**用Ruby on Rails创建身份验证项目**

**一、创建项目**

* 输入rails new *AlbumApp*创建项目
* 输入cd *albumapp*进入项目子目录

**二、创建模型(Model)**

* 输入rails generate model *Album*创建第一个模型*Album*
* 编辑app/models/*album*.rb如下：

class *Album* < ActiveRecord::Base

has\_many :*photos* #表示每个*album*表可对应多个*photo*表

end

* 编辑db/migrate/*<timestamp1>*\_create\_*album*s.rb文件，其中*timestamp1*为创建*Album*模型的时间戳，添加字段如下：

class Create*Album*s < ActiveRecord::Migration

def change

create\_table :*album*sdo |*t*|

*t*.string :*title* #添加*title*字符串字段

*t*.string :*image* #添加*image*字符串字段

*t*.timestamps #自动生成created\_at和updated\_at字段

end

end

end

* 输入rails generate model *Photo*创建第二个模型*Photo*
* 编辑app/models/*photo*.rb如下：

class *Photo* < ActiveRecord::Base

belongs\_to :*album* #表示每个*photo*表属于单一*album*表

end

* 编辑db/migrate/*<timestamp2>*\_create\_*photo*s.rb文件，其中*timestamp2*为创建*Photo*模型的时间戳，添加字段如下：

class Create*Photo*s < ActiveRecord::Migration

def change

create\_table :*photo*sdo |*t*|

*t*.text :*caption* #添加*caption*文本字段

*t*.string :*image* #添加*image*字符串字段

*t*.references :*album* #添加*album*\_id外键字段，用于关联*album*

*t*.timestamps #自动生成created\_at和updated\_at字段

end

end

end

* 输入rails generate model *User*创建第三个模型*User*
* 编辑app/models/*user*.rb如下：

class *User* < ActiveRecord::Base

has\_secure\_password #使用安全保存密码的方法，此方法依赖bcrypt Gem

end

* 编辑Gemfile，添加一行内容：

gem 'bcrypt', '~> 3.1.7'

* 输入bundle install安装bcrypt
* 编辑db/migrate/*<timestamp3>*\_create\_*user*s.rb文件，其中*timestamp3*为创建*User*模型的时间戳，添加字段如下：

class Create*User*s < ActiveRecord::Migration

def change

create\_table :*user*sdo |*t*|

*t*.string :*fname* #添加*fname*字符串字段

*t*.string :*lname* #添加*lname*字符串字段

*t*.string :*email* #添加*email*字符串字段

*t*.string :*password*\_digest #添加*password*\_digest字符串字段

#由bcrypt生成*password*的散列值

*t*.timestamps #自动生成created\_at和updated\_at字段

end

end

end

* 输入rake db:migrate以更新数据库结构
* 编辑db/seeds.rb样本数据文件如下：

*carlos* = *User*.create(*fname*: '*Carlos*', lname: '...', email: '...', password: '...', password\_confirmation: '...')

*freida* = *User*.create(*fname*: '*Frieda*', lname: '...', email: '...', password: '...', password\_confirmation: '...')

*julian* = *User*.create(*fname*: '*Julian*', lname: '...', email: '...', password: '...', password\_confirmation: '...')

*soccer* = *Album*.create(*title*: "...", *image*: "#")

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *soccer*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *soccer*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *soccer*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *soccer*.id)

*birthday* = *Album*.create(*title*: "...", *image*: "#")

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *birthday*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *birthday*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *birthday*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *birthday*.id)

*hiking* = *Album*.create(*title*: "...", *image*: "#")

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *hiking*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *hiking*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *hiking*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *hiking*.id)

*pets* = *Album*.create(*title*: "...", *image*: "#")

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *pets*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *pets*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *pets*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *pets*.id)

*wedding* = *Album*.create(*title*: "...", *image*: "#")

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *wedding*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *wedding*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *wedding*.id)

*Photo*.create(*caption*: "...", *image*: "#", *album*\_id: *wedding*.id)

* 输入rake db:seed将样本数据添加到数据库

**三、创建控制器(Controller)**

* 编辑app/controllers/application\_controller.rb文件，添加控制器逻辑如下：

class ApplicationController < ActionController::Base

protect\_from\_forgery with: :exception

helper\_method :*current\_user* #将*current\_user*作为辅助方法，

#供视图逻辑调用，默认Application控制器定义的方法可被其它控制器调用

def *current\_user* #返回当前会话的用户id，若尚未登录则为空值

@*current\_user* ||= *User*.find(session[:*user*\_id]) if

session[:*user*\_id]

end

def *require\_user* #若尚未登陆，转至登陆页面

redirect\_to '/*login*' unless *current\_user*

end

end

* 输入rails generate controller *Albums*创建控制器*Albums*
* 编辑app/controllers/*albums*\_controller.rb文件，添加控制器逻辑如下：

class *Albums*Controller < ApplicationController

before\_action :*require\_user*, only: [:*index*, :*show*]

