

Angular – Projetos GitHub


- ▶ Blog – Meu Blog, um simples e bonito projeto em Angular Java para você fazer engenharia reversa e aprender

Acesse: <https://github.com/swilliamsilva/Blog>

Exercício: Baixe esse código

teste o programa e siga o fluxo da informação descrito nessa apresentação.

Como exercício crie o seu.



Angular – FrontEnd

- ▶ Para essa aula você precisa conhecer o GIT
- ▶ NodeJS
- ▶ TypeScript
- ▶ HTML
- ▶ CSS3

Angular

- Angular (comumente referido como "Angular 8+" ou "Angular 8") 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.



A melhor fonte de consulta é a documentação oficial e atual.

Documentação do Angular: <https://angular.io/>

Angular TypeScript

- TypeScript é uma linguagem de programação de código aberto desenvolvida pela Microsoft. É um superconjunto sintático estrito de JavaScript e adiciona tipagem estática opcional à linguagem. Tipos fornecem uma maneira de descrever a forma de um objeto, fornecendo melhor documentação e permitindo que o TypeScript valide se seu código está funcionando corretamente. Como TypeScript é um superconjunto de JavaScript, os programas JavaScript existentes também são programas TypeScript válidos.



Consulte sempre a documentação oficial que é a melhor fonte.

Documentação oficial do TypeScript: <https://www.typescriptlang.org/docs/>

Angular – Projetos GitHub

➤ Arquitetura:

Angular.json → Expõe variáveis globais, o arquivo responsável por inicializar o SPA (Simple Page Application)

“main”:src/main.ls”

o arquivo responsável por uma única página é o INDEX.HTML.

No arquivo main.ts onde faz o BootStrap , onde vai encontrar os componentes de forma direta e indireta.

Para carregar a aplicação:

Ng serve -o

O angular usa a porta padrão:

LocalHost: 4200 -> No navegador.

App.module.ts

Ele traz vida ao componente

O módulo é um limitador de contexto

Angular – Estudo do Projeto Blog

- Blog – Meu Blog, um simples e bonito projeto em Angular Java para engenharia reversa.

Acesse: <https://github.com/swilliamsilva/Blog>

Angular – Estudo do Projeto Blog

➤ INICIO DO PROJETO

Projeto: Vamos construir o **frontend** do Blog no Java / Angular

Definições:

FrontEnd é uma SPA ANGULAR = Single Page Application.

O Angular trabalha com o Componentes

Veja a documentação do Angular CLI em (<https://cli.angular.io/>)

CLI – Comand Line Interface = Interface de Linha de Comando

Através do terminal.

Angular – Estudo do Projeto Blog

- Para instalar o Angular – Comandos

Siga essa sequencia de comandos no Terminal

- **`npm i -g @angular/CLI`**
- **`ng new blog`**
- **`cd blog`**
- **`ng serve`**

*** npm = Node Package Manager*

Angular – Estudo do Projeto Blog

- Para instalar o Angular – Comandos

Siga essa sequencia de comandos no Terminal

- **`npm i -g @angular/CLI`**
- **`ng new blog`**
- **`cd blog`**
- **`ng serve`**

*** npm = Node Package Manager*

Angular – Estudo do Projeto Blog

`ng i -g @angular/cli`

Quando usamos a opção `-g` significa que todos os projetos do angular já vai utilizar a versão do angular e não preciso ficar instalando por projeto.

Se não colocar o `-g` ele instala apenas local.

A vantagem é que podemos ter vários projetos angular com versões diferentes do angular.

É mais fácil fazer gestão de versão.

Se fizer uma versão global é uma mesma versão utilizada para todos os projetos.

Angular – Estudo do Projeto Blog

➤ **Ng new Blog**

Com esse comando eu vou criar uma novo Blog

O Angular constroi a aplicação

Cd Blog

Através do terminal você entra na pasta do projeto que ele vai criar.

Ng serve

É para subir o servidor com a aplicação.

Angular – Estudo do Projeto Blog

➤ Estrutura de Execução

O arquivo **main.ts** aciona o arquivo **app.module.ts** e dentro dele tem a chamada para o primeiro módulo da aplicação que é o **AppComponent**.

