



Desarrollo Web Full Stack Node

Mis películas - API

Introducción

Si bien es probable que ya tengas armado un backend que extraiga información de la base de datos de películas y las muestre en un bonito frontend, se acerca un nuevo cliente a la mesa...un cliente externo.

Este cliente está maravillado con tu sitio web y quiere utilizar toda esa valiosa información e integrarla en otros sitios y aplicaciones.

Dado esto, te requiere que armes una API para poder extraer la información de tu sistema.

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.
- Scripts de base de datos. Los encontrás dentro de */database/scripts*
- Modelos creados : El paquete de Sequelize requiere la creación y configuración de modelos para esclarecer la forma de nuestra base de datos. Pero tranquilo@ te los damos armados.
- Ecosistema Express: Te proveemos una ya inicializada
- Postman: Para testar la API usaremos Postman. Si no lo tenés instalado, es un gran momento.





Objetivo

Para la construcción de esta API del sitio web, el cliente espera contar con la posibilidad de acceso a recursos:

1. /api/movies
2. /api/genres - **NINJA LEVEL**
3. /api/actors - **NINJA LEVEL**

Todos estos recursos se estipulan siguiendo el estándar **REST** que permitirá para cada recurso:

- Listar
- Ver detalles
- Crear
- Actualizar
- Borrar

Como adicional deseado el cliente pide:

- Incluir tests en postman

Consignas

1. Sobre la inicialización del proyecto

Configurar la conexión a la base de datos desde la aplicación base que les proveemos. Recordar el archivo **.env** y también el archivo **.sql** para la creación de las tablas y la base de datos en Workbench.

2. Rutas y Controllers

Te recomendamos que para cada uno de los recursos crees un archivo de ruta y un controlador para organizar mejor la aplicación.

Empecemos por los archivos de ruta y controlador para películas pero ya



pensando que más adelante vendrán archivos para más recursos...

Es una buena idea crear una subcarpeta dentro de las rutas llamada api para agrupar todos los archivos que respondan a apis. Lo mismo aplica para los controladores.

3. Sobre los endpoints y cómo probarlos

El recurso películas tendrá 5 endpoints. Te invitamos a que pienses cuales serán siguiendo el estándar REST. Recordá que debes proveer endpoints para:

1. Listado
2. Detalle de un elemento
3. Creación
4. Actualización
5. Borrado

Para hacerte las cosas más fáciles, te recomendamos que tal vez empieces utilizando los métodos GET y POST de HTTP. Si bien es cierto que no es estándar, y no es la mejor manera de hacerlo puede ayudarte a empezar a hacer las cosas más rápidamente.

Cada uno de estos endpoints se verá en el archivo de rutas y tendrá un método controlador que lo represente.

Bonus track (NINJA LEVEL) 🐱👤

Te dejamos algunas ideas de mejoras para esta API:

1. Verbos HTTP: Si utilizaste solo GET y POST...es hora de mejorar esto. Asegurate que los endpoints de actualización usen PUT o PATCH y que el de eliminación utilice el verbo DELETE.
2. Agreguemos más recursos: Hagamos los endpoints para acceder a géneros y actores.
3. Que el endpoint de movies muestre los datos asociados (genres y actors)



🕶️ **Ayuda:** Si ya tenés un backend completo hecho con sequelize encontrarás que la lógica de cada método ya la tengas escrito en otros lados de tu sitio ahora...
¿Cuál será la diferencia? 🤔

En vez de utilizar el método `res.render` utilizaremos `res.send`. No olvides que el formato esperado de respuesta es JSON.

No olvides tener siempre a mano Postman para probar cada uno de los endpoints que vayas creando y asegurar que la respuesta sea la esperada así como verificar que en la base de datos veas los cambios estipulados.

Comencemos a codear...