**用Ruby on Rails创建一对多关联模型项目**

**一、创建项目**

* 输入rails new *TravelApp*创建项目
* 输入cd *travelapp*进入项目根目录

**二、创建模型(Model)**

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

class *Tag* < ActiveRecord::Base

has\_many :*dests* #表示每个*tag*表可对应多个*dest*表

end

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

class Create*Tag*s < ActiveRecord::Migration

def change

create\_table :*tag*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 *Dest*创建第二个模型*Dest*
* 编辑app/models/*dest*.rb如下：

class *Dest* < ActiveRecord::Base

belongs\_to :*tag* #表示每个*dest*表属于单一*tag*表

end

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

class Create*Dest*s < ActiveRecord::Migration

def change

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

*t*.string :*name* #添加*name*字符串字段

*t*.string :*image* #添加*image*字符串字段

*t*.string :*desc*  #添加*desc*字符串字段

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

*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：

*t1* = *Tag*.create(*title*: "...", *image*: "#")

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t1*.id)

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t1*.id)

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t1*.id)

*t2* = *Tag*.create(*title*: "...", *image*: "#")

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t2*.id)

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t2*.id)

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t2*.id)

*t3* = *Tag*.create(*title*: "...", *image*: "#")

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t3*.id)

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t3*.id)

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t3*.id)

*t4* = *Tag*.create(*title*: "...", *image*: "#")

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t4*.id)

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t4*.id)

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t4*.id)

*t5* = *Tag*.create(*title*: "...", *image*: "#")

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t5*.id)

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t5*.id)

*Dest*.create(*name*: "...", *desc*: "...", *image*: "#", *tag*\_id: *t5*.id)

* 输入rake db:seed将样本数据添加到数据库

**三、创建控制器(Controller)**

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

class *Tags*Controller < ApplicationController

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

@*tags* = *Tag*.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

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

@*dests* = @*tag*.*dests* #查找@*tag*记录对应的所有*dest*数据并传给视图

end

end

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

class *Dests*Controller < ApplicationController

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

@*dest* = *Dest*.find(params[:id])

end

def *edit* # 根据ruby标准定义，*edit*代表返回HTML输入表单用于更新记录

@*dest* = *Dest*.find(params[:id])

end

def *update* # 根据ruby标准定义，*update*代表更新纪录

@*dest* = *Dest*.find(params[:id])

if @*dest*.*update*\_attributes(*dest*\_params) #记录已更新

redirect\_to(:action=>'*show*',:id=>@*dest*.id)

else #尚未更新

render '*edit*'

end

end

private

def *dest*\_params #定义仅供*update* action调用的私有方法

params.require(:*dest*).permit(:*name*, :*desc*)

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

get '*tags*' => '*tags*#*index*'

get '*tags*/:id' => '*tags*#*show*', as: :*tag*

#设置别名*tag*，自动生成*tag*\_path方法

get '*dests*/:id' => '*dests*#*show*', as: :*dest*

#设置别名*dest*，自动生成*dest*\_path方法

get '*dests*/:id/*edit*' => '*dests*#*edit*', as: :*edit\_dest*

#设置别名*edit\_dest*，自动生成*edit\_dest*\_path方法

patch '*dests*/:id' => '*dests*#*update*'

end

**五、创建视图(View)**

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

<div class="header">

<div class="container">

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

<h1>...</h1>

</div>

</div>

<div class="*tags*">

<div class="container">

<div class="cards row">

<% @*tags*.each do |*t*| %>

<div class="card col-xs-4">

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

<h2>

<%= *t*.*title* %>

</h2>

<%= link\_to "...", *tag*\_path(*t*) %> #自动生成对应的链接

</div>

<% end %>

</div>

</div>

</div>

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

<div class="header">

<div class="container">

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

<h1>...</h1>

</div>

</div>

<div class="*tag*">

<div class="container">

<h2 class="*tag*-*title*">

<%= @*tag*.*title* %>

</h2>

<div class="cards row">

<% @*dests*.each do |*d*| %>

<div class="card col-xs-4">

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

<h2>

<%= *d*.*name* %>

</h2>

<p>

<%= *d*.*desc* %>

</p>

<%= link\_to "...", *dest*\_path(*d*) %> #自动生成对应的链接

</div>

<% end %>

</div>

</div>

</div>

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

<div class="header">

<div class="container">

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

<h1>...</h1>

</div>

</div>

<div class="*dest*">

<div class="container">

<div class="row">

<div class="col-xs-12">

<%= image\_tag @*dest*.*image* %>

<h2>

<%= @*dest*.*name* %>

</h2>

<p>

<%= @*dest*.*desc* %>

</p>

<%= link\_to "...", *edit\_dest*\_path(@*dest*) %>

#自动生成对应的链接

</div>

</div>

</div>

</div>

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

<div class="header">

<div class="container">

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

<h1>...</h1>

</div>

</div>

<div class="*dest*">

<div class="container">

<%= image\_tag @*dest*.*image* %>

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

<div class="field">

<%= *f*.label :*dest* %><br />

<%= *f*.text\_area :*name* %><br />

<%= *f*.text\_area :*image* %><br />

<%= *f*.text\_area :*desc* %><br />

</div>

<div class="action">

<%= *f*.submit "*Edit*" %>

</div>

<% end %>

</div>

</div>

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

[见travelapp.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/*tags*](http://localhost:3000/tags) 查看效果