

View Encapsulation Emulated no Angular

O Angular utiliza o **View Encapsulation** para controlar como os estilos de um componente são aplicados. O modo **Emulated**, que é o padrão, encapsula os estilos de um componente, aplicando-os apenas ao DOM daquele componente. Isso é feito simulando o escopo de CSS por meio de atributos personalizados dos elementos do componente.

Como Funciona o **Emulated**?

- Adiciona atributos exclusivos (como **_ngcontent-xyz**) aos elementos do DOM do componente e ao seletor CSS correspondente.
- Os estilos definidos em um componente não vazam para outros componentes.
- Não requer suporte nativo de Shadow DOM.

Exemplo Prático

1. Arquivo CSS do Componente (**example.component.scss**):

```
h1 {  
  color: blue;  
}  
  
p {  
  font-size: 18px;  
  color: gray;  
}
```

2. Arquivo HTML do Componente (**example.component.html**):

```
<div>  
  <h1>Título do Componente</h1>  
  <p>Este parágrafo é estilizado apenas dentro deste componente.</p>  
</div>
```

3. Componente Angular (**example.component.ts**):

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

```
@Component({
  selector: 'app-example',
  templateUrl: './example.component.html',
  styleUrls: ['./example.component.scss'], // Estilos encapsulados
  encapsulation: ViewEncapsulation.Emulated, // Configuração explícita (opcional,
  pois é o padrão)
})
export class ExampleComponent {}
```

4. Outro Componente na Aplicação (**another.component.html**):

```
<div>
  <h1>Outro título</h1>
  <p>Este parágrafo não é afetado pelos estilos do componente `example`.</p>
</div>
```

Como os Estilos São Aplicados no DOM

Após a renderização no navegador, o Angular aplica atributos como **_ngcontent-xyz** para criar um escopo exclusivo:

DOM Gerado

```
<div _ngcontent-abc="">
  <h1 _ngcontent-abc="">Título do Componente</h1>
  <p _ngcontent-abc="">Este parágrafo é estilizado apenas dentro deste
  componente.</p>
</div>
```

CSS Gerado

```
h1[_ngcontent-abc] {
  color: blue;
}
```

```
p[_ngcontent-abc] {
  font-size: 18px;
  color: gray;
}
```

Esses atributos evitam que os estilos vazem para outros componentes.

O Que Pode Ser Feito com **ViewEncapsulation.Emulated**

1. Estilos Isolados por Componente:

- Os estilos são limitados ao componente, garantindo modularidade e evitando conflitos.

2. Personalização Local:

- Permite aplicar estilos específicos para um componente sem impactar o restante da aplicação.

3. Compatibilidade Ampla:

- Funciona em navegadores que não suportam nativamente o Shadow DOM.

4. Simulação de Shadow DOM:

- Oferece escopo de estilo semelhante ao Shadow DOM sem usar recursos nativos.

Vantagens

- **Encapsulamento:** Os estilos são limitados ao escopo do componente.
- **Controle:** Evita vazamentos de estilo para outros componentes.
- **Padrão:** Não requer configuração adicional.

Desvantagens

- **CSS Global Ainda Pode Influenciar:** Estilos globais definidos em `styles.scss` ainda podem afetar os componentes.
- **Overhead de Atributos:** Adiciona atributos personalizados ao DOM, o que pode aumentar o tamanho final do HTML.

Quando Usar **ViewEncapsulation.Emulated**

- Quando você deseja o comportamento padrão de encapsulamento.
- Ao criar componentes reutilizáveis que não devem ser afetados por estilos globais.

3. Para aplicações que precisam funcionar em navegadores sem suporte ao Shadow DOM.

Comparação com Outros Modos

Modo	Descrição	Encapsulamento	Compatibilidade
Emulated	Encapsula estilos simulando Shadow DOM.	Sim	Compatível com todos os navegadores.
None	Desativa o encapsulamento, tornando os estilos globais.	Não	Compatível com todos os navegadores.
Shadow DOM	Usa o Shadow DOM nativo para encapsular estilos.	Sim	Requer suporte ao Shadow DOM.

Conclusão

O **ViewEncapsulation.Emulated** é a escolha padrão no Angular porque equilibra encapsulamento e compatibilidade. É ideal para a maioria dos cenários, garantindo que os componentes sejam estilizados de maneira isolada, sem dependências de suporte a recursos nativos de navegadores.