

Atividade Angular 02

1 - Crie os seguintes componentes:

- linha-do-tempo
- post-da-linha

O componente linha-do-tempo é filho de app.component. O componente post-da-linha é filho de linha-do-tempo.

2 - No componente linha-do-tempo, você deverá implementar a seguinte interface (Bootstrap):

Nome

Os seguinte elementos irão ficar dentro de uma `<div class="form-group">`:

- O label “Nome” e o Input ao lado deverão ficar em uma mesma `<div class="row">` do bootstrap, ou seja, o nome ficará em uma `<div class="col-1">` e o Input em `<div class="col-11">`, totalizando assim 12 colunas.
- Logo abaixo, para o `<textarea>`, você também irá criar uma outra linha (row) mas com 12 colunas.
- Já o Buton “enviar”, não precisa estar em uma row, mas ainda dentro da `<div class="form-group">`.

3 - O que deve ser implementado nesse componente linha-do-tempo?

O componente linha-do-tempo deverá armazenar um vetor de objetos JSON com o seguinte formato:

```
{id: <valor do id>, nome: <valor do nome>, texto: <valor do texto>, likes:0}
```

Toda vez que o botão “enviar” for clicado, chame um método em linha-do-tempo, passando como parâmetros o nome e o texto. Nesse mesmo método, você irá fazer um “push” do objeto JSON criado para dentro do vetor. Implemente o id como um contador, ou seja, para cada post, o id incrementa em 1.

4 - O que deve ser implementado nesse componente post-da-linha?

O componente post-da-linha deverá criar uma variável do tipo `@Input`, a qual irá receber um objeto do mesmo tipo que o criado pelo componente pai, linha-do-tempo (descrito em 3). Esse objeto será passado via seletor do componente post-da-linha.

5 - Mas como os posts criado em linha-do-tempo serão passados para o componente post-da-linha?

Ainda no HTML de linha-do-tempo (logo abaixo do `<div class="form-group">`), você deverá fazer um `*ngFor`, chamando o seletor de post-da-linha e passando assim o parâmetro post para o componente-filho. Veja:

```
...
<div class="row" *ngFor="let p of posts">
  <div class="col-12">
    <app-post-da-linha [postIn]="p"></app-post-da-linha>
  </div>
</div>
```

6 - O que deverá ser implementado na interface de posts-da-linha?

Em posts-da-linha.html, você deverá mostra todo o conteúdo de um post, menos o id. Veja:

Nome: Jefferson
Texto: Eita que eu acho que vai chover...
Likes: 0
LIKE

Esse conteúdo deverá ficar dentro de um `<div class="alert alert-primary">`, usando interpolação para mostrar os dados.

No fim, a interface final ficaria assim:

Nome

Enviar

Nome: Jefferson
Texto: Eita que eu acho que vai chover...
Likes: 1
LIKE

Nome: Thiago
Texto: Será mesmo? Não acredito...
Likes: 2
LIKE

7 - Ok, mas e o botão LIKE em post-da-linha.html?

Ahá! Através desse botão, você deverá enviar ao componente PAI, linha-do-tempo, o id do post que foi dado o like. Lá em linha-do-tempo, você deverá “procurar” dentro do vetor de posts qual o post tem o id passado como parâmetro. Uma vez encontrado, basta incrementar os likes. Olha a moleza:

```
findPost(id:number):any{
    for(let post of this.posts){
        if(post.id==id) return post;
    }
    return null;
}
```

8 - Mas como passar o id de um post que foi dado o like em post-da-linha para o componente pai?

Ainda dentro de post-da-linha você deverá criar uma variável evento com `@Output`, o qual irá “emitir” o id do post selecionado. Ou seja, quando você clicar em LIKE, você irá chamar um método dentro de post-da-linha.ts o qual receberá o id do post. Esse método irá então disparar o evento do `@Output` passando como parâmetro, o id do post.

9 - E como o linha-do-tempo irá capturar esse evento?

Ataves do seletor de post-da-linha, como explicado antes em 5. Veja nos exercícios anteriores em sala de aula, como um componente pai captura um evento do filho.

O trabalho DEVERÁ ser entregue em aula, ou a equipe (dois alunos) não levará presença. PARA CADA FUNCIONALIDADE FALTANTE, MENOS 1 PONTO. Todo trabalho começa com DEZ.Envia apenas a pasta APP.