

ACTIVIDAD
AJAX: Asynchronous JavaScript y XML

Cesar Jesus Lara Avila

Calagua Mallqui Jairo Andre

20210279F

Facultad de Ciencias, Universidad Nacional de Ingeniería

noviembre del 2023

PARTE 1

Se nos pide modificar la acción “show”:

```
Semana8 > myrottenpotatoes > app > controllers > movies_controller.rb
1  class MoviesController < ApplicationController
2    def index
3      @movies = Movie.all
4    end
5    def show
6      id = params[:id] # retrieve movie ID from URI route
7      @movie = Movie.find(id) # look up movie by unique ID
8      # will render render app/views/movies/show.html.haml by default
9      respond_to |format|
10     format.html do
11     end
12     format.js do
13       # Si es una solicitud AJAX, renderiza la vista parcial 'movie'
14       render partial: 'movie', object: @movie
15     end
16   end
17 end
18
```

Esta modificación hace uso de ‘respond_to’ para manejar diferentes tipos de solicitudes. Cuando se recibe una solicitud HTML normal, seguirá funcionando como antes, renderizando la vista completa.

PARTE 2:

El código JavaScript proporcionado establece un objeto ‘MoviePopup’ con tres funciones principales: ‘setup’, ‘getMovieInfo’, y ‘showMovieInfo’. La función ‘setup’ inicializa un div oculto (‘popupDiv’) al final del cuerpo de la página y adjunta un evento de clic a los enlaces dentro del elemento con id ‘movies’. Este evento de clic invoca ‘MoviePopup.getMovieInfo’. La función ‘getMovieInfo’ realiza una solicitud AJAX utilizando ‘\$.ajax’ al URL proporcionado por el enlace y muestra la información de la película llamando a ‘MoviePopup.showMovieInfo’ en caso de éxito. La función ‘showMovieInfo’ utiliza los datos recibidos para mostrar la información de la película en un div flotante.

Me piden agregar el código proporcionado a la parte del código CSS, así:

```
80  #movieInfo {
81      padding: 2ex;
82      position: absolute;
83      border: 2px double grey;
84      background: wheat;
85  }
```

Y los resultados que tengo son los siguiente:

← → ↻ ⓘ 127.0.0.1:3000/movies ☆ □ 🔍

All Movies

Add Movie

Movie Title	Rating	Release Date
Star Wars	PG	1977-04-25
Aladdin	G	1992-11-25
When Harry Met Sally	R	1989-07-21
The Help	PG-13	2011-08-10
Raiders of the Lost Ark	PG	1981-06-12
Aladdin	G	1992-11-25
When Harry Met Sally	R	1989-07-21
The Help	PG-13	2011-08-10
Raiders of the Lost Ark	PG	1981-06-12
Star Wars	PG	1977-04-25
Requiem for a Dream	R	2000-10-27
Aladdin	G	1992-11-25
When Harry Met Sally	R	1989-07-21
The Help	PG-13	2011-08-10
Raiders of the Lost Ark	PG	1981-06-12
Aladdin	G	1992-11-25
When Harry Met Sally	R	1989-07-21
The Help	PG-13	2011-08-10
Raiders of the Lost Ark	PG	1981-06-12
Aladdin	G	1992-11-25
When Harry Met Sally	R	1989-07-21
The Help	PG-13	2011-08-10
Raiders of the Lost Ark	PG	1981-06-12
T	G	2023-10-31

Para esto fue necesario realizar una migración en nuestro proyecto, luego definir la estructura de la tabla añadiendo campos como 'username', 'email'.

