Home (/) / 第二期 Rails 全栈班（課程） (/courses/8) / Syllabus (/courses/8/syllabus) / （解答）招聘网站第五部分

🡐 Prev (/posts/297) Next 🡒 (/posts/

（解答）招聘网站第五部分

目标

- 加入 sidebar
- 限制已隐藏职缺不能存取
- 将“职缺”状态换成图示
- 设计能“隐藏”“重新显示”的按钮

步骤

Step 0

git checkout -b step5

Step 1 加入 admin layouts

```
touch app/views/layouts/admin.html.erb
```

`app/views/layouts/admin.html.erb`

```
<!DOCTYPE html>
<html>
    <head>
        <title>Job Listing </title>
        <%= csrf_meta_tags %>

        <%= stylesheet_link_tag    'application', media: 'all', 'data-turbolinks-track': 'reload' %>
        <%= javascript_include_tag 'application', 'data-turbolinks-track': 'reload' %>
    </head>

    <body>

        <div class="container-fluid">
            <%= render "common/navbar" %>
            <%= render "common/flashes" %>
            <div class="row">
              <div class="col-md-2">
                sidebar
              </div>

              <div class="col-md-10">
                <%= yield %>
              </div>
        </div>

        <%= render "common/footer" %>

    </body>
</html>
```

然后在 `app/controllers/admin/jobs_controller.rb` 加入一行 `layout "admin”`

`app/controllers/admin/jobs_controller.rb`

```
class Admin::JobsController < ApplicationController
  before_filter :authenticate_user!, only: [:new, :create, :update, :edit, :destroy]
  before_filter :require_is_admin
  layout "admin"
```

就可以看到新的 layout 被套用了。

Step 2 加入 正确的 sidebar

接着我们要修改 app/views/layout/admin.html.erb
把 sidebar 换成

```
    <ul class="nav nav-pills nav-stacked" style="max-width: 300px; padding-top:20px;">
       <li class="active"><%= link_to("Jobs", admin_jobs_path ) %> </li>

    </ul>
```

成果就会变成这样：

Step 3: 把 Public / Hidden 换成图示（用 FontAwesome）

原先的 (Public / Hidden) 实在太丑了：

我们应该要换成图示，像这样：

那么要怎么做呢?
有一个厉害的网站叫做 FontAwesome，提供各式各样厉害的图示：http://fontawesome.io/icons/ (http://fontawesome.io/icons/)

如何安装 FontAwesome？修改 Gemfile 挂上

```
gem "font-awesome-rails"
```

然后执行
`bundle install`

接着修改 `app/assets/stylesheets/application.scss`
加入：

```
app/assets/stylesheets/application.scss
```

```
*= require font-awesome
```

也加入：

```
app/assets/stylesheets/application.scss
```

```
@import "font-awesome";
```

然后重开 `rails server`

接着我们要换掉 Helper，修改 `app/helpers/jobs_helper.rb` 当中的 `render_job_status(job)`

```
app/helpers/jobs_helper.rb
```

```
  def render_job_status(job)
    if job.is_hidden
      content_tag(:span, "", :class => "fa fa-lock")
    else
      content_tag(:span, "", :class => "fa fa-globe")
    end
  end
```

这样你的后台列表就会变成：

Step 4: 限制已经“被隐藏的工作”，不可以被看到。

把 `app/controllers/jobs_controller.rb` 当中的 `show action`
改成

```
app/controllers/jobs_controller.rb
```

```
def show
  @job = Job.find(params[:id])

  if @job.is_hidden
    flash[:warning] = "This Job already archieved"
    redirect_to root_path
  end
end
```

成果就会是这样：

Step 5 : 设计按钮，将“隐藏职缺”改成是一个按钮，按下去“隐藏职缺”，再按一次“显示职缺”

如同这样：

首先呢，修改 `app/views/admin/jobs/index.html.erb`
加上一个判断式，显示 Publish / Hide 的按钮

app/views/admin/jobs/index.html.erb

```
<td>
  <%= link_to("Edit", edit_admin_job_path(job)) %> |
  <%= link_to("Destroy", admin_job_path(job), :method => :delete, :data => { :confirm => "Are you sure?" }) %>

  |
  <% if job.is_hidden %>

    <%= link_to("Publish", "#", :class => "btn btn-xs btn-default") %>
  <% else %>
    <%= link_to("Hide", "#", :class => "btn btn-xs btn-default") %>
  <% end %>
</td>
```

接着呢?
再把 `#` 改成“正确的路径”

app/views/admin/jobs/index.html.erb

```
<% if job.is_hidden %>

  <%= link_to("Publish", publish_admin_job_path(job) , :method => :post, :class => "btn btn-xs btn-default") %>
<% else %>
  <%= link_to("Hide", hide_admin_job_path(job), :method => :post,  :class => "btn btn-xs btn-default") %>
<% end %>
```

也改 routing 加上去 publish 与 hide

config/routes.rb

```
namespace :admin do
  resources :jobs do
    member do
      post :publish
      post :hide
    end
  end
end
```

此时别高兴呢！若你按下去 publish 的按钮，他铁定跟你说 publish action 还没设计：

Step 6 : 设计 publish / hide 的 action

所以你还得在 `app/controllers/admin/jobs_controller.rb` 加入：

app/controllers/admin/jobs_controller.rb

```
  def publish
    @job = Job.find(params[:id])
    @job.is_hidden = false
    @job.save
    redirect_to :back
  end

  def hide
    @job = Job.find(params[:id])
    @job.is_hidden = true
    @job.save
    redirect_to :back
  end
```

Step 7 : 重构到 model

但我们觉得这样写

app/controllers/admin/jobs_controller.rb

```
  def publish
    @job = Job.find(params[:id])
    @job.is_hidden = false
    @job.save
    redirect_to :back
  end
```

太土炮了！

要是我们这样写：

app/controllers/admin/jobs_controller.rb

```
  def publish
    @job = Job.find(params[:id])
    @job.publish!
    redirect_to :back
  end
```

看起来才牛逼啊！

如何做呢？首先到 `app/models/job.rb` 实作 `publish!` 与 `hide!`

app/models/job.rb

```
class Job < ApplicationRecord
  def publish!
    self.is_hidden = false
    self.save
  end

  def hide!
    self.is_hidden = true
    self.save
  end
end
```

再去把 `app/controllers/admin/jobs_controller.rb` 中的两个 action 换掉：

app/controllers/admin/jobs_controller.rb

```
  def publish
    @job = Job.find(params[:id])
    @job.publish!

    redirect_to :back
  end

  def hide
    @job = Job.find(params[:id])

    @job.hide!

    redirect_to :back
  end
```

Step 8 : git 储存

```
git add .
git commit -m "implement step 5"
```

Prev (/posts/297)

Next (/posts/