

Refatorando + Rspec

Dia 5



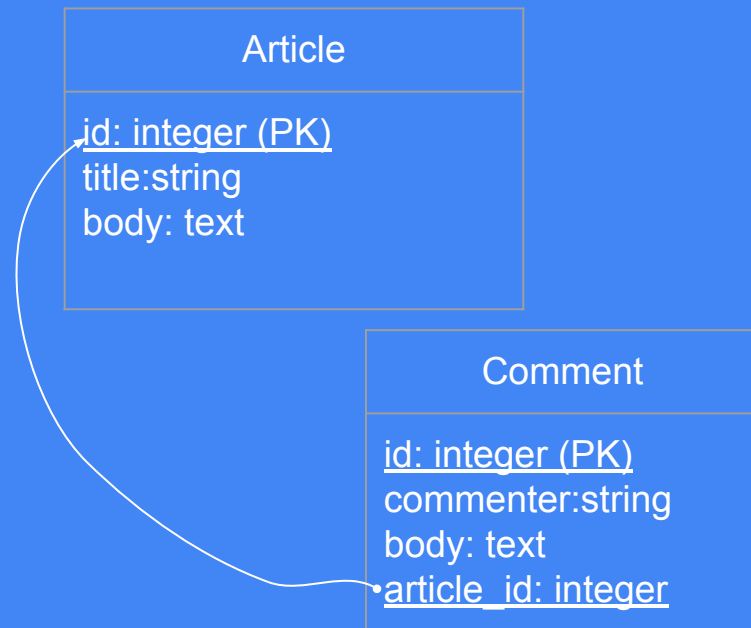
Agenda

Agenda

- Refatorando o Blog
- Testes com Rspec

Refatorando o Blog

Associação dos Models



Permitindo o usuário adicionar
comentários (como ficou)

Permitindo o usuário adicionar comentários

Model ✓

Controller/Rota ✓ Sempre que fizer algo no controller, lembre das rotas

View <-

articles/show.html.erb

```
<h2>Add a comment:</h2>
<%= form_with url: comments_path do |form| %>
  <%= form.hidden_field :article_id, :value => @article.id%>
  <p>
    Commenter:<br>
    <%= form.text_field :commenter %>
  </p>
  <p>
    Body:<br>
    <%= form.text_area :body %>
  </p>
  <p>
    <%= form.submit %>
  </p>
<% end %>
```

Permitindo o usuário adicionar comentários

Model ✓

Controller/Rota ✓ Sempre que fizer algo no controller, lembre das rotas

View <-

CommentsController

def create

```
  Comment.create!(commenter: params[:commenter], body: params[:body], article_id:  
params[:article_id])  
  redirect_to article_path(params[:article_id])
```


Refatorando as Rotas

Refatorando as rotas

https://guides.rubyonrails.org/getting_started.html#adding-a-route-for-comments

config/routes.rb

```
Rails.application.routes.draw do  
  root "articles#index"
```

```
  resources :articles do  
    resources :comments, only: [:create]  
  end  
end
```

Refatorando o formulário

Refatorando o formulário

https://guides.rubyonrails.org/getting_started.html#generating-a-controller

```
<h2>Add a comment:</h2>
```

```
<%= form_with model: [ @article, @article.comments.build ] do |form| %>
```

```
  <p>
```

```
    <%= form.label :commenter %><br>
```

```
    <%= form.text_field :commenter %>
```

```
  </p>
```

```
  <p>
```

```
    <%= form.label :body %><br>
```

```
    <%= form.text_area :body %>
```

```
  </p>
```

```
  <p>
```

```
    <%= form.submit %>
```

```
  </p>
```

```
<% end %>
```

Refatorando o controller

Refatorando as rotas

https://guides.rubyonrails.org/getting_started.html#generating-a-controller

comments_controller.rb

```
def create
  article = Article.find(params[:article_id])
  comment = @article.comments.create(comment_params)
  redirect_to article_path(article)
end
```

```
private
def comment_params
  params.require(:comment).permit(:commenter, :body)
end
```