



## Criando nosso método find\_by\_slug

Nossas URIs agora estão um pouco mais amigáveis como no caso de <http://localhost:3000/jobs/4-teacher> (<http://localhost:3000/jobs/4-teacher>). Essas URIs tem diversas vantagens, conforme discutimos no capítulo anterior.

Vimos também que quando fazemos `to_i()`, a String "4-teacher" vira o número 4, e isso faz com que o Rails consiga buscar o emprego com aquele número. Mas se escrevermos um outro texto qualquer, como por exemplo, "4-outrotexto", ele também encontra a mesma vaga de emprego.

Agora a dúvida: o que fazer quando o usuário digita um texto qualquer na URL, junto de um ID válido? Um erro 404, dizendo que não existe esse emprego? Ou redirecionar ele para a URL correta? As duas são corretas e plausíveis, e depende do qual comportamento você quer. Aqui, implementaremos os dois.

Vamos começar pelo 404. Lembre-se que sobreescrevemos o método `to_param()` e que no `jobs_controller.rb`, fazemos um `find()` pelo ID.

Uma primeira solução seria então sobreescrever o método `find()` e fazer com que ele só devolva o objeto caso o slug (que é o nome que damos para aquela string na URL) seja a esperada. Mas sobreescrever métodos do ActiveRecord, geralmente não é uma boa ideia, já que podemos quebrar o comportamento que é esperado em outras classes ou mesmo plugins que usamos em nosso projeto.

Vamos então criar um novo método, que busca por slug, em nossa classe `Job` :

```
def self.find_by_slug(slug)
  result = find(slug)
  if result.to_param == slug
    result
  else
    raise ActiveRecord::RecordNotFound
  end
end
```

Repare que no código acima, buscamos no banco pelo ID (final, o `to_i` vai converter a string para o número). Em seguida, comparamos se o `to_param()` daquele resultado é igual ao slug recebido. Se for igual, retorna o objeto; caso contrário, lança uma exceção de registro não encontrado.

De volta ao controller, vamos mudar o `find()` para `find_by_slug` :

```
@job = Job.find_by_slug(params[:id])
```

Pronto. O 404 está implementado, afinal se a exceção `RecordNotFound` chegar, o Rails devolve 404.

No próximo capítulo, mudaremos a estratégia, e redirecionaremos para a URL certa.