Toda vez que criar um componente vai ter que declarar ele dentro do **app.module.ts** em **declarations**

.....

```
@ng module ({  
  declarations: [  
    AppComponent,  
    HomeComponent,  
    ...  
  ],  
  .
```

Angular – Estudo do Projeto Blog

- O **App.Component.ts** ele importa o Component do Angular e chama esse primeiro componente de **'app-root'**.
- ...
- **@ Component ({**
Seletor: 'app-root',
- que esta associado a mais duas partes do componente **./app.Component.html** e o **/app.component.css**

Angular – Estudo do Projeto Blog

➤ *Index.html*

Nele cria uma tag `<app-root>` que é uma chamada para o Selector que no arquivo `app.component.ts`

O `app.root` é o componente principal que aciona o arquivo `app.component.ts` o arquivo `HTML` e o arquivo `CSS`.

Na criação o angular coloca o `CSS` dentro do `HTML`

`ng serve -o`

Vai abrir a nossa aplicação e já abre o Browser

Angular – Estudo do Projeto Blog

- Iniciando a aplicação e criando os componentes

Fazendo isso ele vai criar os arquivos auxiliares que são o **.html**

.css

.ts

.spec

Ele também cria dentro da **pasta App** uma pasta chamada **Home**.

Se você abrir essa pasta irá ver todos os arquivos lá dentro.

Nos arquivos **Home.componente.ts** ele já monta a estrutura básica

Selector

Templateurl

styleurl

Angular – Estudo do Projeto Blog

- *Iniciando a aplicação e criando os componentes*

... e já vem com o método inicial que é o `ngOnInit(): void`

Então se cria todas os demais componentes que são as páginas do mesmo projeto:

ng g c Nav-Bar

ng g c footer

ng g c feed

ng g c contato

Angular – Estudo do Projeto Blog

O arquivo **app-routing-module.ts** que é o arquivo de rotas ele foi criado junto com com a aplicação e nele fica as rotas da aplicação.

No **path** eu coloco o caminho da minha rota.

Quando a execução abrir a aplicação a primeira linha que vai executar é o caminho da aplicação.

Quando você digitar LocalHost:4200 que é o caminho de execução da aplicação ele vai executar a linha.

Path: “, redirect to:” **Home, pathMatch: ‘full’},**

Para a página **Home** da aplicação.

Angular – Estudo do Projeto Blog

Essa linha é a raiz da aplicação o comando: **pathMatch:'full'** ele vai fazer um redirecionamento total para esse lugar.

Ele entende que as ' ' aspas faz que a aplicação vá para a raiz do projeto.

No segundo posso eu chamo os demais componentes com o path.

```
{path:'ome', component:HomeComponent},  
{path:'feed', Component:FeedComponent},  
{path:'contato',Component:ContatoComponent},
```

ContatoComponent → Esse é o nome do component

Angular – Estudo do Projeto Blog

Vamos instalar o BootStrap

npm i bootstrap

Digitando este comando no terminal na pasta do seu projeto ele vai fazer a conexão e baixar todos os pacotes que é necessário do BootStrap

Ele também precisa de algumas outras recursos para funcionar, um deles é POPPER e o outro é o JSQUERY.

npm i jquery@~3.4.1

npm i popper.js

Angular – Estudo do Projeto Blog

Vamos instalar também o conjunto de ICONES e FONTES que tem os ícones das redes sociais, de menu e das grandes empresas.

ng add @fortawesome/angular_fontawsome

Apesar de ser uma instalação **ng** ele faz uma conexão com o **npm**.

Demora um pouco a instalação.

Tem que selecionar as opções FREE.

Angular – Estudo do Projeto Blog

Tem que selecionar as três opções FREE.

Selecione e pressione ENTER

Esse pacote oferece os ícones de diversas marcas:

Solid icons

Regular Icons

Brands Icons sendo esse para marcas de empresas

Isto é para colocar ícones no footer do seu projeto e acessar as redes sociais.

Não é possível alterar essas imagens mas é possível configurar essas fontes para ser reconhecido dentro do projeto do angular.

Isso se faz no arquivo **Angular.Json** em **Styles** e **Scripts**

Angular – Estudo do Projeto Blog

No arquivo **Angular.Json** vamos fazer a **configuração do BootStraps**.

O BootStrap quando é instalado ele fica dentro do **node.modules.js** em
“./node_modules/bootstrap/dist/css/bootstrap.min.css”

Esse é o comando pra o CSS do BootStrap

No Script você vai mencionar os arquivos JavaScript que o BootStrap vai utilizar o JSQUERY e o POPPER.

Angular – Estudo do Projeto Blog

Agora vamos fazer a configuração do **FONT AWSOME = FONTES MARAVILHOSAS**

Esta configuração é feita dentro do arquivo **app.module.ts**

Dentro desse arquivo tem a listagem dos componentes.