Una vez realizado esto ejecutamos la migración con "rails db:migrate", esto ejecutará la migración y creará la tabla 'moviegoers' en nuestra base de datos.

```
Semana8 > myrottenpotatoes > db > migrate > 20231123140500_create_moviegoers

1  class CreateMoviegoers < ActiveRecord::Migration[7.0]
2    def change
3      create_table :moviegoers do |t|
4        t.string :username
5        t.string :title
6        t.string :email
7        t.timestamps
8      end
9    end
10  end
```

PARTE 3:

Cuando se crean nuevos elementos dinámicamente en tiempo de ejecución y se necesita asociarles eventos, jQuery proporciona el método on con una sintaxis especial para delegación de eventos. En lugar de asignar el manejador directamente a los elementos, se asigna al elemento principal y se especifica el selector de los elementos secundarios que pueden ser agregados dinámicamente más adelante. Por ejemplo:

```
11  $('#movies').on('click', '.myClass', function() {
12    // Nuestra lógica aquí
13  });
```

Ahora, en el botón Add Movie, nos aparecía un error que se debía a que no habíamos creado un archivo .erb. El método 'new' dentro de un controlador de recursos suele utilizarse para mostrar un formulario nuevo.

Entonces creamos "new.html.erb" y gregamos lo siguiente:

```

Semana8 > myrottenpotatoes > app > views > movies > new.html.erb
1  <!-- app/views/movies/new.html.erb -->
2  <h1>New Movie</h1>
3  <%= form_with(model: @movie, local: true) do |form| %>
4    <div>
5      <%= form.label :title %>
6      <%= form.text_field :title %>
7    </div>
8
9    <div>
10     <%= form.label :rating %>
11     <%= form.text_field :rating %>
12   </div>
13
14   <div>
15     <%= form.label :release_date %>
16     <%= form.date_field :release_date %>
17   </div>
18
19   <div>
20     <%= form.submit %>
21   </div>
22 <% end %>

```

Por lo que ahora, en nuestro navegador ya se nos mostraba la pestaña con casillas para poder editar nuestras películas, pero si editamos ello, nos aparece el siguiente error:

NameError in Movies#index

Showing C:/Users/calag/Desktop/Desarrollo-software-2023/Semana8/myrottenpotatoes/app/views/layouts/application.html.erb where line #12 raised:

undefined local variable or method `movie' for #<ActionView::Base:0x0000000000bfb8>
Did you mean? @movies

Extracted source (around line #12):

```

10   <div class="container">
11     <%- if flash[:notice] %>
12     <div class="row"<%= (' adult' unless movie.rating =~ /^G|PG$/ ) %>>
13     <%- elsif flash[:alert] %>
14     <div class="alert alert-danger text-center"><%=flash[:alert]%></div>
15   <% end %>

```

Rails.root: C:/Users/calag/Desktop/Desarrollo-software-2023/Semana8/myrottenpotatoes

[Application Trace](#) | [Framework Trace](#) | [Full Trace](#)
app/views/layouts/application.html.erb:12

Este error está relacionado con el uso de una variable 'movie' que no está definida en la plantilla 'application.html.erb'. La parte del código que causa el problema parece estar intentado acceder a 'movie.rating' sin haber definido 'movie', entonces solamente realizo el siguiente cambio al código:

```
9 <body>
10 <div class="container">
11 <%= if flash[:notice] %>
12 <div class="row"<%= (' adult' unless @movies.first.rating =~ /^G|PG$/)> %>
13 <%= elsif flash[:alert] %>
14 <div class="alert alert-danger text-center"><%=flash[:alert]%></div>
```

Y al volver a desplegar nuestro trabajo ya podemos ver que se ha solucionado el problema y que incluso puede aparecer la película que hemos agregado:



Movie Title	Rating	Release Date
Star Wars	PG	1977-04-25
Aladdin	G	1992-11-25
When Harry Met Sally	R	1989-07-21
The Help	PG-13	2011-08-10
Raiders of the Lost Ark	PG	1981-06-12
Aladdin	G	1992-11-25
When Harry Met Sally	R	1989-07-21
The Help	PG-13	2011-08-10
Raiders of the Lost Ark	PG	1981-06-12
Star Wars	PG	1977-04-25
Requiem for a Dream	R	2000-10-27
Aladdin	G	1992-11-25
When Harry Met Sally	R	1989-07-21
The Help	PG-13	2011-08-10
Raiders of the Lost Ark	PG	1981-06-12
Aladdin	G	1992-11-25
When Harry Met Sally	R	1989-07-21
The Help	PG-13	2011-08-10
Raiders of the Lost Ark	PG	1981-06-12
Aladdin	G	1992-11-25
When Harry Met Sally	R	1989-07-21
The Help	PG-13	2011-08-10
Raiders of the Lost Ark	PG	1981-06-12
T	G	2023-10-31
Ed	G	2023-10-31
The Roundup	G	2023-11-05
La Princesa Y El Sapo	PG	2023-11-23

