

Explorando o parâmetro do `@HostBinding` - Desabilitando Botão no Angular 🚀

O decorador `@HostBinding()` permite que a gente **vincule** propriedades de um elemento HTML diretamente dentro de uma diretiva. Podemos explorar diferentes maneiras de desabilitar um botão e controlar seu comportamento.

1 Exemplo Básico - Desabilitando um Botão com `@HostBinding`

Aqui, criamos uma diretiva para **desativar o botão por padrão**.

♦ HTML

```
<button appDisableDirective>Botão com Diretiva</button>
```

♦ Diretiva `DisableDirective`

```
import { Directive, HostBinding } from '@angular/core';

@Directive({
  selector: '[appDisableDirective]'
})
export class DisableDirective {
  @HostBinding('disabled') isDisabled = true; // O botão começa desabilitado
}
```

✅ O que acontece?

O botão ficará **desativado** (**`disabled = true`**) por padrão assim que a diretiva for aplicada.

2 Exemplo Dinâmico - Desativar com Tempo (setTimeout)

Podemos **habilitar o botão após alguns segundos**.

◆ Diretiva **DisableDirective**

```
import { Directive, HostBinding } from '@angular/core';

@Directive({
  selector: '[appDisableDirective]'
})
export class DisableDirective {
  @HostBinding('disabled') isDisabled = true; // Começa desativado

  constructor() {
    setTimeout(() => {
      this.isDisabled = false; // Habilita o botão após 3 segundos
    }, 3000);
  }
}
```

✓ O que acontece?

Assim que o botão for renderizado, **ele estará desativado por 3 segundos** e depois será ativado automaticamente.

3 Exemplo com Input - Desabilitar ao Digitar

Podemos **desabilitar um botão enquanto o campo de input estiver vazio**.

◆ HTML

```
<input type="text" #meuInput (input)="checkInput(meuInput.value)"
placeholder="Digite algo...">
<button appDisableDirective [isDisabled]="isButtonDisabled">Enviar</button>
```

◆ Diretiva **DisableDirective**

```
import { Directive, HostBinding, Input } from '@angular/core';

@Directive({
  selector: '[appDisableDirective]'
})
export class DisableDirective {
  @HostBinding('disabled') @Input() isDisabled = true; // Vinculado a uma
  entrada
}
```

◆ Componente Controlador

```
import { Component } from '@angular/core';

@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.scss']
})
export class AppComponent {
  isButtonDisabled = true;

  checkInput(value: string) {
    this.isButtonDisabled = value.trim() === ""; // Desabilita se estiver vazio
  }
}
```

✓ O que acontece?

O botão só será habilitado **quando o usuário digitar algo no campo de input**.

4 Exemplo com Evento **click** - Alternando o Estado

Podemos usar um botão para **alternar entre ativado e desativado**.

◆ HTML

```
<button appDisableDirective (click)="toggleDisabled()">Clique para Alternar</button>
```

◆ Diretiva **DisableDirective**

```
import { Directive, HostBinding, HostListener } from '@angular/core';

@Directive({
  selector: '[appDisableDirective]'
})
export class DisableDirective {
  @HostBinding('disabled') isDisabled = false;

  @HostListener('click') toggleDisabled() {
    this.isDisabled = !this.isDisabled; // Alterna entre ativado e desativado
  }
}
```

✅ O que acontece?

Cada vez que o botão for clicado, **ele será desativado ou ativado alternadamente.**

5 Exemplo com Estilização - Alterando Cor ao Desativar

Podemos mudar o estilo do botão quando ele for desativado.

◆ Diretiva **DisableDirective**

```
import { Directive, HostBinding } from '@angular/core';

@Directive({
  selector: '[appDisableDirective]'
})
export class DisableDirective {
  @HostBinding('disabled') isDisabled = true;
  @HostBinding('style.backgroundColor') bgColor = 'gray';
}
```

```
constructor() {  
  setTimeout(() => {  
    this.isDisabled = false;  
    this.bgColor = 'green';  
  }, 3000);  
}
```

✓ O que acontece?


O botão começa **cinza e desativado**, mas **após 3 segundos, ele fica verde e ativado**.

Conclusão

O `@HostBinding()` permite **controlar diretamente propriedades e estilos dos elementos**. Podemos usar isso para: ✓ **Desativar botões automaticamente**

✓ **Criar interações dinâmicas**

✓ **Melhorar a usabilidade e experiência do usuário**

 **Agora você pode usar o `@HostBinding` de várias formas para desabilitar botões dinamicamente no Angular!**