

Ruby on Rails 5.x

Do início ao fim!

Entendendo e usando o counter_cache

Entendendo e usando o counter_cache

- Counter Cache

https://guides.rubyonrails.org/association_basics.html#options-for-belongs-to-counter-cache

Entendendo e usando o counter_cache

- O **counter_cache** fica sempre como uma opção do **belongs_to**, sendo assim, teremos em **time_to_answer/app/models/question.rb**

```
class Question < ApplicationRecord
  belongs_to :subject, counter_cache: true, inverse_of: :questions
  ...
end
```

Entendendo e usando o counter_cache

- Agora precisamos criar uma migração para adicionar um campo para fazer o cache em Subjects...

```
rails g migration AddQuestionsCountToSubjects questions_count:integer
```

```
rails db:migrate
```

Entendendo e usando o counter_cache

- Altere a view...

app/views/site/shared/_questions.html.erb

```
<span class="label label-info" style="margin-right: 10px;">
  <%= link_to "#{question.subject.description}
  (#{question.subject.questions_count})",
  site_search_subject_path(question.subject_id, question.subject.description),
  style:"color: white;" %>
</span>
```

Entendendo e usando o counter_cache

- Faça um teste e veja que ele já atualiza o contador...

```
x = Subject.last  
Subject.reset_counters(x.id, :questions)
```

Entendendo e usando o counter_cache

- Crie uma task para resetar o contador de todos os assuntos...

```
desc "Reseta o contador dos assuntos"
task :reset_subject_counter: :environment do
  show_spinner("Resetando contador dos assuntos...") do
    Subject.find_each do |subject|
      Subject.reset_counters(subject.id, :questions)
    end
  end
end
```