Finalmente, si quiero editar una película me aparece el siguiente error:



← → ↻ ⓘ 127.0.0.1:3000/movies/4/edit

No template for interactive request

MoviesController#edit is missing a template for request formats: text/html

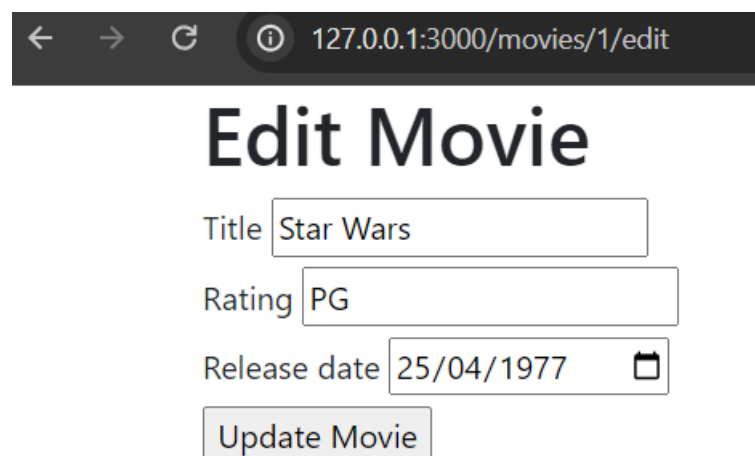
NOTE!
Unless told otherwise, Rails expects an action to render a template with the same name, contained in a folder named after its controller. If this controller is an API responding with 204 (No Content), which does not require a template, then this error will occur when trying to access it via browser, since we expect an HTML template to be rendered for such requests. If that's the case, carry on.

Este error nos está indicando que falta una plantilla para la acción 'edit' del controlador 'MoviesController'. Cuando se hace clic en el botón "Edit this movie", Rails espera una plantilla HTML correspondiente para mostrar el formulario de edición de esa película en particular.

Entonces debemos crear una vista para la acción 'edit' dentro de la carpeta movies, así:

```
application.html.erb M  edit.html.erb U X  application.css M
Semana8 > myrottenpotatoes > app > views > movies > edit.html.erb
1  <h1>Edit Movie</h1>
2  <%= form_for @movie do |f| %>
3      <!-- Aquí coloca los campos que quieras editar -->
4      <div class="field">
5          <%= f.label :title %>
6          <%= f.text_field :title %>
7      </div>
8
9      <div class="field">
10         <%= f.label :rating %>
11         <%= f.text_field :rating %>
12     </div>
13
14     <div class="field">
15         <%= f.label :release_date %>
16         <%= f.date_field :release_date %>
17     </div>
18
19     <div class="actions">
20         <%= f.submit 'Update Movie' %>
21     </div>
22 <% end %>
```

Y ahora sí, ya podemos editar las películas que queremos:




← → ↻ 127.0.0.1:3000/movies/1/edit

Edit Movie

Title

Rating

Release date 

Pero al editar, nos aparece lo siguiente:

NoMethodError in MoviesController#update

undefined method `update_attributes' for #<Movie id: 1, title: "Star Wars", rating: "PG", description: nil, release_date: "1977-04-25 00:00:00.000000000 +0000", created_at: "2023-10-04 16:37:50.689777000 +0000", updated_at: "2023-10-04 16:37:50.689777000 +0000">

Did you mean?

update_attribute

Extracted source (around line #33):

```
31 def update
32   @movie = Movie.find params[:id]
33   if (@movie.update_attributes(movie_params))
34     redirect_to movie_path(@movie), :notice => "#{@movie.title} updated."
35   else
36     flash[:alert] = "#{@movie.title} could not be updated: " +
```

