

[(ngModel)] – diretiva Angular para vinculação bidirecional

[(ngModel)] é uma diretiva em Angular que fornece vinculação bidirecional de dados. Ela é comumente usada em formulários para vincular os valores dos elementos de formulário (como inputs) a propriedades em um componente TypeScript. A vinculação bidirecional significa que as alterações no modelo afetarão a visualização e vice-versa.

A diretiva [(ngModel)] é parte do módulo FormsModule em Angular, portanto, para usá-la, você precisa importar o módulo em seu componente. Aqui está um exemplo básico de como usar [(ngModel)]:

1. **Importe o FormsModule no seu módulo:**

```
// Importe o módulo FormsModule
import { FormsModule } from '@angular/forms';

// ...

@NgModule({
  imports: [
    // Adicione FormsModule aos imports
    FormsModule,
    // ...
  ],
  // ...
})
export class SeuModulo { }
...

```

2. **Use [(ngModel)] em um componente:**

Considere o seguinte componente TypeScript:

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

@Component({
  selector: 'seu-componente',
  template: `
    <!-- Use [(ngModel)] para vincular a propriedade 'nome' -->
    <input [(ngModel)]="nome" placeholder="Seu Nome">
    <p>Olá, {{ nome }}!</p>
  `,
})
export class SeuComponente {
  // A propriedade 'nome' é vinculada ao input usando [(ngModel)]
  nome: string = "";
}
...

```

Neste exemplo, a propriedade `nome` do componente é vinculada ao valor de um input usando `[(ngModel)]`. Qualquer alteração no input será refletida na propriedade `nome` do componente e vice-versa.

É importante observar que `[(ngModel)]` só funciona quando você está usando o formulário dentro de um formulário Angular (<form>), e o formulário precisa ser associado a uma instância do `ngForm`. Além disso, para usar `[(ngModel)]`, certifique-se de ter importado o `FormsModule` em seu módulo, como mencionado anteriormente.

No exemplo fornecido, `{{ nome }}` é uma interpolação em Angular. A interpolação é uma forma de incorporar valores de propriedades do componente diretamente no modelo HTML. No caso específico de `{{ nome }}`, ele exibirá o valor atual da propriedade `nome` do componente no local onde a interpolação está sendo usada.

Aqui está um trecho do exemplo fornecido para contextualizar:

```
``typescript
@Component({
  selector: 'seu-componente',
  template: `
    <!-- Use [(ngModel)] para vincular a propriedade 'nome' -->
    <input [(ngModel)]="nome" placeholder="Seu Nome">
    <p>Olá, {{ nome }}!</p>
  `,
})
export class SeuComponente {
  // A propriedade 'nome' é vinculada ao input usando [(ngModel)]
  nome: string = '';
}
```

Neste exemplo, o conteúdo do parágrafo `<p>` contém a interpolação `{{ nome }}`. Isso significa que o valor atual da propriedade `nome` do componente será exibido no lugar da interpolação quando a página for renderizada.

Quando o usuário digitar algo no campo de input, a diretiva `[(ngModel)]` atualiza a propriedade `nome` do componente, e como há uma vinculação bidirecional, qualquer alteração na propriedade `nome` também será refletida na exibição usando `{{ nome }}`.

Então, se o usuário digitar "João" no campo de input, o parágrafo exibirá "Olá, João!". Isso ilustra a interação entre as diretivas de vinculação bidirecional e as interpolações em Angular para manter o modelo e a visualização sincronizados.