

Melhorando a navegação com slugs

Seja bem-vindo ao curso de Rails 3 do alura

Recomendamos esse curso para aquelas pessoas que já conheçam o rails3 ou trabalha em algum projeto legado do mesmo. Para quem é novo e deseja começar a aprender do rails agora recomendamos o curso de rails 4:

<https://www.alura.com.br/course/ruby-on-rails-4-do-zero> (<https://www.alura.com.br/course/ruby-on-rails-4-do-zero>)

Temos hoje uma listagem de todas as vagas premium. Para navegar entre elas, basta clicar no link "Show" existente em cada uma das vagas. Mas repare no link das vagas, por exemplo, <http://localhost:3000/jobs/3> (<http://localhost:3000/jobs/3>) : eles não são "bonitos".

O número "3" ali, que é o id do emprego tem problemas. O usuário final pode saber exatamente quantos objetos (empregos) desse tipo temos, e trabalhar com essa informação de alguma maneira. Ou, no pior caso, tentar fazer algum tipo de ataque à aplicação, como forçar URLs de deleção, e etc.

Além disso, URLs como essas são péssimas para o Google (ou qualquer outro search engine). O Google usa a própria URL como critério para identificar qual informação existe naquela página. Ou seja, uma URL mais semântica pode te ajudar inclusive a subir na busca do Google (algo muito importante hoje em dia).

Precisamos fazer então com que nossas URIs deixem de ser baseadas em números, e passem a ser baseadas em algum texto. Por exemplo, a URL <http://localhost:3000/jobs/3-professor-universitario> (<http://localhost:3000/jobs/3-professor-universitario>) é bem mais semântica que a anterior.

Vamos a classe `Job.rb`. Ali temos 3 atributos declarados: a descrição, o título e se ele é premium ou não. Repare que em nosso ERB, o Rails gera a URL baseada no atributo "id", existente em todos os nossos modelos. Mas como é que ele faz isso? Será que é uma convenção?

O Rails, na hora de gerar a URL, invoca o método chamado `to_param`. E, se quisermos mudar isso, basta sobrescrever esse método. Nesse método, retornaremos o id junto do nome da vaga:

```
def to_param
  "#{id}-#{title}"
end
```

Se atualizarmos a página, repare que a URL agora é composta pelo id, um hífen, e o nome da vaga. E o interessante é: se clicarmos no link, ele ainda consegue buscar a vaga!

Mas como ele é capaz de buscar no banco de dados um emprego com o id "3-teacher", por exemplo? Isso pode ser explicado pela maneira que o Ruby converte string para inteiros. Se abrirmos o `irb`, e escrevermos o seguinte código:

```
"3-teacher".to_i
```

O resultado é "3". O ruby remove o resto da string, nos deixando somente o id numérico. Dessa forma, o Rails buscará pelo id 3.

Ainda temos o problema que se ele acessar a url "3-umtextoqualquer", ele ainda funcionará. Resolveremos esse problema nos próximos capítulos desse curso.