Em **declarations**

Quando formos adicionar os componentes ele foi adicionando automaticamente os componentes.

Além desses componentes que criamos ele precisa adicionar, “importar” outros componentes então se você objetivar quando você faz a instalação na pasta ele já adiciona esses componentes em:

Imports: [

Browser Module,

AppRouting Module,

FontAwe Some Module,

formsModule.

Angular – Estudo do Projeto Blog

Construindo a navbar

Abrir o site do Bootstrap: **Getbootstrap.com**

Vai em **Documentation/Components/NavBar**

Tem vários exemplos de navbar, escolha um com a seta.

Então vamos no primeiro clicando em copiar.

Volte para o projeto em:

navbar.componentes.html e coloco lá dentro dentro do Body.

Conseguimos assim construir com apenas alguns comandos várias páginas usando o esquema de roteamento do angular.

Angular – Estudo do Projeto Blog

Agora vamos construir o FOOTER

No **footer.componentes.ts** é preciso fazer os imports:

```
Import {fa.Instagram}from '@fortawesome/free-brands-svg-icons';
```

..

Fazer as importações no construtor;

```
faInstagram = fainstagram
```

```
faFacebook = faFacebook
```

No arquivo **footer.componente.html** dentro da **<div>** e dentro da **tag <a>**.

Não criamos uma **tag** que não existe no HTML mas que é uma **tag do angular**.

Criado pelo próprio fonte awe some ‘

```
`<fa_icon[icon] = "facebook"></fa_icon>
```

Angular – Estudo do Projeto Blog

Agora podemos criar Classes e fazer a formatação no CSS

Criamos as classes **icoFacebook**

icoInstagram

icoLinkedin e podemos ver as formatações dos arquivos:

Footer.componente.css

Por exemplo:

```
.icoFacebook: hover{  
    color: #3b5998;  
}
```

Angular – Estudo do Projeto Blog

▶ *Trabalhando o componente HOME*

A **div** `<div class="Parallax"></div>`

Produz aquele efeito de uma página passando para trazer.

`<h1 class = 'Test_center">` - Já é uma classe do Bootstrap

Então essa imagem Parallax nos colocamos e configurarmos dentro do
Home.components.css

Angular – Estudo do Projeto Blog

- **Pode pegar as imagens no pixabay.com imagens gratuitas.**

Para pegar a imagem clic com o mouse direito na imagem e copie o link da imagem e copie o código.

Angular – Estudo do Projeto Blog

Trabalhando o componente Contato

Na página de contato é criar um formulário.

Vamos no site do Bootstrap lá em

DOCUMENTATION / COMPONENTS / FORMS

Escolha o formulário, copie e cole no **contato.component.html**

Angular – Estudo do Projeto Blog

➤ *Vamos fazer o feed*

Getbootstrap.com

Documentation / Component / Card

Angular – Estudo do Projeto Blog

➤ Comunicação com o Back End

Vamos construir uma **model** que vai fazer a representação do nosso banco de dados.

Model é o servisse que vai fazer essa comunicação com o Backend

1) Primeiro vamos construir dentro da pasta APP um novo arquivo chamado MODEL **APP / model**

Ela vai representar o meu modelo do meu banco de dados.

Dentro da pasta model vamos criar um arquivo tipo TS com o nome do nosso banco de dados

Vou chamar esse arquivo de **Post.ts**

Angular – Estudo do Projeto Blog

- O model é uma Classe que vamos exportar e a partir disso é que vamos fazer a nossa comunicação com o Backend.

Export class Post

Post é o mesmo nome do arquivo que acabei de criar e aqui dentro vamos colocar os campos do meu banco.

Olhando no Post dentro do BackEnd vamos observar que temos os campos

Nome

Mensagem

Id

Angular – Estudo do Projeto Blog

- Então em Post na pasta model vamos colocar:

```
Export class Post {  
    public id number  
    public nome: string  
    public mensagem: String  
}
```

Somente com isso já construímos a model que é o suficiente para poder fazer o servisse.

Então para fazer essa comunicação de fato com o BackEnd vamos construir agora o mesmo servisse.

