

Exercício Angular - Consumindo Api Rest, Formulário e Rotas - Prof. Ramon Pereira - UniBH 2020/1

Nesta atividade iremos praticar os conceitos de consumir dados de uma API Rest, criar formulários, enviar formulários e navegar entre os componentes utilizando as Rotas.

O objetivo será criar uma aplicação para exibir os dados da Fake API <https://jsonplaceholder.typicode.com/>.

- 1) Criar um projeto do Angular (ng new) **IMPORTANTE:** dizer sim para a criação de do módulo de rotas rotas (primeiro pedido que é feito).
- 2) Criar os seguintes componentes: (ng generate component NOMEDOCOMPONENTE)
 - a) Nav
 - b) User
 - c) Post
 - d) Comment
- 3) Crie um arquivo de interface para cada um dos componentes User, Posts, Comments.
 - a) Crie uma pasta models e salve todos os modelos dentro dela.
 - b) Conferir no site as propriedades que devem existir em cada um dos componentes. (<https://jsonplaceholder.typicode.com/>)
 - c) Na interface do Post, além de todas as propriedades citadas adicione uma booleana chamada `exibirComentario` inicializada como `false` por padrão.
- 4) Crie uma pasta chamada `services` para os serviços. Crie um serviço para cada um dos componentes, exceto para a barra de navegação. (ng generate service services/NOMEDOSERVICO)
- 5) Instale o Bootstrap (npm install bootstrap)
- 6) Importe o Bootstrap no arquivo `Angular.json`

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

**** Adicionar a linha destacada acima**
- 7) Nos serviços User, Post e Comment:
 - a) crie os métodos `Get`, `GetByld` e `Post` utilizando `Async Await`. Os métodos `GetByld` deverão receber um `id` como parâmetro e os métodos `posts` deverão receber um objeto do tipo específico de cada serviço como parâmetro.

- b) Os serviços deverão ser Injectable e providos forRoot. Este já é o comportamento padrão quando cria pelo angular CLI, só conferir.
 - c) Para realizar as requisições HTTP você deverá importar o HttpClient do '@angular/common/http' em cada um dos serviços.
- 8) No app module:
- a) Para utilizar o Http, você deverá importar o HttpClientModule também do '@angular/common/http'. Este módulo deverá estar inserido nos imports e nos providers.
 - b) Para utilizar o formulário, você também deverá importar o FormsModule import { FormsModule } from '@angular/forms' e adicionar ele nos Imports.
- 9) No Typescript de cada um dos componentes:
- a) importe sua respectiva Interface
 - b) importe seu respectivo serviço (se não tiver importado na questão anterior)
 - c) crie uma propriedade do tipo da interface
 - d) crie uma propriedade que será um array do tipo da interface
 - e) instancie o serviço através de injeção de dependência no construtor da classe do componente
 - f) execute a chamada da função get dentro do ngOnInit (não esqueça de colocar ele como assíncrono)
- 10) No typescript do Post Component
- a) Crie a função onSubmit do formulário
 - b) A função onSubmit deverá receber os dados de submissão do formulário e ativar o serviço de inserção de dados do PostServices
 - c) Instancie a propriedade Post como vazia para que ela possa ser usada com o ngModel (post: Post = {} as Post;)
- 11) No typescript do Comment Component além do já citado
- a) Deverá ter um @Input para receber o id do Post
 - b) Deverá carregar os comentários desse post através do serviço de getComentarios, adicionando o parâmetro de filtragem por id do post na rota
- 12) No typescript do Routing Module
- a) Adicionar na constante routes as rotas para cada um dos componentes no padrão path, componente.
ex: { path: 'home', component: HomeComponent }
- 13) No template do Nav Component:
- a) Adicione um nav contendo uma ul e um li para cada um dos itens: Usuários, Posts, Comentários
 - b) Estilize esses elementos para serem exibidos em linha (display: inline)
 - c) Adicione o link de routerlink para cada um deles a referida rota criada no componente (exercício 12). Dica:

14) No template do User component:

- a) Crie uma tabela para listar os dados do usuário que serão retornados do serviço, deverá ser exibido um usuário por linha. (Utilize ngFor)

15) No template dos Comentários:

- a) Crie um cartão para exibir os comentários de um determinado Post. Pode ser um bloco de divisão.

16) No template dos Posts

- a) Crie um formulário para inserir um novo Post;
- b) Declare o nome do formulário como ngForm
- c) Adicione o evento (ngSubmit) chamando a função de onSubmit passando o formulário como parâmetro
- d) Adicione os inputs: Título (name: title), comentário (name: body) e um select para selecionar um usuário (name: userId) este select deverá ser preenchido dinamicamente utilizando o serviço de getUsers (Neste momento Instanciar o serviço de getUsers com injeção de dependência no typescript)
- e) Crie uma divisão com <hr>
- f) Crie divisões para exibir todos os dados dos posts que serão retornados pelo serviço de Posts
- g) No final de cada divisão deverá ter a chamada do componente Comentários passando como input o id do post e com um ngIf para verificar o estado do booleano exibirComentario deste post;
- h) Adicione um botão para Exibir Comentários que ative uma função exibirComentarios que recebe um id de post como parâmetro.
- i) A função exibirComentarios deverá gerenciar o estado da propriedade exibirComentario do respectivo post alterando para true para que os comentários sejam exibidos na tela.

17) No template do app.componente:

- a) Remova todo conteúdo
- b) Inclua o componente barra de Navegação (NavComponent)
- c) Inclua o componente <router-outlet></router-outlet>