

Aula 5

Descrição

Recapitular o que vimos

Angular

O que é ?



Angular é uma plataforma de aplicações web de código-fonte aberto e front-end baseado em TypeScript liderado pela Equipe Angular do Google e por uma comunidade de indivíduos e corporações. Angular é uma reescrita completa do AngularJS, feito pela mesma equipe que o construiu

Angular - Ambiente



- Instalando o Node.JS
- Angular CLI <https://angular.io/guide/setup-local>
 - npm install -g @angular/cli
 - Set-ExecutionPolicy -Scope CurrentUser -ExecutionPolicy RemoteSigned
- Instalar um editor de texto (... VS-CODE)
- Instalar um browser

Angular - Primeiro Projeto



- `ng new todo --routing false --style css --skip-git --skip-tests`
- Rodar aplicação template (`ng serve`)
- Criar uma pasta models, adicionar a classe TodoItem
 - descrição, status
- Criar uma classe TodoList
 - lista de todoItems, dono da lista
- Mostrar os dados do todo
 - Apagar o conteúdo do `app.component.html`
 - Editar o `html` para trazer dados do `todoList`

Angular - Primeiro Projeto



- `ng new todo --routing false --style css --skip-git --skip-tests`
- Rodar aplicação template (`ng serve`)
- Verificar o que criou
- Verificar os arquivos gerados pela scaffolding
- Dúvidas?

Angular - Primeiro Projeto



- Criar uma pasta models, adicionar a classe TodoItem
 - descrição, status

```
export class Tarefa{  
    constructor(nomeTarefaVal: string, completaVal: boolean =  
false) {  
        this.nomeTarefa = nomeTarefaVal;  
        this.completo = completaVal;  
    }  
    nomeTarefa: string;  
    completo: boolean;  
}
```

Angular - Primeiro Projeto



- Criar uma classe TodoList
 - lista de todoItems, dono da lista

```
export class TarefasLista {  
    constructor(public usuario: string, private listaTarefas: Tarefa[] =  
[]) {  
    }  
  
    get tarefas(): readonly Tarefa[] {  
        return this.listaTarefas;  
    }  
  
    addTarefa(nomeTarefa: string) {  
        this.listaTarefas.push(new Tarefa(nomeTarefa));  
    }  
}
```

Angular - Primeiro Projeto



- Mostrar os dados do todo
 - Apagar o conteúdo do app.component.html
 - Editar o html para trazer dados do listadeTarefa

```
<h3 class="bg-primary text-center text-white p-2">
```

```
  {{ username }} Lista de tarefas
```

```
  <h6 class="mt-1">{{ itemCount }} Incompletos</h6>
```

```
</h3>
```


Angular - Primeiro Projeto



- Atualizando o componente
 - Alterar o app.component.ts para ter os dados do todoList

```
export class AppComponent {  
  private list = new TodoList("Senac", [  
    new TodoItem("Estudar HTML"),  
    new TodoItem("Estudar CSS"),  
    new TodoItem("Estudar JAVASCRIPT"),  
  ]);  
  get username(): string {  
    return this.list.user;  
  }  
  get itemCount(): number {  
    return this.list.items.filter(item => !item.complete).length;  
  }  
}
```

Angular - Primeiro Projeto



- Entendo os imports
- Entendo o decorator
- Entendo a classe

Angular - Primeiro Projeto



- Estilizando os elementos
 - npm install bootstrap@4.4.1
 - mostrar os itens estilizados
 - DataBind, dono da lista e total de itens
- FormsModule

Angular - Primeiro Projeto



- Estilizando os elementos

```
<table class="table table-striped table-bordered table-sm">
  <thead>
    <tr>
      <th>#</th>
      <th>Nome da tarefa</th>
      <th>Completa</th>
    </tr>
  </thead>
  <tbody>
    <tr *ngFor="let item of lista.tarefas; let i = index">
      <td>{{ i + 1 }}</td>
      <td>{{ item.nomeTarefa }}</td>
      <td [ngSwitch]="item.completo">
        <span *ngSwitchCase="true">Sim</span>
        <span *ngSwitchDefault>Nao</span>
      </td>
    </tr>
  </tbody>
</table>
```

Angular - Primeiro Projeto



- Adicionar item

```
<div class="container-fluid">  
  <div class="row">  
    <div class="col">  
      <input class="form-control" placeholder="Enter task here" #todoText />  
    </div>  
    <div class="col-auto">  
      <button class="btn btn-primary" (click)="addTarefa(todoText.value)">  
        Adicionar  
      </button>  
    </div>  
  </div>  
</div>
```

Angular - Primeiro Projeto



- Filtrar lista
 - Adicionar propriedade no app.component.ts
 - `showComplete: boolean = false;`

```
<div class="bg-secondary text-white text-center p-2">  
  <div class="form-check">  
    <input class="form-check-input" type="checkbox" [(ngModel)]="showComplete" />  
    <label class="form-check-label" for="defaultCheck1">  
      Completas  
    </label>  
  </div>  
</div>
```

Angular - Primeiro Projeto



- Alterar apresentação

```
<h3 class="bg-primary text-center text-white p-2">
  {{ username }} minhas tarefas
  <h6 class="mt-1">{{ itemCount }} {{ showComplete ? "" : "Incompleto" }} tarefas</h6>
</h3>
<tbody>
  <tr *ngFor="let item of lista.tarefas; let i = index">
    <td>{{ i + 1 }}</td>
    <td>{{ item.nomeTarefa }}</td>
    <td>
      <label for="c{{ i }}"></label>
      <input type="checkbox" id="c{{ i }}" [(ngModel)]="item.completo" />
    </td>
  </tr>
</tbody>
```

Angular - Primeiro Projeto



- Alterar apresentação

```
addTarefa(nomeTarefa: string) {  
    this.lista.addTarefa(nomeTarefa);  
}
```

Dica extra, alterar o arquivo angular.json para aparecer o bootstrap

```
"styles": [  
    "src/styles.css",  
    "node_modules/bootstrap/dist/css/bootstrap.min.css"  
],
```


Angular - Exercício



1. Agora que temos um esboço do TODO, adicionar novos dados no nosso objeto Tarefa:
 - Adicione detalhe
 - Adicione data
2. Agora, gere um novo aplicativo e crie um calculado como fizemos como o TODO.