Angular – Estudo do Projeto Blog

➤ Então vamos usar o recurso do próprio angular para construir o meu service.

ng g s service/Post → Para gerar um Service/Post

Ele vai gerar dois arquivos que são de fato os Services

Já podemos observar que geramos dois arquivos na pasta Service

Post.service.ts

Post.servic.spec.ts

Nos sempre trabalhamos nesse arquivo Post.service.ts

E aqui fazemos tudo funcionar e como nosso banco de dados é uma API então vamos precisar usar um módulo do Angular que se chama HTTP.CLI

Angular – Estudo do Projeto Blog

- Esse módulo vai ser responsável por através desse método HTTP fazer com que a gente consiga se comunicar com o BACKEND.

Para a gente importar esse módulo vamos no `app.module.ts` e nas imports vamos adicionar o módulo do `Http`.

Imports:

`browser Module,`

`....`

`httpClientModule`

Só que isso não basta, é preciso fazer o import dele também:

`Import{httpClient module} from '@Angular/Command/HTPP';`

Angular – Estudo do Projeto Blog

- ▶ Quando digitamos o import dentro do VSCODE e não inicializamos ele fica meio que apagadinho então vamos ao construtor pra fazer as implementações a utilizar.
- ▶ Então dentro do CONSTRUTOR já precisamos inicializar com o modelo HTTP
- ▶ Então vamos usar o comando Private **HTTP**: HttpClient

HTTP pode ser qualquer nome.

Ficando assim:

Construtor (private http:HttpClient) {}

Para poder fazer as correções com o BackEnd o Get/Post e tudo mais.

Angular – Estudo do Projeto Blog

Então vamos criar um método agora para fazer essa correção então vamos fazer um método pra listar as postagens.

Vamos chamar de:

`GetPost()` → Como método tem que ter os parênteses, dentro das classes,

Vamos colocar o endereço da minha API

`GetPost()` {

`return this` → Toda vez que trabalhamos com OO o `this` quer dizer que estou me referindo aos atributos da classe.

No caso aqui eu estou me referindo ao meu HTTP: então fica assim:

`GetPost()` {

`return this.http.get (' ')` → Dentro de aspas coloca a API

Get pode ser substituído por Put, Delete ou Post

Angular – Estudo do Projeto Blog

- Para saber o endereço da API nos vamos em **PACKAGE.JSON** no **BACKEND**, no **START** ele fala **LocalHost:3000**

Então no `GetPost()` vai ficar assim:

```
GetPost() {  
    return this.http.get('http://localhost:3000/posts')  
}
```

Então com essa linha eu já consigo o acesso aos dados da API com o método GET.

Agora vamos voltar lá na Feed agora no arquivo TS.

Angular – Estudo do Projeto Blog

- No arquivo **feed.component.ts** já temos o **service** que conecta ao Backend e agora é preciso consumir esse serviço que por sua vez vai consumir a API que contem esses dados.

Então tem que fazer essa configuração no arquivo do componente.

Principalmente dentro do CONSTRUTOR vamos fazer uma injeção de dependências e para isso vamos fazer o que est na linha:

Constructor (private PostService: PostService)

Mas antes é preciso importar o meu servisse

Import (PostService from '../servisse/Post.Service',

Agora já temos ele funcionando.

Angular – Estudo do Projeto Blog

- O arquivo já está dentro do componente.
- Outra coisa que vamos precisar. Vai ser o **MODEL** a classe quem contém os campos.
- A model contém os campos que existem lá no banco de dados então vou usar esse model para poder acessar esses campos que tem lá no banco json.
- Para isso vamos fazer a importação dela no **feed.componente.ts**.

import (Post) from “..model/Post”;

Com isso eu tenho a conexão com os posts e feito isso temos o método:

ngOnInit():void (Esse método carrega na hora que abre a aplicação. Tudo que estiver dentro dela. É o método principal desse componente.

Angular – Estudo do Projeto Blog

- Então ele vai preencher com os dados que estão vindo lá do banco de dados.
- Antes do construtor vamos criar uma variável.
- Vamos chama ela de **ListPost: Post** → Tipo de variável.
- Quem é o Post? É a minha MODEL então a variável vai ser do tipo ARRAY.
- ListPost: Post[];
- Então o que vamos colocar dentro de Subscribe.
- No vamos colocar uma função dentro dele que vai ser uma variável tipo DATA.
- **This.PostService.GetPost().subscribe((data:Post[]))**
- => **this.ListPost = data**
- **data** Vai receber
- A variável **ListPost** vai receber

Angular – Estudo do Projeto Blog

Isso quer dizer que a variável está recebendo os dados que vêm lá da API com isso já temos os dados já populados aqui.

Então é preciso chamar esse método dentro do

```
ngOnInit()void {  
    this.findPosts()  
}
```

Isso já garante que os dados estejam dentro do componente.

