**用Ruby on Rails创建单模型项目**

**一、创建项目**

* 输入rails new *MessengerApp*创建项目
* 输入cd *messengerapp*进入项目根目录

**二、创建模型(Model)**

* 输入rails generate model *Message*创建模型*Message*
* 编辑db/migrate/*<timestamp>*\_create\_*message*s.rb文件，其中*timestamp*为创建模型的时间戳，添加字段如下：

class Create*Message*s < ActiveRecord::Migration

def change

create\_table :*message*s do |*t*|

*t*.text :*content* #添加文本字段，其中*content*为字段名称

*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：

m1 = *Message*.create(*content*: "...")

m2 = *Message*.create(*content*: "...")

...

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

class *Messages*Controller < ApplicationController

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

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

end

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

@*message* = *Message*.new #将模型从输入表单接收的新数据传给视图

end

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

@*message* = *Message*.new(*message*\_params)

if @*message*.save #记录已创建

redirect\_to '/*messages*'

else #尚未创建

render '*new*'

end

end

private

def *message*\_params # 定义仅供create action调用的私有方法

params.require(:*message*).permit(:*content*)

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 '*messages*' => '*messages*#*index*' #路由规则格式为：

get '*messages*/*new*' => '*messages*#*new*' #<请求方法> '<路径>' =>

post '*messages*' => '*messages*#*create*' #'<控制器名>#<action名>'

end

**五、创建视图(View)**

* 新建app/views/*messages*/*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="#">

<h1>...</h1>

</div>

</div>

<div class="*messages*">

<div class="container">

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

<div class="*message*">

<p class="*content*"><%= *m*.*content* %></p>

<p class="time"><%= *m*.created\_at %></p>

</div>

<% end %>

<%= link\_to "*New*", "*messages*/*new*" %>

</div>

</div>

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

<div class="header">

<div class="container">

<img src="#">

<h1>...</h1>

</div>

</div>

<div class="*create*">

<div class="container">

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

<div class="field">

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

<%= *f*.text\_area :*content* %>

</div>

<div class="actions">

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

</div>

<% end %>

</div>

</div>

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

[见messengerapp.scss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/*messages*](http://localhost:3000/messages) 查看效果