

02

## Corrigindo o acesso a slugs inválidos

No capítulo anterior fizemos com que, caso o usuário digite um slug inválido, ele retorne um 404 para o usuário. Agora vamos fazer com que, caso ele acerte um id mas erre o texto, a aplicação redirecione para a URL correta.

Para isso, vamos começar mudando a implementação do método `find_by_slug`. Ao invés dele lancá uma exceção do Rails e gerar o 404, vamos lançar uma exceção nossa.

Vamos criar a nossa exceção dentro da nova pasta `lib/exceptions/slug_error.rb`:

```
class SlugError < StandardError
end
```

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

Vamos agora configurar nossa aplicação para sempre abrir o arquivo "slug\_error.rb". No arquivo `application.rb`. Veja a configuração `config.autoload_paths`, que está comentada. Nela, podemos colocar os diretórios que devem ser automaticamente carregados pela aplicação:

```
config.autoload_paths += %W({config.root}/lib/exceptions)
```

Nesse momento, se abrirmos o browser e acessarmos uma URL com um slug inválido, a exceção "SlugError" acontece.

Precisamos agora ensinar nosso controller a se recuperar dessa exceção. Mas precisamos fazer isso de forma genérica, pois isso pode acontecer em todos os métodos do controller. Para isso, escrevemos fora de qualquer método, mas dentro do controller o seguinte código:

```
rescue_from SlugError, :with => :rescue_from_slug_error
```

Estamos aí dizendo que sempre que um SlugError ocorrer, o método `rescue_from_slug_error` deve ser invocado.

Vamos ao método. Vamos começar apenas fazendo com que ele escreva algo na tela para testar.

```
def rescue_from_slug_error
  render :text => 'slug error'
end
```

Se agora acessarmos uma URL com slug inválido, o texto acima aparece.

Agora vamos lá. Vamos fazer com que esse método redirecione para o objeto real. Para isso, precisamos ter em algum lugar o objeto real. Vamos passá-lo como parâmetro para a exceção:

```
def self.find_by_slug(slug)
  result = find(slug)
  if result.to_param == slug
    result
  else
    raise SlugError.new(result)
  end
end

class SlugError < StandardError
  attr_reader :objeto
  def initialize(objeto)
    @objeto = objeto
  end
end
```

Com isso escrito, agora podemos no método de recuperação, redirecionar para o objeto que está guardado na exception:

```
def rescue_from_slug_error exception
  redirect_to exception.objeto
end
```

Agora sim. Se tentarmos acessar uma URL com um ID válido mas texto errado, o Rails vai lançar a exceção, e o método de recuperação vai redirecionar para o objeto certo, com a URI adequada.

Essa é geralmente uma solução bem elegante. Como esse tipo de recuperação pode ser bem comum na sua aplicação, é bem comum que o coloquemos no controller mais genérico, que é o `ApplicationController` :

```
class ApplicationController < ActionController::base

  rescue_from SlugError, :with => :rescue_from_slug_error
  def rescue_from_slug_error exception
    redirect_to exception.objeto
  end

  // código continua aqui
end
```

Agora esse comportamento será válido para todos os nossos controladores!