

RoR Week 2 (6/17 & 6/19)

Breath and Ride the Rails

The Project App

PHASE I

Start a new project called ewb_projects

```
>>rails new ewb_projects
```

Generate a welcome controller

```
>>rails generate controller welcome index
```

Generate a products controller with the 7 RESTful functions & explore routing

```
>>rails generate controller products index create destroy edit update new show
```

Delete the products controller

```
>>rails destroy controller products
```

Create a “Project” model

```
>>rails generate model Project
```

Take a look at the migration file and add a title or type string and description of type text. OR you can use the migration command

```
>>rails generate migration AddTitleToProject title:string
```

```
>>rails generate migration AddDescriptionToProject description:text
```

Or modify the migration file directly with

```
  t.string :title
```

```
  t.text :description
```

Play with it in console

```
>>rails console
```

Create a few projects

```
lrb>>Project.all
```

```
lrb>>a = Project.new
```

```
lrb>>a.title = “Dominican Republic Sustainable Infrastructure”
```

```
lrb>>a.description = “The international project team is traveling to the DR this July to ....tbc”
```

```
lrb>>a.save
```

Exit the console

```
>>exit
```

Now we have these projects in our database. Lets get them rendered on our website!

Create the controller

```
>>rails g controller projects
```

Modify your routes file such that the Article resource is there now

```
resources :projects
```

Add an index action to your controller such that you can access your model with a variable from your view

```
def index
  @projects = Project.all
end
```

Now go to your view and print the project titles in your database

```
<p>This is a list of EWB Projects</p>
<ul>
  <% @projects.each do |proj| %>
    <li><%= link_to proj.title, proj_path(proj)%></li>
  <% end %>
</ul>
```

Add a show action to your controller such that you can access your model with a variable from your view

```
def show
  @project = Project.find(params[:id])
end
```

Now go to your view and print the article titles in your database

```
<p>This page shows the project in more detail</p>
<p><%= @project.body %></p>
<%= link_to "<<Back to all proejcts", projects_path%>
```

STOP: Now you should understand...

Basic routing with resources, RESTful routing, how to link to routes, how to pass params through your controllers.

Extra Challenges for practice if you want

- 1) Edit the routes file so that the root path points to your list of articles
- 2) Add an image to your database that shows up when you display the project description
- 3) Stylize your page

Delete the project and do it again

```
rm -r ewb_projects
```

-----PHASE II-----

Picking up where you left of from Phase I, we will now allow users to create, edit and destroy projects and later allow only privileged users to create, edit and destroy articles.

Following RESTful conventions we will write a form in the view associated with the projects#new action. This uses the form helper form_for, .submit, etc

```
<%= form_for(@project) do |p| %>
  <ul> <% @project.errors.full_messages.each do |error| %>
    <li><%= error %></li>
  <% end %>
</ul>
```

```
<p>
  <%= p.label :title %><br />
  <%= p.text_field :title %>
</p>
```

```
<p>
  <%= p.label :description %><br />
  <%= p.text_area :description %>
</p>
```

```
<p>
  <%= p.submit %>
</p>
```

```
<% end %>
```

Now we have to make this @project variable accessible to the view via our controller

Add @article = Article.new to the appropriate event in your controller

We also need the create action to process the “new” form. Check out the params available using “fail”

```
def create
  fail
end
```

Now change it to save what the user entered

```
def create
  @project = Project.new
  @project.title = params[:project][:title]
  @project.description = params[:project][:description]
  @project.save
  redirect_to project_path(@project)
end
```

Realistically we don’t want to blindly pass params like this into our database so we use what are called “Strong Parameters” via “require” and “permit”. Permit returns the hash while require returns the hash.

Your projects helper should look like this:

```
def project_params
  params.require(:project).permit(:title, :description)
end
```

Your projects controller should now look like this:

```
def create
  @project = Project.new(project_params)
  @project.save
end
```

```
        redirect_to project_path(@project)
    end
```

PS—don't forget to include the projects helper in your controller

What if we want to delete a project? Add this link to the show view

```
>> <%=link_to "Delete Project", project_path(@project), method: :delete, data: {confirm: "Really delete the project?"}%></br>
```

But now we have to actually delete it via the controller

```
def destroy
    @project = Project.find(params[:id])
    @project.destroy
    redirect_to projects_path
end
```

Test your modification by creating a new article. It should work just the same

Then lets add to our update action in the controller. Add:

```
@project = Project.find(params[:id])
@project.update(project_params)
redirect_to project_path(@project)
```

We can add a flash notice if we want.

>>you figure it out

Now replace the form we made in the “new” and “update” view with the line below and put the form in a partial called ‘_form.html.erb’

```
>><%= render partial 'form' %>
```

****Challenge: add all your flash notices to your layouts file**

****Challenge: add images to your articles with paperclip gem**

>>you figure it out or ask me next class

STOP: YOU SHOULD NOW ALL UNDERSTAND

- 1) How what the create, new, edit, destroy RESTful actions do and how to implement them
- 2) Know how to build a form using the “form_for” helper
- 3) Use flash notices
- 4) Use the link_to, redirect commands
- 5) Be able to rout using prefixes from rake routes
- 6) Be able to add and manipulate objects in the irb console
- 7) Know what form helpers are for and be able to use them in your views
- 8) Understand and be able to add to and edit migrations