0.0",
"@angular/common": "^4.0.0",
"@angular/compiler": "^4.0.0",
"@angular/core": "^4.0.0",
"@angular/forms": "^4.0.0",
"@angular/http": "^4.0.0",
"@angular/platform-browser": "^4.0.0",
"@angular/platform-browser-dynamic": "^4.0.0",
"@angular/router": "^4.0.0",
"@ng-bootstrap/ng-bootstrap": "^1.0.0-alpha.29",
"bootstrap": "^4.0.0-alpha.6",
"core-js": "^2.4.1",
"jquery": "^3.2.1",
"rxjs": "^5.0.0",
"tether": "^1.4.0",
"zone.js": "^0.8.0"
```

Note: I tend to leave the "Angular stuff" with version indicators of "^4.0.0" which means, on a fresh install, or one in which you have removed your node_modules directory and deleted package-lock.json, any subsequent npm install will get the latest version.

6 Configuring

The following changes must be made to ensure that Bootstrap 4 is included when the application is built and run.

6.1.1 .angular-cli.json

```
"styles": [
  "styles.scss"
],
"scripts": [
  "../node_modules/jquery/dist/jquery.min.js",
  "../node_modules/tether/dist/js/tether.min.js",
  "../node_modules/bootstrap/dist/js/bootstrap.min.js"
],
```

6.1.2 styles.css

Put the following at the top:

```
@import '~bootstrap/scss/bootstrap.scss';
```

6.1.3 app.module.ts

```
import {NgbModule} from '@ng-bootstrap/ng-bootstrap';

imports: [
  BrowserModule,
  NgbModule.forRoot(),
  AppRoutingModule
]
```

7 Navigation Bar

Bootstrap version 4 navigation bars are significantly different than the equivalent navigation bar that might have been designed for use in Bootstrap Version 3 (e.g., 3.3.7 at the time this document was first written).

The good news, is, some things are easier, such as changing the background color of the navigation bar.

The navigation bar in this example assumes:

- Navigation is performed using Angular 4.x navigation (remember, as stated earlier, we assume you specified the "—routing" option when you initially scaffolded the Angular application).
- Navigation is contained in its own Angular component.
- There are three components that need to be navigated too:
 - Home
 - About
 - Contact
- The default landing page should be the Home component page.

Given the above assumptions, the navigation.component.html looks like this:

