12

Para saber mais - Lazy Loading

Tornar possível o "carregamento preguiçoso" (ou *lazy loading*) de páginas, componentes, diretivas e pipes era a principal funcionalidade a ser viabilizada pela equipe do Ionic para a versão 3 do framework. **E eles conseguiram!**

A principal vantagem de utilizar essa técnica é a possibilidade que ganhamos de carregar certas partes da aplicação apenas quando for necessário, fazendo com que o módulo principal da aplicação seja carregado muito mais rápido!

Exemplo prático: Temos uma aplicação com 50 páginas, mas foi verificado que a maioria esmagadora dos usuários só utiliza no máximo 2 dessas páginas antes de sair do aplicativo. Isso quer dizer que na maior parte das situações, 48 páginas estão sendo carregadas pelo aplicativo de forma desnecessária, consumindo recursos do celular e fazendo com que o aplicativo demore mais para ser carregado!

O *lazy loadgin* ajuda a resolver casos como esse de modo bem simples! Em vez de termos tudo carregado no módulo principal da aplicação, podemos ter submódulos que somente serão carregados quando necessário! É exatamente por isso que, por exemplo, ao criar uma página chamada `Home` utilizando a Ionic CLI duas coisas acontecem:

1. Surgirá um arquivo `src/pages/home/home.module.ts`, que, como vimos durante a aula, já estará **praticamente** todo configurado para utilizar o *lazy loading*! Bastando apenas que acrescentemos a seção `exports` e declarando dentro dela o componente que acabamos de criar.
2. A classe do nosso componente `src/pages/home/home.ts`, além de vir anotada com o *decorator* `@Component`, também terá o *decorator* `@IonicPage`, indicando que aquele componente é candidato a ser carregado de modo *lazy*.

Por fim, tendo adicionado o *decorator* `@IonicPage` ao nosso componente e uma vez que tenhamos configurado corretamente o seu módulo específico, para fazermos uma navegação de maneira lazy basta utilizar uma instância de `NavController` da seguinte forma (considerando que o nome da classe do componene criado seja `HomePage`, como nesse exemplo):

```
navCtrl.push('HomePage');
```

No entanto, nem tudo são flores! Se um dia mudarmos o nome de nosso componete para `CasaPage`, teríamos que lembrar de todos os pontos onde fizemos a navegamos utilizando a string `'HomePage'` e atualizar para `'CasaPage'`.

Entretanto, a boa notícia é que esse curso é com o instrutor Gabriel Leite! Por isso, irei resgatar todos os meus alunos das trevas e trazê-los para a luz, passando para vocês um conhecimento que nem mesmo a documentação do Ionic mostra!

O pulo do gato aqui é utilizar a propriedade `name` que todo componente do `Angular` tem por baixo dos panos! Portanto, podemos reescrever nossa navegação *lazy* da seguinte forma:

```
navCtrl.push(HomePage.name)
```

Assim, garantimos que se um dia refatorarmos o nome do nosso componente para `CasaPage`, a maioria das boas IDEs saberão refatorar também todos os pontos onde um dia nos referenciamos à esse componente, afinal de contas, agora não estamos mais usando strings!

Por fim, mas não menos importante, caso queiram se aprofundar um pouco mais sobre esse assunto, indico dois bons posts no blog do Ionic (em inglês):

Ionic and Lazy Loading Pt 1 (https://blog.ionicframework.com/ionic-and-lazy-loading-pt-1/)

Ionic and Lazy Loading Pt 2 (https://blog.ionicframework.com/ionic-and-lazy-loading-pt-2/)