

Coding for Development and Social Change, Oct 2014

Ideas Into Code

Turning all that theory into
your own applications

Let's create a Rails app!

- ❖ Specifically, the RailsGirls Diary app...

RG: First, create index.html

Create index.html, with these contents:

```
<!doctype html>
<html>
  <head>
    <title>My Little Webapp: Coding Is Magic</title>
    <meta charset="UTF-8" />
    <link rel="stylesheet" href="https://raw.githubusercontent.com/krzysztofbialek/Rails-
Girls-Warsaw-App/master/style.css" />
  </head>
  <body>
  </body>
</html>
```

RG: add content to index.html

Add the following HTML between the <body> and </body> tags

```
<h1>My Diary</h1>
  <div>
    <h2>Submitted a Rails Girls application</h2>
    <p>1.02.2014</p>
    <p>Just submitted an application to a Rails Girls workshop. Can't wait
to see whether I'll get in!</p>
    <h2>Got in!</h2>
    <p>15.02.2014</p>
    <p>Received an email that my application got accepted! I'll be at a RG
workshop next week!</p>
    <h2>The first day starts...</h2>
    <p>22.02.2014</p>
    <p>Today is the first day of the Rails Girls workshop. My coach is
quite strange but it seems we all have Rails installed now and can start
learning.</p>
  </div>
```

RG: add links into index.html

Add the following before the above diary entries (again, do adjust to taste):

```
<div>
  <h1>My favourite websites</h1>
  <ul>
    <li><a href="http://railsgirls.com">Rails Girls</a></li>
    <li><a href="https://en.wikibooks.org/wiki/
Ruby_Programming">Wikibooks</a></li>
    <li><a href="http://guides.rubyonrails.org">Ruby on Rails Guides</a></
li>
  </ul>
</div>

```

RG: Create a Rails application

- ❖ In the terminal window:
 - ❖ Go to the directory you want your app in
 - ❖ Type this:
rails new diary
cd diary
rails server
- ❖ In your browser, go to <http://localhost:3000/>
- ❖ Press control-c in the terminal to stop your app.

What just happened?

- ❖ Rails created a folder, diary, and added files to it
- ❖ You used “rails server” to start your app
- ❖ You saw the default homepage for a rails app

The new folder

- Gemfile - libraries
- Gemfile.lock - library versions
- README.rdoc - your description of the application
- app/ - where you put your Ruby code
- config/ - application configuration
- db/ - database
- public/ - all webpages that *don't* get generated by Ruby code

The app folder

- **models**
- **views**
- **controllers**
- **mailers**
- **helpers**
- **assets**

RG: Create the Homepage

rails generate controller welcome index

go to <http://localhost:3000/welcome/index>

In the terminal, type "rake routes"

Edit the `config/routes.rb` file and uncomment (remove the `#` from the front) the root `'welcome#index'` line

Edit the `app/views/welcome/index.html.erb` file and copy the contents of the `<body>` tags from your original `index.html` file

Refresh your browser

Rake: <http://www.tutorialspoint.com/ruby-on-rails/rails-and-rake.htm>

RG: Put websites list in controller

In `app/views/welcome/index.html.erb`, replace the `` contents with:

```
<%  
  @websites = [  
    ["http://railsgirls.com", "Rails Girls"],  
    ["https://en.wikibooks.org/wiki/Ruby_Programming", "Wikibooks"],  
    ["http://guides.rubyonrails.org", "Ruby on Rails Guides"],  
  ]  
%>  
<% for url, name in @websites %>  
  <li><a href="<%= url %>"><%= name %></a></li>  
<% end %>
```

Refresh your browser

RG: Put websites list in controller

Now move the @websites definition into app/controllers/welcome_controller.rb:

```
class WelcomeController < ApplicationController
  def index
    @websites = [
      ["http://railsgirls.com", "Rails Girls"],
      ["https://en.wikibooks.org/wiki/Ruby_Programming", "Wikibooks"],
      ["http://guides.rubyonrails.org", "Ruby on Rails Guides"],
    ]
  end
end
```

And delete this code from app/views/welcome/index.html.erb

Refresh your browser

RG: Create the Entry Model

Create the Entry model, with attributes title, date and contents. In the terminal window:

```
rails generate model Entry title:string date:date contents:text
```

```
rake db:migrate
```

RG: Add data to the Entry model

In the terminal window, type:

rails console

This starts rails on the command line. In there, type:

```
Entry.create "title" => "First day of class", "date" => Date.new(2014, 11, 14), "contents" => "Started coding for development and social change today"
```

```
Entry.create "title" => "Second day of class", "date" => Date.new(2014, 11, 15), "contents" => "My professor is quite strange but it seems we all have Rails installed now and can start building."
```

```
Entry.create "title" => "Third day of class", "date" => Date.new(2014, 11, 16), "contents" => "Yay! Free from the post-its!"
```

Entry.all

exit

RG: Connect the Entry view to the Entry Model

Add this to `app/controllers/welcome_controller.rb`

```
@entries = Entry.all
```

Change the diary entry lines in `app/views/welcome/index.html.erb` (the bit after `<h1>My Diary</h1>`) to:

```
<% for entry in @entries %>  
  <h2><%= entry.title %></h2>  
  <p><%= entry.date %></p>  
  <p><%= entry.contents %></p>  
<% end %>
```

You now have the Entry model and view and can view the list of entries from the welcome page. But you can't view, edit or create a new diary entry yet. Let's fix that.