```
<div class="container-fluid">
  <nav class="navbar navbar-toggleable-md navbar-inverse fixed-top bg-primary">
    <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse"
      data-target="#navbarCollapse" aria-controls="navbarCollapse"
      aria-expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <a class="navbar-brand" href="#">Angular Bootstrap V4</a>
    <div class="collapse navbar-collapse" id="navbarCollapse">
      <ul class="navbar-nav mr-auto">
        <li class="nav-item" routerLinkActive="active">
          <a class="nav-link" [routerLink]="['/home']">Home</a>
        </li>
        <li class="nav-item" routerLinkActive="active">
          <a class="nav-link" [routerLink]="['/about']">About</a>
        </li>
        <li class="nav-item" routerLinkActive="active">
          <a class="nav-link" [routerLink]="['/contact']">Contact</a>
        </li>
        <li class="nav-item" routerLinkActive="active">
          <a class="nav-link" [routerLink]="['/accordion']">Accordion</a>
        </li>
      </ul>
    </div>
  </nav>
</div>
```

Notice the following in the above HTML markup:

- The navigation is within a fluid container to allow it to stretch to the full width of the device.
- The blue background color is achieved by specifying `bg-primary`. In Bootstrap version 3.x, you would have to write overrides in CSS (or LESS or SCSS) to achieve the same effect.
- Each `` is decorated with a `nav-item` class.
- Each `<a>` is decorated with a `nav-link` class.
- When displayed on small device or in a narrow browser window, the navigation items rendered in a "hamburger stack":

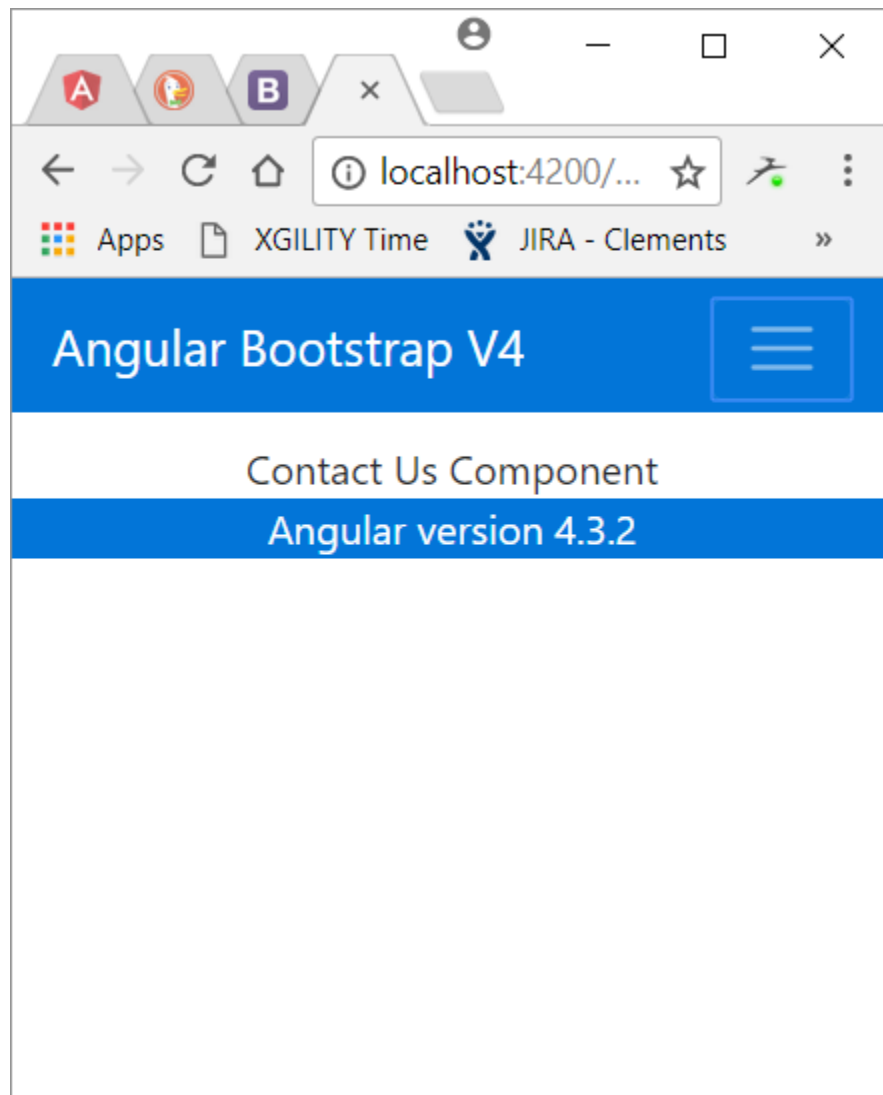


Figure 1 Navigation rendered as a Hamburger stack

8 Accordion

One of the things that ng-bootstrap brings to the table is the ability to use an accordion structure.

The example application contains a component, *AccordionExampleComponent*, that renders three items in an accordion type of user interface.

Clicking on the title of each item exposes the body. Clicking on another title first closes any other item that might be open before exposing the body of the one you just clicked on. If you wish to change this behavior to allow multiple items to be open at the same time, change this line inside `accordion-example.component.html` from:

```
<ngb-accordion [closeOthers]="true">
```

To this:

```
<ngb-accordion [closeOthers]="false">
```

9 Example Code Location

The example code discussed in this document can be found on GitHub [here](#)