


Outras Maneiras de Fazer Navegação Com o Angular

Nós vimos como navegar entre páginas utilizando o elemento `<a>` com o atributo `routerLink`. Mas essa não é a única maneira. Vamos considerar o seguinte cenário: o usuário irá se cadastrar no seu site. Dentro do seu componente pode ter a chamada a uma API que fará o cadastro dele no site. Quando ele acabar seu registro, você pode querer direcioná-lo automaticamente para a página de login do seu site.

Nesse caso, a navegação não poderia ser feita pelo HTML, mas podemos fazer a navegação dentro do componente. Basta, para isso, injetarmos no nosso componente um objeto do tipo Router. Esse objeto nos permite fazer a navegação dentro de uma classe de um componente.

```
1  import { Component, OnInit } from '@angular/core';
2  import { Router } from '@angular/router';
3
4  @Component({
5    selector: 'app-page1',
6    templateUrl: './page1.component.html',
7    styleUrls: ['./page1.component.css']
8  })
9  export class Page1Component implements OnInit {
10
11    constructor(
12      private router: Router
13    ) { }
14
15    ngOnInit(): void {
16
17    }
18  }
```

Agora, com nosso router injetado, nós podemos utilizar o método **`navigateByUrl()`** informando a URL que devemos acessar. Assim, seremos redirecionados para a URL informada.



```
1 import { Component, OnInit } from '@angular/core';
2 import { Router } from '@angular/router';
3
4 @Component({
5   selector: 'app-page1',
6   templateUrl: './page1.component.html',
7   styleUrls: ['./page1.component.css']
8 })
9 export class Page1Component implements OnInit {
10
11   constructor(
12     private router: Router
13   ) { }
14
15   ngOnInit(): void {
16
17     this.router.navigateByUrl('/page2')
18   }
19 }
```

Seremos redirecionados para a página **page2**.

Também podemos utilizar o método **navigate()**. Mas, diferentemente do `navigateByUrl`, o **navigate** recebe um array com uma string informando a URL que devemos acessar.



```
1  import { Component, OnInit } from '@angular/core';
2  import { Router } from '@angular/router';
3
4  @Component({
5    selector: 'app-page1',
6    templateUrl: './page1.component.html',
7    styleUrls: ['./page1.component.css']
8  })
9  export class Page1Component implements OnInit {
10
11    constructor(
12      private router: Router
13    ) { }
14
15    ngOnInit(): void {
16
17      this.router.navigate(['/page2'])
18    }
19  }
20
```

Fazemos isso pois o método nos permite passar query params para ele. Mas não iremos nos aprofundar nesse assunto por enquanto.