

Indo além

Dia 4



Agenda

Agenda

- Criando um Blog
- Associações com Active Record
- Usando Rake tasks
- Conhecendo alguns helpers
- Conhecendo algumas gems

Criando um Blog

Tabela

Article

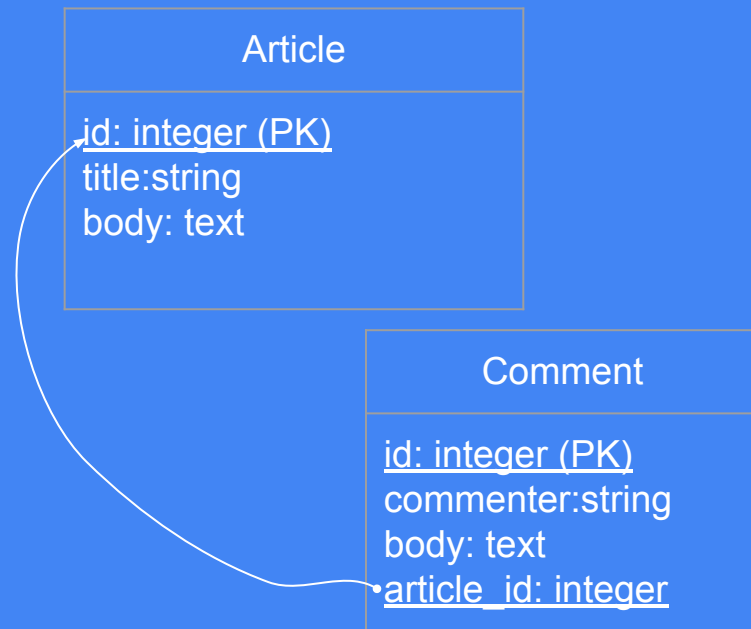
id: integer (PK)
title:string
body: text

Criando um Blog

Iniciando a aplicação

```
rails new blog  
rails g scaffold Article title:string body:text  
rails db:create db:migrate
```

Associação



Criando um Blog

Criando o model Comment

```
rails g model Comment commenter:string body:text article_id:integer  
OU
```

```
rails g model Comment commenter:string body:text article:references
```

```
rails db:migrate
```


Associações com Active Record

Associações com Active Record

Associações

- https://guides.rubyonrails.org/association_basics.html

belongs_to

- https://guides.rubyonrails.org/association_basics.html#detailed-association-reference

Associações com Active Record

Associações

- https://guides.rubyonrails.org/association_basics.html

belongs_to

- https://guides.rubyonrails.org/association_basics.html#detailed-association-reference

`Comment.create(commenter: 'jackson@jack.com', body: 'muito legal esse artigo', article_id:
Article.first.id)`

`Comment.create(commenter: 'jackson@jack.com', body: 'muito legal esse artigo', article: Article.first)`

Associações com Active Record

Associações

has_many

```
class Article
  has_many :comments
```

Article.comments

```
Article.first.comments.create(commenter: 'jose', body: 'nooooossaaa')
```

Mostrando os comentários

Mostrando os comentários

articles/show.html.erb

```
<h3>Comments</h3>
<hr/>
<% @article.comments.each do |comment| %>
  <div>
    <p><%= comment.body %><p>
    <em><%= comment.commenter %></em>
    <hr/>
  </div>
<% end %>
```

Permitindo o usuário adicionar
comentários

Permitindo o usuário adicionar comentários

Model ✓
Controller <-
View

```
rails generate controller Comments
```

```
def create  
  # Aqui vai criar o comentário relacionado ao artigo  
end
```


Permitindo o usuário adicionar comentários

Model ✓
Controller <-
View

```
rails generate controller Comments
```

```
def create  
  # Aqui vai criar o comentário relacionado ao artigo  
end
```

Rota?

Permitindo o usuário adicionar comentários

Model ✓

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

View <-

```
rails generate controller Comments
```

```
def create
```

```
  # Aqui vai criar o comentário relacionado ao artigo
```

```
end
```

Rota?

```
resources :comments, only: [:create]
```

Permitindo o usuário adicionar comentários

Model ✓

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

View <-

articles/show.html.erb

https://guides.rubyonrails.org/form_helpers.html

https://apidock.com/rails/v5.2.3/ActionView/Helpers/FormHelper/form_with

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_with é um helper // https://apidock.com/
```

```
<p>
```

```
  Commenter:<br>
```

```
  <%= form.text_field :commenter %>          # text_field é um helper // https://apidock.com/
```

```
</p>
```

```
<p>
```

```
  Body:<br>
```

```
  <%= form.text_area :body %>          # text_area é um helper // https://apidock.com/
```

```
</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: ??)

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 %> https://apidock.com/rails/v5.2.3/ActionView/Helpers/FormHelper/hidden_field
```

```
<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])
```

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])
```


Usando rake tasks

Usando rake tasks

- <https://www.seancdavis.com/posts/how-to-write-a-custom-rake-task/>

`rails -T` // T maiúsculo, para ver as Tasks disponíveis

Usando rake tasks

```
rails -T // T maiúsculo, para ver as Tasks disponíveis
```

```
rails g task dev setup
```

Usando rake tasks

```
rails -T // T maiúsculo, para ver as Tasks disponíveis
```

```
rails g task dev setup
```

dev.rake

```
desc "Add fake articles and comments"
task setup: :environment do
  article = Article.create!(title: 'Title 1', body: 'body 1')
  article.comments.create(commenter: 'jack', body: 'my comment')
  puts article.inspect
  puts article.comments.inspect
end
```

Usando rake tasks

```
rails -T // T maiúsculo, para ver as Tasks disponíveis
```

```
rails g task dev setup
```

dev.rake

```
desc "Add fake articles and comments"
task setup: :environment do
  10.times do
    article = Article.create!(title: 'Title 1', body: 'body 1')
    article.comments.create(commenter: 'jack', body: 'my comment')
    puts article
    puts article.comments
  end
end
```

Usando rake tasks

Exemplo de gem para terminal

> <https://ttytoolkit.org/>

> <https://github.com/piotrmurach/tty-spinner>

Conhecendo a gem faker

Conhecendo a gem faker

- <https://github.com/faker-ruby/faker>

```
rails g task dev setup
```

Gemfile

```
gem 'faker'
```

```
ou
```

```
gem 'faker', :git => 'https://github.com/faker-ruby/faker.git', :branch => 'master'
```

```
bundle install
```


Conhecendo a gem faker

dev.rake

```
desc "Add fake articles and comments"
task setup: :environment do
  10.times do
    article = Article.create!(title: Faker::Lorem.paragraph(sentence_count: 1), body:
Faker::Lorem.paragraph)
    article.comments.create(commenter: Faker::Name.name, body: Faker::Lorem.paragraph)
    puts article.inspect
    puts article.comments.inspect
  end
end
```