

Desarrollo Web Full Stack Node

Ejercitación - M07C18

Mis películas - CRUD y relaciones

Introducción

Algo muy común en el mundo del desarrollo web es realizar las funcionalidades para dejar listo un CRUD completo: Creación de recursos, lectura de los mismos, actualización y borrado. Este será el foco del trabajo uniendolo con las posibles relaciones que tengan las tablas en la base de datos.

Requisitos

- **Paquete Sequelize instalado:** Para lograr el objetivo es fundamental lograr instalar Sequelize y los paquetes relacionados necesarios.
- **Conexión con base de datos:** El paquete sequelize no solamente precisa ser instalado sino que también precisa ser inicializado y configurado correctamente.
- **Sitio Web mostrando información real:** Hasta ahora no hemos obtenido información de una base de datos formal. Ya sea información de un JSON o escrita a mano, es hora de reemplazar dichos atajos con una versión consistente y real.

Parte 1 - CRUD

Objetivo

Para la construcción de esta versión del sitio web, el cliente espera contar con la posibilidad de acceso a las siguientes URL's:

- /movies (GET)
 - Se deberán listar todas las películas de la base de datos. Cada título de película deberá ser un hipervínculo para ver el detalle de la misma.
- /movies/detail/:id (GET)
 - Detalle de la película. Se deberá mostrar la película correspondiente al id que aparezca en la URL. Cada película deberá listar sus datos (Título, rating, premios, duración y fecha de estreno) y contar con dos (2) botones de acción: BORRAR y MODIFICAR.
- /movies/create (GET)
 - Muestra el formulario para creación de una película.
- /movies/create (POST)
 - Recibe los datos del formulario anterior y escribe la información en base de datos
- /movies/edit/:id (GET)
 - Muestra un formulario ya completo con los datos de la película según el id que indica la URL
- /movies/edit/:id (POST) --- **Idealmente PUT**
 - Recibe información del formulario mencionado anteriormente y en conjunto con el id que indica la URL actualiza la información de la película
- /movies/delete/:id (POST) --- **Idealmente DELETE**
 - Elimina la película indicada en la URL según el ID

Consignas

Es importante comenzar del trabajo realizado en la clase anterior.

1. Sobre las páginas

- /movies (GET)
 - El controlador deberá utilizar la conexión a base de datos y el modelo de Pelicula ya creado. Con eso en mente el método **findAll** permitirá obtener todas las películas de la base de datos.
- /movies/detail/:id (GET)
 - El controlador deberá utilizar la conexión a base de datos y el modelo de Pelicula ya creado. Con eso en mente el método **findByPk** permitirá obtener la película buscada. Recordá utilizar **req.params** para obtener el id de la URL.
- /movies/create (GET)
 - Muestra el formulario para creación de una película.
- /movies/create (POST)
 - El controlador deberá utilizar la conexión a base de datos y el modelo de Pelicula ya creado. Luego el método **create** permitirá la creación del registro. Luego el controlador debe redirigir al listado de películas
 - Idealmente la información recibida debe estar validada con **express-validator**
- /movies/edit/:id (GET)
 - El controlador deberá utilizar la conexión a base de datos y el modelo de Pelicula ya creado. Mediante el método **findByPk** y el id obtenible mediante **req.params** se enviará a la vista la información que ya tiene la película. Esa variable se comparte con la vista y mediante los atributos **value** de los campos del formulario se carga la información.
- /movies/edit/:id (POST) — **Idealmente PUT**
 - Recibe información del formulario mencionado anteriormente recuperable mediante **req.body**. El controlador deberá utilizar la conexión a base de datos y el modelo de Pelicula ya creado. Mediante el método **update** se modifica la información de la base de datos. Luego el controlador debe redirigir al listado de películas
 - Idealmente la información recibida debe estar validada con **express-validator**

- Utilizar los paquetes y modificaciones necesarias para que el método que responda al pedido sea **PUT**
- /movies/delete/:id (POST) — **Idealmente DELETE**
 - Previamente debe agregarse en el detalle de la película un botón de borrado que envíe a esta URL. El controlador recupera el id de la URL mediante **req.params**. El controlador deberá utilizar la conexión a base de datos y el modelo de Pelicula ya creado. Mediante el método **destroy** se elimina el registro y luego se redirige al listado de películas.
 - Proba de eliminar varias películas. ¿Esto anda bien siempre?

2. Bonus Track

Si lograste realizar toda la práctica, una buena idea es replicar el proceso pero con Actores.

Parte 2 - Relaciones

1. Definición de relaciones

Realizar en los modelos la definición de las siguientes relaciones:

- Una película tiene un género
- Un género tiene muchas películas
- Una película tiene muchos actores
- Un actor tiene muchas películas

2. Modificaciones en las páginas web

Una vez creadas las relaciones se piden las siguiente modificaciones:

1. El detalle de una películas debe incluir el nombre del género y un listado de los actores que trabajaron en la película.
 - a. ¡No olvides que **findAll** y **findByPk** pueden recibir la configuración **include** que nos permite acceder a las relaciones!
2. El nombre del género en la página del detalle de la película debe redirigir a una **nueva página** con el detalle del género. Esta página debe tener como mínimo el nombre del género y un listado con todas las películas del mismo.
3. El nombre del actor en la página del detalle de la película debe redirigir a una **nueva página** con el detalle del actor. Esta página debe tener como mínimo el nombre y apellido del actor y un listado con todas las películas donde actuó.
4. El formulario de creación y de edición de una película debe contar con un **<select>** que permite elegir el género de la película. Esto debe estar validado y plasmarse en base de datos. En el caso del formulario de edición ya debe figurar por default el género que ya tenía la película.
5. Modificar el eliminado de una película para que primero se borren las actuaciones en caso de haberlas.
 - a. Es probable que necesites crear un modelo para la tabla **actor_movie**
6. Crear una nueva página que permita seleccionar una película y un actor y permita guardar en la base de datos una nueva actuación.

¡Mucha suerte!