progress engine
we will develop it

#tceh

2016

Врубиться в Ruby

Лекция 25

Лекция 25

- Поиск в rails
- Ransack
- Поисковые движки

Поиск - простой пример

```
def index
  if params[:search]
    @posts = Post.search(params[:search]).order("created_at DESC")
  else
    @posts = Post.all.order('created_at DESC')
  end
end
```

```
def self.search(search)
  where("name LIKE ?", "%#{search}%")
  where("content LIKE ?", "%#{search}%")
end
```

```
<%= form_tag(posts_path, :method => "get", id:
"search-form") do %>

<%= text_field_tag :search, params[:search],
placeholder: "Search Posts" %>

<%= submit_tag "Search" %>
<% end %>
<% if @posts.present? %>
  <%= render @posts %>
<% else %>
  <p>There are no posts containing the term(s) <%=
params[:search] %>.</p>
<% end %>
```

Ransack

https://github.com/activerecord-hackery/ransack

```
def index
  @q = Person.ransack(params[:q])
  @people = @q.result.includes(:articles).page(params[:page])

  # or use `to_a.uniq` to remove duplicates (can also be done in the
view):
  @people =
@q.result.includes(:articles).page(params[:page]).to_a.uniq
end
```

```
<%= search_form_for @q do |f| %>

  # Search if the name field contains...
  <%= f.label :name_cont %>
  <%= f.search_field :name_cont %>

  # Search if an associated articles.title starts with...
  <%= f.label :articles_title_start %>
  <%= f.search_field :articles_title_start %>

  # Attributes may be chained. Search multiple attributes for one value...
  <%= f.label :name_or_description_or_email_or_articles_title_cont %>
  <%= f.search_field :name_or_description_or_email_or_articles_title_cont %>

  <%= f.submit %>
<% end %>
```

```ruby
class Post < ActiveRecord::Base
  belongs_to :author

  # Abbreviate :author_first_name_or_author_last_name to :author

ransack_alias :author, :author_first_name_or_author_last_name
end
```

```
class Post < ActiveRecord::Base
  belongs_to :author

  # Abbreviate :author_first_name_or_author_last_name to :author

ransack_alias :author, :author_first_name_or_author_last_name
end
```

Associated models

```
class Employee < ActiveRecord::Base
  belongs_to :supervisor

  # has attributes first_name:string and last_name:string
end

class Department < ActiveRecord::Base
  has_many :supervisors

  # has attribute title:string
end

class Supervisor < ActiveRecord::Base
  belongs_to :department
  has_many :employees

  # has attribute last_name:string
end
```

Associated models

```
class SupervisorsController < ApplicationController
  def index
    @q = Supervisor.ransack(params[:q])
    @supervisors = @q.result.includes(:department, :employees)
  end
end
```

Associated models

```
<%= search_form_for @q do |f| %>
  <%= f.label :last_name_cont %>
  <%= f.search_field :last_name_cont %>

  <%= f.label :department_title_cont %>
  <%= f.search_field :department_title_cont %>

  <%= f.label :employees_first_name_or_employees_last_name_cont %>
  <%=
f.search_field :employees_first_name_or_employees_last_name_cont %>

  <%= f.submit "search" %>
<% end %>
...
<%= content_tag :table do %>
  <%= content_tag :th, sort_link(@q, :last_name) %>
  <%= content_tag :th, sort_link(@q, :department_title) %>
  <%= content_tag :th, sort_link(@q, :employees_last_name) %>
<% end %>
```

Postgres full text search

http://rachbelaid.com/postgres-full-text-search-is-good-enough/

https://github.com/Casecommons/pg_search

https://github.com/textacular/textacular https://robots.thoughtbot.com/implementing-multi-table-full-text-search-with-postgres

Поисковые движки

1 - Sphinx (http://sphinxsearch.com)

2 - Solr (http://lucene.apache.org/solr/)

3 - Elasticsearch

Практика

https://github.com/elastic/elasticsearch-rails

```
gem 'elasticsearch-model'
gem 'elasticsearch-rails'
```

```
require 'elasticsearch/model'

class Article < ActiveRecord::Base
  include Elasticsearch::Model
  include Elasticsearch::Model::Callbacks
end
Article.import # for auto sync model with elastic search
```

```
def search
  if params[:q].nil?
    @articles = []
  else
    @articles = Article.search params[:q]
  end
end
```

```
def self.search(query)
  __elasticsearch__.search(
    {
      query: {
        multi_match: {
          query: query,
          fields: ['title^10', 'text']
        }
      }
    }
  )
end
```

```
def self.search(query)
  __elasticsearch__.search(
    {
      query: {
        multi_match: {
          query: query,
          fields: ['title^10', 'text']
        }
      },
      highlight: {
        pre_tags: ['<em>'],
        post_tags: ['</em>'],
        fields: {
          title: {},
          text: {}
        }
      }
    }
  )
end
```

https://www.sitepoint.com/full-text-search-rails-elasticsearch/

https://github.com/ankane/searchkick

```
class Product < ActiveRecord::Base
  searchkick
end
```

Product.reindex

```
products = Product.search "apples"
products.each do |product|
  puts product.name
end
```

Практика

Вопросы?

+7 (926) 889-16-32

alec@alec-c4.com

http://alec-c4.com/