These are my notes as I was going through the video tutorials (see the README.md for link) that I viewed as I was creating this project in WebStorm 2024.3 using Angular 19 during December 2024.

* Tutorial #4
* Screen shot of example JSON entry for a house listing.
* A screenshot of a computer program

  Description automatically generated
* Services
  + Unlike the tutorial, I prefer to put all services in a separate directory named *services*, which is where I put the *HousingService* created during the tutorial.
  + I added my own service, AngularVersionInformation, which allows me to display the version of Angular (e.g., 19.0.3) wherever I want to see it.

**Using Bootstrap SCSS style in an Angular application:**

You can use **Bootstrap styles in Angular via SCSS**, and this approach offers better control over customization. Instead of importing the precompiled Bootstrap CSS file, you can include the source SCSS files in your Angular project. This allows you to customize Bootstrap variables and include only the styles you need.

**Steps to Use Bootstrap Styles with SCSS in Angular**

* 1. **Install Bootstrap via npm**: Install Bootstrap and its SCSS source files using npm:
     + npm install bootstrap
  2. **Import Bootstrap SCSS into Your Global Styles**: Open the src/styles.scss file (or create it if it doesn't exist) and import Bootstrap's SCSS file:

*// Import Bootstrap variables and mixins*

@import 'node\_modules/bootstrap/scss/functions';

@import 'node\_modules/bootstrap/scss/variables';

@import 'node\_modules/bootstrap/scss/mixins';

*// Optionally override Bootstrap variables here*

$primary: #007bff; $font-size-base: 1rem;

*// Import the rest of Bootstrap*

@import 'node\_modules/bootstrap/scss/bootstrap';

* 1. **Customize Bootstrap Variables (Optional)**:
     + Before importing the main bootstrap file, you can override Bootstrap’s SCSS variables (e.g., $primary, $font-size-base) to fit your design needs.
     + For a list of available variables, refer to the official Bootstrap documentation.
  2. **Include Bootstrap in Component Styles (Optional)**:
     + While Bootstrap styles are typically added globally via styles.scss, you can also use SCSS partials to include specific Bootstrap styles in individual Angular components:

scss

Copy code

*// In component's SCSS file (e.g., my-component.component.scss)* @import 'node\_modules/bootstrap/scss/buttons';

* 1. **Optimize the Build**:
     + By importing only the necessary parts of Bootstrap (e.g., buttons, forms, grids), you can reduce the CSS file size, improving performance.

**Benefits of Using Bootstrap with SCSS in Angular:**

* 1. **Customization**: Override variables to match your application's design system without modifying Bootstrap's source files.
  2. **Modular Imports**: Include only the styles you need, reducing the CSS footprint.
  3. **Integration with Angular**: SCSS works seamlessly with Angular's component-based architecture, allowing styles to be scoped at the component level.

**Example: Customizing Buttons**

You can customize button styles by overriding variables and importing only the button styles:

scss

Copy code

$btn-primary-bg: #ff5722; *// Change primary button background color* $btn-primary-border: #ff5722; *// Change border color* @import 'node\_modules/bootstrap/scss/buttons';

This approach allows you to blend Bootstrap's functionality with Angular's flexibility and modularity. Let me know if you need help setting up specific features!

Bootstrap provides a wide range of SCSS partials that you can import into your project. These partials are organized modularly, allowing you to import only the parts you need, which is especially useful for customizing your styles and optimizing your CSS output.

Here’s a breakdown of the key SCSS partials available in Bootstrap:

**1. Core Utilities**

* 1. \_functions.scss: Contains utility functions (e.g., color-contrast, shade-color).
  2. \_variables.scss: Defines all of Bootstrap’s customizable variables (e.g., colors, spacing, typography).
  3. \_mixins.scss: Provides reusable SCSS mixins for common patterns (e.g., gradients, media queries).

**2. Grid System**

* 1. \_grid.scss: Defines the grid system (containers, rows, columns, responsive breakpoints).

**3. Reboot**

* 1. \_reboot.scss: Resets and normalizes browser default styles.

**4. Base Components**

* 1. \_root.scss: Defines CSS variables for theme and customization.
  2. \_type.scss: Typography styles (e.g., headings, body text).
  3. \_images.scss: Utility classes for handling responsive images.
  4. \_code.scss: Styles for inline and block code elements.
  5. \_tables.scss: Table styling.
  6. \_forms.scss: Base form styling, including inputs, checkboxes, and more.

**5. Components**

Each Bootstrap component has its own SCSS partial, allowing fine-grained control:

* 1. \_buttons.scss: Styles for buttons and button groups.
  2. \_dropdown.scss: Dropdown menus.
  3. \_modal.scss: Modal windows.
  4. \_alerts.scss: Alert messages.
  5. \_nav.scss: Navigation styles, including tabs and pills.
  6. \_navbar.scss: Navbars.
  7. \_card.scss: Card components.
  8. \_progress.scss: Progress bars.
  9. \_list-group.scss: List groups.

**6. Helpers**

* 1. \_utilities.scss: General-purpose utility classes (e.g., margin, padding, color).
  2. \_helpers.scss: Helper functions and classes.

**7. Custom Components**

For advanced features:

* 1. \_carousel.scss: Carousel components.
  2. \_popover.scss: Popover functionality.
  3. \_tooltip.scss: Tooltips.

**How do import selective pieces of Bootstrap into an Angular application:**

To import only ***buttons*** and ***forms*** into your project, you can do the following in styles.scss:

@import 'node\_modules/bootstrap/scss/functions';

@import 'node\_modules/bootstrap/scss/variables';

@import 'node\_modules/bootstrap/scss/mixins';

@import 'node\_modules/bootstrap/scss/buttons';

@import 'node\_modules/bootstrap/scss/forms';