

Como Referenciar Componentes de Outros Módulos no Angular 19

No Angular, podemos organizar a aplicação em **módulos** para modularizar e reutilizar funcionalidades. Um componente declarado em um módulo não é automaticamente visível para outros módulos. Para utilizá-lo em outro módulo, você precisa exportá-lo corretamente.

Passos para Referenciar Componentes de Outros Módulos

1. Criar o Módulo com o Componente Desejado

Suponha que você tenha um componente chamado `Comp2Component` no módulo chamado `ModuloExterno`.

Arquivo: `modulo-externo.module.ts`

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common'; // Módulo básico para funcionalidades Angular
import { Comp2Component } from './comp2/comp2.component'; // Importa o componente

@NgModule({
  declarations: [
    Comp2Component // Declara o componente dentro deste módulo
  ],
  imports: [
    CommonModule // Importa funcionalidades básicas do Angular, como diretivas ngIf e ngFor
  ],
  exports: [
    Comp2Component // Exporta o componente para que outros módulos possam usá-lo
  ]
})
export class ModuloExterno {}
```

2. Importar o Módulo no Módulo Desejado

Agora, no módulo onde você quer utilizar o componente `Comp2Component`, importe o `ModuloExterno`.

Arquivo: `app.module.ts`

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

```
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component'; // Componente principal
import { ModuloExterno } from './modulo-externo/modulo-externo.module'; // Importa o
módulo externo

@NgModule({
  declarations: [
    AppComponent // Declara o componente principal
  ],
  imports: [
    BrowserModule, // Funcionalidades do navegador
    ModuloExterno // Importa o módulo que contém o componente que queremos usar
  ],
  providers: [],
  bootstrap: [AppComponent] // Define o componente principal para inicializar a aplicação
})
export class AppModule {}
```

3. Usar o Componente Exportado no Template

Depois de importar o módulo que contém o componente desejado, você pode usar o **selector** do componente diretamente no template de qualquer componente deste módulo.

Arquivo: **app.component.html**

```
<!-- Utiliza o componente exportado de ModuloExterno -->
```

```
<app-comp2></app-comp2>
```

Problemas Comuns

1. Erro: 'app-comp2' is not a known element

- Isso acontece se:
 - O módulo onde o componente está declarado não foi importado no módulo atual.
 - O componente não foi exportado no módulo onde foi declarado.
- Solução: Verifique se o componente está corretamente exportado no módulo de origem e se o módulo de origem foi importado.

2. Exportação Desnecessária de Componentes

- Não exporte componentes que não precisam ser utilizados fora de seu módulo.

Exemplo Completo

Módulo Externo

Arquivo: **modulo-externo.module.ts**

```
import { NgModule } from '@angular/core';
import { CommonModule } from '@angular/common';
import { Comp2Component } from '../comp2/comp2.component';

@NgModule({
  declarations: [Comp2Component],
  imports: [CommonModule],
  exports: [Comp2Component] // Torna o componente acessível para outros módulos
})
export class ModuloExterno {}
```

Arquivo: **comp2.component.ts**

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

@Component({
  selector: 'app-comp2', // Nome do seletor que será usado no HTML
  template: `<p>Componente de Módulo Externo!</p>`,
  styleUrls: ['./comp2.component.scss']
})
export class Comp2Component {}
```

Módulo Principal

Arquivo: **app.module.ts**

```
import { NgModule } from '@angular/core';
import { BrowserModule } from '@angular/platform-browser';
import { AppComponent } from './app.component';
import { ModuloExterno } from '../modulo-externo/modulo-externo.module';

@NgModule({
  declarations: [AppComponent],
  imports: [
```

```
    BrowserModule,  
    ModuloExterno // Importa o módulo externo  
  ],  
  providers: [],  
  bootstrap: [AppComponent]  
})  
export class AppModule {}
```

Componente Principal

Arquivo: **app.component.html**

```
<h1>Componente Principal</h1>  
<app-comp2></app-comp2> <!-- Usa o componente do módulo externo -->
```

Resumo

- **Declarar** componentes no módulo onde pertencem.
- **Exportar** componentes no módulo de origem para permitir seu uso em outros módulos.
- **Importar** o módulo de origem no módulo onde o componente será utilizado.