Rails.root: C:/Users/calag/Desktop/Desarrollo-software-2023/Semana8/myrottenpotatoes

[Application Trace](#) | [Framework Trace](#) | [Full Trace](#)

app/controllers/movies_controller.rb:33:in `update'

Request

Parameters:

Este mensaje de error indica que el método 'update_attributes' no está definido para el objeto '@movie'. En su lugar, debemos utilizar 'update' en el controlador 'MoviesController' para actualizar los atributos de la película, así:

```
31 def update
32   @movie = Movie.find params[:id]
33   if (@movie.update(movie_params))
34     redirect_to movie_path(@movie), :notice => "#{@movie.title} updated."
35   else
36     flash[:alert] = "#{@movie.title} could not be updated: " +
37     @movie.errors.full_messages.join(", ")
38     render 'edit'
39   end
40 end
```


Ahora nos aparece otro error:

```
NoMethodError in Movies#show

Showing C:/Users/calag/Desktop/Desarrollo-software-2023/Semana8/myrottenpotatoes/app/views/layouts/application.html.erb where line #12
raised:

undefined method `first' for nil:NilClass

Extracted source (around line #12):

10   <div class="container">
11     <%- if flash[:notice] %>
12     <div class="row"<%= (' adult' unless @movies.first.rating =~ /^G|PG$/ ) %>>
13     <%- elsif flash[:alert] %>
14     <div class="alert alert-danger text-center"><%=flash[:alert]%></div>
15     <% end %>

Rails.root: C:/Users/calag/Desktop/Desarrollo-software-2023/Semana8/myrottenpotatoes
```

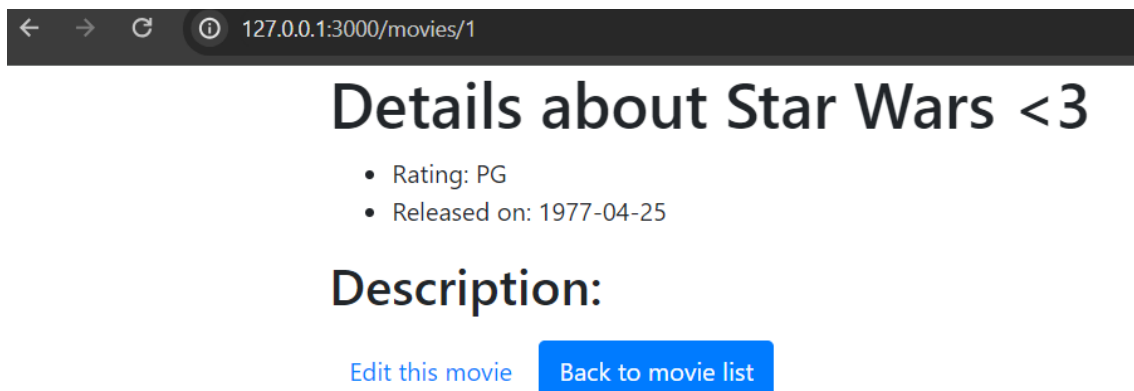
Este error indica que '@movies' está siendo llamado en la vista 'application.html.erb', específicamente en la línea 12, y se está tratando de acceder a la propiedad 'first' de '@movies', pero '@movies' parece estar siendo 'nil'.

Esto se puede solucionar asegurándonos de que la referencia a '@movies' solo se use en contextos donde realmente esté definida, entonces hacemos:

```
10   <div class="container">
11     <%- if flash[:notice] %>
12     <div class="row"<%= (' adult' unless @movies&.first&.rating =~ /^G|PG$/ ) %>>
13     <%- elsif flash[:alert] %>
14     <div class="alert alert-danger text-center"><%=flash[:alert]%></div>
```

El uso de '&.' antes de 'first' y 'rating' evita que se genere un error si '@movies' es 'nil' o si 'first' o 'rating' son 'nil'. Esto permite que el código maneje de manera segura estas situaciones en caso de que '@movies' no esté definida o no contenga elementos.

Entonces si volvemos a realizar el despliegue, ya podemos ver los cambios que antes no se podían:



Y listo, el trabajo ya está hecho.