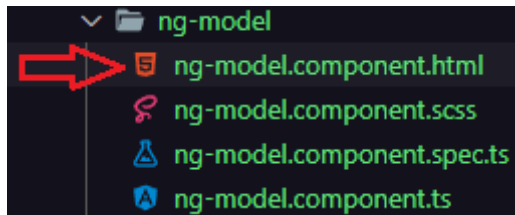


ngModel

A diretiva "ngModel" é usada para vincular os campos que serão validados ao formulário, ele é referenciado junto a propriedade "name" da tag que foi atribuído

Para utilizar:

1) Para utilizar as funcionalidades do ngModel devemos atribui-lo as propriedades de uma tag que tenha uma propriedade "name" atribuída. Desta forma, a tag será vinculada aos controles do formulário.



```
1 <div class="div-componente">
2   <h2 class="title-componente">ngModel</h2>
3   <div class="componente-container">
4     <p class="descricao-componente">
5       A diretiva "ngModel" é usada para vincular os campos que serão validados ao formulário, ele
6       que foi atribuído. Neste exemplo, a tag "input" está recebendo o "name" de "textoNome" e a d
7       for submetido, seja impresso no console as informações dos campos que estão sendo controlado
8     </p>
9   </div>
10  <div class="componente-exemplo">
11    <form class="formulario-exemplo" #formDir="ngForm" (ngSubmit)="enviarFormulario(formDir)">
12      <div class="intens-formulario">
13        <input
14          type="text"
15          placeholder="Digite um nome"
16          class="input-texto"
17          name="textoNome"
18          ngModel
19        >
20        <button class="botao-enviar">Enviar</button>
21      </div>
22    </form>
23  </div>
24  <div class="componente-buttons-container">
25    <app-arquivo-pdf [endereco]="endereco"></app-arquivo-pdf>
26    <app-repositorio [github]="github"></app-repositorio>
27  </div>
28 </div>
```

2) Com a tag atribuída aos controles do formulário com a diretiva “ngModel”, podemos configurar o método “enviarFormulario” para exibir no console os valores dos controles do formulário. Esta configuração é realizada no arquivo “.ts” do componente.



```
src > app > modules > formulario > components > ng-model > ng-model.component.ts > ...
1  import { Component } from '@angular/core';
2  import { NgForm } from '@angular/forms';
3
4  @Component({
5    selector: 'app-ng-model',
6    templateUrl: './ng-model.component.html',
7    styleUrls: ['./ng-model.component.scss']
8  })
9  export class NgModelComponent {
10
11    endereco: string = '../../../assets/ngForm.pdf'
12    github: string = 'https://github.com/LuanSOliveira/Testes-Angular/t
13
14    enviarFormulario(form: NgForm):void{
15      console.log(form.controls);
16    }
17  }
18
```

The code snippet shows the configuration of the 'enviarFormulario' method in the 'NgModelComponent' class. The method is highlighted with a red box, and a red arrow points to it from the right.

3) Desta forma, quando o formulário for submetido, serão apresentados no console as informações dos controles do formulário.

ngModel

A diretiva "ngModel" é usada para vincular os campos que serão validados ao formulário, ele é referenciado junto a propriedade "name" da tag que foi atribuído. Neste exemplo, a tag "input" está recebendo o "name" de "textoNome" e a diretiva "ngModel" para que quando o formulário for submetido, seja impresso no console as informações dos campos que estão sendo controlados pelo "formDir". Ou seja, os campos com o "ngModel".

Enviar

Como Utilizar

Repositório

Required

Elements

Console

Sources

Network

Performance

3

Default levels

3 Issues

▼ {textNome: FormControl}

▼ textNome: FormControl

defaultValue: null

errors: null

pristine: false

status: "VALID"

▶ statusChanges: EventEmitter_ {closed: false, currentObservers: Array(0), observed: true}

▶ valueChanges: EventEmitter_ {closed: false, currentObservers: Array(0), observed: true}

value: "VALORES DOS CONTROLES"

._composedAsyncValidatorFn: null

._composedValidatorFn: null

._hasOwnPendingAsyncValidator: false

._onChange: [f]

._onCollectionChange: () => {}

._onDisabledChange: [f]

▶ _parent: FormGroup { _pendingDirty: false, _hasOwnPendingAsyncValidator: false }

._pendingChange: false

._pendingDirty: true

._pendingTouched: true

._pendingValue: "VALORES DOS CONTROLES"

._rawAsyncValidators: null

._rawValidators: null

asyncValidator: (...)

dirty: (...)

disabled: (...)

enabled: (...)

invalid: (...)

parent: (...)

pending: (...)

root: (...)

untouched: (...)

updateOn: (...)

valid: (...)

validator: (...)

▶ [[Prototype]]: AbstractControl

▶ [[Prototype]]: Object