**用Ruby on Rails创建多对多关联模型项目**

**一、创建项目**

* 输入rails new *MovieApp*创建项目
* 输入cd *movieapp*进入项目子目录

**二、创建模型(Model)**

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

class *Movie* < ActiveRecord::Base

has\_many :*parts* #表示每个*movie*表可对应多个*part*表

has\_many :*actors*, through: :*parts*

#表示每个*movie*表可通过*part*表对应多个*actor*表

end

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

class Create*Movie*s < ActiveRecord::Migration

def change

create\_table :*movie*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*.string :*year* #添加*year*字符串字段

*t*.string :*plot* #添加*plot*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 *Actor*创建第二个模型*Actor*
* 编辑app/models/*actor*.rb如下：

class *Actor* < ActiveRecord::Base

has\_many :*parts* #表示每个*actor*表可对应多个*part*表

has\_many :*movies*, through: :*parts*

#表示每个*actor*表可通过*part*表对应多个*movie*表

end

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

class Create*Actor*s < ActiveRecord::Migration

def change

create\_table :*actor*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 :*image* #添加*image*字符串字段

*t*.string :*bio* #添加*bio*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 *Part*创建第三个模型*Part*
* 编辑app/models/*part*.rb如下：

class *Part* < ActiveRecord::Base

belongs\_to :*movie* #表示每个*part*表属于单一*movie*表

belongs\_to :*actor* #表示每个part表属于单一*actor*表

end

* 编辑db/migrate/*<timestamp3>*\_create\_*part*s.rb文件，其中*timestamp3*为创建*Part*模型的时间戳，添加字段如下：

class Create*Part*s < ActiveRecord::Migration

def change

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

*t*.belongs\_to :*movie*, index: true #添加*movie*\_id字段关联*movie*

t.belongs\_to :*actor*, index: true #添加*actor*\_id字段关联*actor*

*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：

*a*1 = *Actor*.create(*fname*: "…", *lname*: "…", *image*: "#", *bio*: "…")

*a*2 = *Actor*.create(*fname*: "…", *lname*: "…", *image*: "#", *bio*: "…")

...

*a*31 = *Actor*.create(*fname*: "…", *lname*: "…", *image*: "#", *bio*: "…")

*m*1 = *Movie*.create(*title*: "…", *image*: "#", *year*: "…", *plot*:"…")

*m*2 = *Movie*.create(*title*: "…", *image*: "#", *year*: "…", *plot*:"…")

...

*m*10 = *Movie*.create(*title*: "…", *image*: "#", *year*: "…", *plot*:"…")

*m*1.*actors* << [*a*1, *a*2, *a*3, *a*4, *a*5]

*m*2.*actors* << [*a*1, *a*6, *a*25]

...

*m*10.*actors* << [*a*31, *a*27, *a*28, *a*2]

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

class *Movies*Controller < ApplicationController

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

@*movies* = *Movie*.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

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

@*actors* = @*movie*.*actors*

#查找@*movie*记录对应的所有*actor*数据并传给视图

end

end

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

class *Actors*Controller < ApplicationController

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

@*actors* = *Actor*.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

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

@*movies* = @*actor*.*movies*

#查找@*actor*记录对应的所有*movie*数据并传给视图

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 '*movies*' => '*movies*#*index*'

get '*movies*/:id' => '*movies*#*show*', as: :*movie*

#设置别名*movie*，自动生成*movie*\_path方法

get '*actors*' => '*actors*#*index*'

get '*actors*/:id' => '*actors*#*show*', as: :*actor*

#设置别名*actor*，自动生成*actor*\_path方法

end

**五、创建视图(View)**

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

<div class="hero">

<div class="container">

<h2>…</h2>

<p>...</p>

<a href="#">...</a>

</div>

</div>

<div class="main">

<div class="container">

<h2>...</h2>

<% @*movies*.each do |*m*| %>

<div class="*movie*">

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

<h3><%= *m*.*title* %></h3>

<p><%= *m*.*year* %></p>

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

</div>

<% end %>

</div>

</div>

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

<div class="main *movie*-*show*">

<div class="container">

<div class="*movie*">

<div class="info">

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

<h3 class="*movie*-*title*"><%= @*movie*.*title* %></h3>

<p class="*movie*-*year*"><%= @*movie*.*year* %></p>

<p class="*movie*-*plot*"><%= @*movie*.*plot* %></p>

</div>

</div>

<h2>...</h2>

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

<div class="*actor*">

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

<h3 class="*actor*-*name*">

<%= *a*.*fname* + " " + *a*.*lname* %>

</h3>

<p class="*actor*-*bio*">

<%= *a*.*bio* %>

</p>

</div>

<% end %>

</div>

</div>

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

<div class="main *actor*-*index*">

<div class="container">

<div class="row">

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

<div class="*actor* col-xs-2">

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

<h3><%= *a*.*fname* + " " + *a*.*lname* %></h3>

<p><%= *a*.*bio* %></p>

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

</div>

<% end %>

</div>

</div>

</div>

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

<div class="main *actor*-*show*">

<div class="container">

<div class="*actor*">

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

<div class="info">

<h3 class="*actor*-*name*">

<%= @*actor*.*fname* + " " + @*actor*.*lname* %>

</h3>

<p class="*actor*-*bio*"><%= @*actor*.*bio* %></p>

</div>

</div>

<h2>...</h2>

<% @*movies*.each do |*m*| %>

<div class="*movie*">

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

<h3 class="*movie*-*title*"><%= *m*.*title* %></h3>

<p class="*movie*-*year*"><%= *m*.*year* %></p>

<p class="*movie*-*plot*"><%= *m*.*plot* %></p>

</div>

<% end %>

</div>

</div>

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

[见movieapp.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/movies> 查看效果
* 在浏览器地址栏输入 <http://localhost:3000/actors> 查看效果