Button Component

Responsive Buttons built with the latest Bootstrap 5. Buttons provide predefined styles (warning, info, danger} for multiple button types: outline, rounded, social, floating, fixed, tags, etc.

# **Button Types**

Bootstrap includes several predefined button styles, each serving its own semantic purpose, with a few extras thrown in for more control.



These properties are handled with the following enum type as a component property:

export enum ButtonType {

  Primary = 'primary',

  Secondary = 'secondary',

  Success = 'success',

  Danger = 'danger',

  Warning = 'warning',

  Info = 'info',

  Light = 'light',

  Dark = 'dark',

  Link = 'link',

}

Since there may be different button types on the same page, this feature should be treated as a component input, not an option. For more information see the [link](https://getbootstrap.com/docs/5.0/components/buttons/#examples).

# **Sizes**

Different sizes can be added to a button with btn-lg and btn-sm classes.This property should be treated as a component property, just like button types. For more information you can check this [link](https://getbootstrap.com/docs/5.0/components/buttons/#sizes).





export enum ButtonSize {

  Default = '',

  Large = 'btn-lg',

  Small = 'btn-sm'}

# **Outline**

With Bootstrap 5, a button can be configured as outlined using any of the .btn-outline-\* prefixes. For general view integrity, this property should be treated as a component option, since some of the buttons on a page are outlined and others cannot be in the default button.

export interface ButtonOptions {

  /\*\*

   \* Replaces the default modifier classes with the .btn-outline-\* ones to remove all

   \* background images and colors on any button.

   \*

   \* @link https://getbootstrap.com/docs/5.0/components/buttons/#outline-buttons

   \*/

  isOutline: boolean;

},

Depending on whether this component option is true/false, prefixed classes should be applied to the button html in the component template. The following solution may be suitable as a proposal:

export type KeyValuePair = { [type: string]: string };

export type KeyValuePairSelector = (predicate: boolean) => KeyValuePair;

export const getButtonTypesClassMap: KeyValuePairSelector = (outline) => {

  return {

    primary: outline ? 'btn-outline-primary' : 'btn-primary',

    secondary: outline ? 'btn-outline-secondary' : 'btn-secondary',

    success: outline ? 'btn-outline-success' : 'btn-success',

    danger: outline ? 'btn-outline-danger' : 'btn-danger',

    warning: outline ? 'btn-outline-warning' : 'btn-warning',

    info: outline ? 'btn-outline-info' : 'btn-info',

    light: outline ? 'btn-outline-light' : 'btn-light',

    dark: outline ? 'btn-outline-dark' : 'btn-dark',

    link: 'btn-link',

  };

};

The class mapping to be returned by the outline option from this function can be stored in a variable and used in the template together with "ButtonTypes".

# Events

The button only contains the click event. Since it may be desired to control component changes when a button is clicked, event binding should be done in the same zone as NgZone.