

Ingeniería de Software

Rails MVC

Juan Pablo Sandoval

Embedded Ruby -ERB

Que es *ERB*?

- *Un archivo ERB termina con al extensión “.html.erb”*
- *Es una mezcla en HTML y Ruby.*
- *En resumen dentro del un archivo ERB todo es html con excepción de lo que esta entre `<%= ... %>`*

Ejemplo 1

```
<h1>  
    Hello <%= name %>!  
</h1>
```

*Si la variable **name** es igual a “Juan Pablo”, el código anterior generará:*

```
<h1>  
    Hello Juan Pablo!  
</h1>
```

Ejemplo 2

```
<% if @favorite_food == "chocolate" %>  
    Are you a chocolate lover? Here are some of  
    our best PREMIUM chocolate bars!  
<% else %>  
    Here are our top 10 snacks that people  
    bought this month.  
<% end %>
```

*ERB no generará la sección de código que no necesita, en este ejemplo solo generará el cuerpo del **if** o del **else**.*

Ejemplo 3

```
<% @books.each do |book| %>
  <%= book.title %>
  <%= book.author %>
  <br>
<% end %>
```

*ERB generará un html con la información de todos los libros agregando el tag **
** después de cada libro.*

ERB + Rails

ERB + Rails

```
class BooksController < ApplicationController
  def index
    @books = Book.all
  end
end
```

En un controlador si nosotros no utilizamos la función render para devolver un HTTPResponse de forma “explicita”, Rails buscara un archivo ERB con el mismo nombre de la acción, generará el HTML y enviara el HTTPResponse.

*/app/views/books/**index.html.erb***

ERB + Rails

*Ojo el nombre del **controlador** y el **método** debe mapear de forma exacta con las dirección y el nombre del archivo.*

*/app/views/books/**index.html.erb***

index.html.erb

*La variable @books esta definida e inicializada en el método **index** dentro del controlador **BooksController**.*

```
<h1>Listing Books</h1>
<table>
  <thead>
    <tr>
      <th>Title</th>
      <th>Content</th>
    </tr>
  </thead>
  <tbody>
    <% @books.each do |book| %>
      <tr>
        <td><%= book.title %></td>
        <td><%= book.content %></td>
      </tr>
    <% end %>
  </tbody>
</table>
<br>
```

Rails Form Helpers

```
<%= form_with url: "/search", method: :get do |form| %>
  <%= form.label :query, "Search for:" %>
  <%= form.text_field :query %>
  <%= form.submit "Search" %>
<% end %>
```

```
<form action="/search" method="get" accept-charset="UTF-8" >
  <label for="query">Search for:</label>
  <input id="query" name="query" type="text" />
  <input name="commit" type="submit"
    value="Search" data-disable-with="Search" />
</form>
```

Rails Form Helpers - Checkbox

```
<%= form.check_box :pet_dog %>  
<%= form.label :pet_dog, "I own a dog" %>  
<%= form.check_box :pet_cat %>  
<%= form.label :pet_cat, "I own a cat" %>
```

```
<input type="checkbox" id="pet_dog" name="pet_dog" value="1" />  
<label for="pet_dog">I own a dog</label>  
<input type="checkbox" id="pet_cat" name="pet_cat" value="1" />  
<label for="pet_cat">I own a cat</label>
```

Rails Form Helpers - Otros ejemplos

```
<%= form.text_area :message, size: "70x5" %>
<%= form.hidden_field :parent_id, value: "foo" %>
<%= form.password_field :password %>
<%= form.number_field :price, in: 1.0..20.0, step: 0.5 %>
<%= form.range_field :discount, in: 1..100 %>
<%= form.date_field :born_on %>
<%= form.time_field :started_at %>
<%= form.datetime_local_field :graduation_day %>
<%= form.month_field :birthday_month %>
<%= form.week_field :birthday_week %>
<%= form.search_field :name %>
<%= form.email_field :address %>
<%= form.telephone_field :phone %>
<%= form.url_field :homepage %>
<%= form.color_field :favorite_color %>
```

Path/URL Helpers - Rutas

```
Rails.application.routes.draw do
  resources :photos
end
```

HTTP Verb	Path	Controller#Action	Used for
GET	/photos	photos#index	display a list of all photos
GET	/photos/new	photos#new	return an HTML form for creating a new photo
POST	/photos	photos#create	create a new photo
GET	/photos/:id	photos#show	display a specific photo
GET	/photos/:id/edit	photos#edit	return an HTML form for editing a photo
PATCH/PUT	/photos/:id	photos#update	update a specific photo
DELETE	/photos/:id	photos#destroy	delete a specific photo

Path/URL Helpers

`photos_path -> te genera el string "/photos"`

`new_photo_path -> te genera el string "/photos/new"`

`edit_photo_path(:id) -> te genera el string "/photos/:id/edit"`

`photo_path(id) -> te genera el string "/photos/:id"`

render:

Using render:

```
def update
  @book = Book.find(params[:id])
  if @book.update(book_params)
    redirect_to(@book)
  else
    render "edit"
  end
end
```

Si quieres renderizar otra vista de nombre diferente al nombre del método, puedes utilizar el método render, en el ejemplo anterior dependiendo a la condición se renderiza la vista "edit.html.erb"

Using render:

```
render "products/show"
```

Otras formas de utilizar render

```
render :edit  
render action: :edit  
render "edit"  
render action: "edit"  
render "books/edit"  
render template: "books/edit"
```

Todas son equivalentes.

Using redirect_to:

```
def index  
  @books = Book.all  
end
```

```
def show  
  @book = Book.find_by(id: params[:id])  
  if @book.nil?  
    redirect_to action: :index  
  end  
end
```

redirect_to no solo renderiza la vista “index.html.erb”, ya que para renderizar la vista “index” es probable que necesite el atributo @books. En realidad *redirect_to*: hace un refresh del browser en la pagina de “index”, por lo que básicamente se enviará un nuevo HTTPRequest.

Mas sobre ERB, Helpers, Render y Layout?

- https://guides.rubyonrails.org/form_helpers.html
- https://guides.rubyonrails.org/layouts_and_rendering.html
- <https://guides.rubyonrails.org/routing.html>

Ejemplo - BookStore