# 调用*index*和*show* action之前先调用*require\_user*方法

def *index* # 根据ruby标准定义，*index*代表显示所有记录信息

@*albums* = *Album*.all #将模型从数据库取得的所有数据传给视图

end

def *show* # 根据ruby标准定义，*show*代表显示某条记录信息

@*album* = *Album*.find(params[:id]) #查找:id对应的数据并传给视图

@*photos* = @*album*.*photos*

# 查找@*album*记录对应的所有*photo*数据并传给视图

end

end

* 输入rails generate controller *Users*创建控制器*Users*
* 编辑app/controllers/*users*\_controller.rb文件，添加控制器逻辑如下：

class *Users*Controller < ApplicationController

def *new* # 根据ruby标准定义，*new*代表返回HTML输入表单用于创建新记录

@*user* = *User*.new

end

def *create* # 根据ruby标准定义，*create*代表创建新记录

@*user* = *User*.new(*user*\_params)

if @*user*.save

session[:*user*\_id] = @*user*.id #创建新会话

redirect\_to '/'

else

redirect\_to '/*signup*'

end

end

private

def *user*\_params

params.require(:*user*).permit(:*fname*,:*lname*,:*email*,:*password*)

end

end

* 输入rails generate controller *Sessions*创建控制器*Sessions*
* 编辑app/controllers/*sessions*\_controller.rb文件，添加控制器逻辑如下：

class *Sessions*Controller < ApplicationController

def *new* # 根据ruby标准定义，*new*代表返回HTML输入表单用于创建新记录

end

def *create* # 根据ruby标准定义，*create*代表创建新记录

@*user* = *User*.find\_by\_email(params[:*session*][:*email*])

if @*user* && @*user*.authenticate(params[:*session*][:*password*])

session[:*user*\_id] = @*user*.id #创建新会话

redirect\_to '/'

else

redirect\_to '*login*'

end

end

def *destroy* # 根据ruby标准定义，*destroy*代表删除记录

session[:*user*\_id] = nil #移除当前会话

redirect\_to '/'

end

end

**四、创建路由(Routes)**

* 编辑config/routes.rb文件，添加路由规则如下：

Rails.application.routes.draw do

root '*albums*#*index*'

get '*albums*' => '*albums*#*index*'

get '*albums*/*new*' => '*albums*#*new*'

get '*albums*/:id' => '*albums*#*show*', as: :*album*

#设置别名*album*，自动生成*album*\_path方法

post '*albums*' => '*albums*#*create*'

get '*signup*' => '*users*#*new*'

get '*login*' => '*sessions*#*new*'

post '*login*' => '*sessions*#*create*'

delete '*logout*' => '*sessions*#*destroy*'

resources :*users*

end

**五、创建视图(View)**

* 编辑app/views/layouts/application.html.erb网页模板文件，在body元素中添加显示逻辑如下：

<div class="header">

<div class="container">

<%= link\_to root\_path do %>

<img src="#" width="80" height="80">...

<% end %>

<div class="nav pull-right">

<% if *current\_user* %>

<!-- 若用户已登录，显示用户电邮地址及登出链接 -->

<ul>

<li><%= *current\_user*.*email* %></li>

<li>

<%= link\_to "*Logout*", *logout*\_path, method:"delete" %>

</li>

</ul>

<% else %>

<!-- 若用户尚未登陆，显示登陆链接和注册链接 -->

<ul>

<li><%= link\_to "*Login*", '*login*' %></li>

<li><%= link\_to "*Signup*", '*signup*' %></li>

</ul>

<% end %>

</div>

</div>

</div>

* 新建app/views/*albums*/*index*.html.erb网页模板文件，添加显示逻辑如下：

<div class="*albums*">

<div class="container">

<div class="row">

<% @*albums*.each do |*a*|%>

<div class="item col-md-4">

<div class="img-container">

<%= image\_tag *a*.*image* %>

</div>

<div class="*album*-info">

<h2><%= *a*.*title* %></h2>

<p>

<%= link\_to "...",*album*\_path(*a*) %>

<!-- 自动生成对应的链接 -->

</p>

</div>

</div>

<% end %>

</div>

</div>

</div>

* 新建app/views/*albums*/*show*.html.erb网页模板文件，添加显示逻辑如下：

<div class="*photos*">

<div class="container">

<h1><%= @*album*.*title* %></h1>

<div class="row">

<% @*photos*.each do |*p*|%>

<div class="item col-md-4">

<div class="img-container">

<%= image\_tag *p*.*image* %>

</div>

<div class="*photo*-info">

<p><%= *p*.*caption* %></p>

</div>

</div>

<% end %>

</div>

</div>

</div>

* 新建app/views/*users*/*new*.html.erb网页模板文件，添加显示逻辑如下：

<div class="*signup*">

<div class="container">

<div class="form">

<h1>...</h1>

<%= form\_for(@*user*) do |*f*| %>

<%= *f*.text\_field :*fname*, :placeholder => "*First Name*" %>

<%= *f*.text\_field :*lname*, :placeholder => "*Last Name*" %>

<%= *f*.email\_field :*email*, :placeholder => "*Email*" %>

<%= *f*.password\_field :*password*,

:placeholder => "*Password*" %>

<%= *f*.submit "*Signup*", class: "btn-submit" %>

<% end %>

</div>

</div>

</div>

* 新建app/views/*sessions*/*new*.html.erb网页模板文件，添加显示逻辑如下：

<div class="*login*">

<div class="container">

<div class="form">

<h1>...</h1>

<%= form\_for(:*session*, url: *login*\_path) do |*f*| %>

<%= *f*.email\_field :*email*, :placeholder => "*Email*" %>

<%= *f*.password\_field :*password*,

:placeholder => "*Password*" %>

<%= *f*.submit "*Login*", class: "btn-submit" %>

<% end %>

</div>

</div>

</div>

* 编辑app/assets/stylesheets/application.css样式表文件如下：

[见albumapp.css文件]

**六、调试**

* 输入rails server启动服务器
* 在浏览器地址栏输入 [http://localhost:3000/*signup*](http://localhost:3000/signup) 并注册新用户
* 输入rails console进入控制台
* 输入*User*.all查询数据库中已注册用户的记录信息
* 输入exit退出控制台
* 在浏览器地址栏输入 [http://localhost:3000/*login*](http://localhost:3000/login) 并登陆后查看效果
* 登出后在浏览器地址栏输入 [http://localhost:3000/*albums*](http://localhost:3000/albums) 查看效果