Create the Entry controller

In the terminal window:

```
rails generate controller Entries
```

This created the EntriesController class. Then type

```
rake routes
```

Edit config/routes.rb and add this line after root 'welcome#index':

```
resources "entries"
```

Then type “**rake routes**” in the terminal window again.

Create the Entry index controller

- ❖ Go to <http://localhost:3000/entries>
- ❖ Open `app/controllers/entries_controller.rb` and add
- ❖ **`def index`**
- ❖ **`end`**
- ❖ Refresh your browser.
- ❖ Now edit `app/controllers/entries_controller.rb` so it looks like this:
- ❖ **`def index`**
- ❖ **`@entries = Entry.all`**
- ❖ **`end`**

Create the Entry index view

Open `app / views / entries / index.html.erb` and add this (you've already seen this in `welcome.html.erb`):

```
<% for entry in @entries %>
```

```
  <h2><%= entry.title %></h2>
```

```
  <p><%= entry.date %></p>
```

```
  <p><%= entry.contents %></p>
```

```
<% end %>
```

Refresh your page

Create an Entry view

Go to <http://localhost:3000/entries/1>

Add this to `app/controllers/entries_controller.rb`:

```
def show
```

```
  @entry = Entry.find(params["id"])
```

```
end
```

Edit `app/views/entries/show.html.erb` and add this to it:

```
<h2><%= @entry.title %></h2>
```

```
<p><%= @entry.date %></p>
```

```
<p><%= @entry.contents %></p>
```

Refresh your browser. Try going to <http://localhost:3000/entries/2>

Add links to individual entries to the Entries list

in the terminal, type “rake routes”

Edit `app/views/entries/index.html.erb`. Change the line with the entry title in it to

```
<h2><%= link_to(entry.title, entry_path(entry)) %></h2>
```

And add a way to get back to the list of entries: in file `app/views/entries/show.html.erb` add this line:

```
<p><%= link_to("Back to all entries", entries_path) %></p>
```

Add the “New Entry” view

Add this to `app/views/entries/index.html.erb`:

```
<%= link_to("New entry", new_entry_path) %>
```

Create `app/views/entries/new.html.erb` and add this:

```
<%= form_for(Entry.new) do |form| %>
```

```
  <p><%= form.label("title") %></p>
```

```
  <p><%= form.text_field("title") %></p>
```

```
  <p><%= form.label("contents") %></p>
```

```
  <p><%= form.text_area("contents") %></p>
```

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

```
<% end %>
```

```
<p><%= link_to("Back to all entries", entries_path) %></p>
```

Add the “New Entry” controller

Add this to app/controllers/entries_controller.erb:

```
def create  
  
  entry_params = params["entry"]  
  
  entry = Entry.create(entry_params)  
  
  redirect_to(entry_path(entry))  
  
end
```

<http://localhost:3000/entries/new> will fail when you hit “create entry” - this is a security thing.
Change the entry_params line above to:

```
entry_params = params["entry"].permit("title", "contents")
```

And try creating a new entry again

Add the “Edit Entry” controller

Add this link to `app/views/entries/show.html.erb`

```
<p><%= link_to("Edit this entry", edit_entry_path(@entry)) %></p>
```

Add an edit controller to `app/controllers/entries_controller.erb`:

```
def edit
```

```
  @entry = Entry.find(params["id"])
```

```
end
```

Add the “Edit Entry” view

Create `app/views/entries/edit.html.erb` and add this code to it:

```
<%= form_for(@entry) do |form| %>

  <p><%= form.label("title") %></p>

  <p><%= form.text_field("title") %></p>

  <p><%= form.label("contents") %></p>

  <p><%= form.text_area("contents") %></p>

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

<% end %>

<p><%= link_to("Back to this entry", entry_path(@entry)) %></p>

<p><%= link_to("Back to all entries", entries_path) %></p>
```

And add the controller for updated entries

Try hitting the edit button. Fails? You need to add the update function to the controller `app/controllers/entries_controller.erb...`

```
def update
```

```
  entry_params = params["entry"].permit("title", "contents")
```

```
  entry = Entry.find(params["id"])
```

```
  entry.update(entry_params)
```

```
  redirect_to(entry_path(entry))
```

```
end
```

More resources

- ❖ http://guides.rubyonrails.org/getting_started.html
- ❖ <http://railsforzombies.org/>
- ❖ TeamTreehouse