Agora é preciso ir lá no HTML e listar isto.

Então vamos para o `feed.component.html`.

Vamos construir toda a estrutura dos post eles serão listados aqui.

part7

Angular – Estudo do Projeto Blog

```
<div class="container">
```

```
<div class="card mb_4" *ngFor="Let Post of ListPost">
```

mb-4 → para dar uma margem ao botão.

Let Post of ListPost → É um for que do angular que vai percorrer toda a list post.

```
<div class="card-header">
```

É uma classe para título e dentro dessa classe nós vamos ter.

```
<h5 class = "card_title"> {{post.id}} </h5>]
```

post.id – Interpolação – Aqui você chama o conteúdo dos campos dentro da variável Post para imprimir.

card_title – É o título

```
</div>
```

Angular – Estudo do Projeto Blog

```
<p class="card-text">{{post.nome}} </p>  
<p class="card-text"> {{Post.mensagem}} </p>
```

Exibe o corpo da mensagem.

Angular – Estudo do Projeto Blog

Agora vamos fazer a gravação:
As postagens de fato.

Voltando lá no servisse e criar um novo método para fazer as postagens de fato.

```
GetPosts() {
```

```
..
```

```
..
```

```
PostMensagem(post:Post) {
```

Ele precisa retornar o mesmo caminho da rota. Então na rota Post eu preciso adicionar uma nova postagem.

Return this.http.post ('http://localhost:3000/Posts', Post)

post = Model

Post = Passei na requisição o Post de fato.

Angular – Estudo do Projeto Blog

No get ele retorna todas as postagens.

E aqui eu preciso fazer apenas uma postagem assim vamos precisar da intervenção da Model.
E tem que fazer o import do Model.

Import {Post} from ‘.../model/Post’;

Você digita seu texto agora e publica e grava no Backend.

Angular – Estudo do Projeto Blog

Então temos que fazer uma conexão do botão então para isso temos que voltar no:

Feed.Component.ts

E criar mais um método

..

```
cadastrarMensagem() {  
This.postService.postMensagem(this.post),  
Subscribe((data:Post) => {
```

Parâmetro

Preciso me conectar ao serviço que vai me conectar API.
Só que agora eu acesso o **postMensagem** que é o método que eu criei, lá dentro do service.

Angular – Estudo do Projeto Blog

..Para isso é preciso criar uma variável chamada Post não uma array agora apenas Post.

Post:Post = new Post;

É preciso dar um retorno para esse método.

```
This.post = data  
Location.assign ('/feed')  
}
```

Eu coloco essa rota para ele direcionar depois que o Post for cadastrado.

Angular – Estudo do Projeto Blog

.Agora temos que ir lá no HTML para fazer a conexão de fato com o botão

...

..

```
<div Class= 'd-flex justify-content-end'>
```

```
<button type = "button" class 'btn btn-outline-info w-25 "  
(click) = 'cadastrarMensagem()'>
```

```
Publicar</button>
```

```
</div>
```

....

Esse evento aciona o método Cadastrar mensagem.

Angular – Estudo do Projeto Blog

.Precisa ainda capturar as informações que estão vindo do formulário e possa para esse método cadastrar mensagem. Para isso vale precisar passar uma diretiva dentro desse campo.

Dentro desses dois campos:

Nome e Mensagem

Para isso vamos precisar uma diretiva do angular chamada **[(Angular)]** e nela vamos chamar o meu elemento **POST**

Angular – Estudo do Projeto Blog

.Dentro do Input

```
<input type="text" name="nome" id="nome"  
Class = "Form-control" placeholder = "Digite o seu nome"  
[(ngModel)] = 'post.nome'>
```

Angular – Estudo do Projeto Blog

.A mesma coisas em mensagens onde adiciona a diretiva.

Essa diretiva precisa ser importada pois isso nos vamos lá no App.module.ts e importar essa diretriz.

Imports:

BrowserModule

...

FormsModule

E precisa também fazer o import em cima.

```
Import(formsModule) from '@Angular/common/HTTP';
```



Angular – Estudo do Projeto Blog

.FIM DO MÓDULO INTERMEDIÁRIO `;

Angular – Estudo do Projeto Blog

- **Api Filmes – Estudo Avançado de Angular:**

Acessê: <https://github.com/swilliamsilva/api-filmes-curso-avancado>

Agora que você estudou no intermediário vamos para o avançado.

Baixe o código e identifique todos os componentes e o fluxo de informações.