Programación Web Full Stack

 Creando nuestras propias APIs

Práctica integradora

Objetivo

Hasta ahora toda la comunicación entre el cliente y el servidor sigue el esquema habitual de pedido y respuesta, donde NodeJS entrega la vista terminada al navegador. Si el cliente quiere actualizar una parte de ese contenido, es necesario enviar todo de nuevo.

Vamos a ver cómo podemos usar las APIs para crear componentes dinámicos dentro de nuestras páginas, y de esa forma internalizar el fundamento teórico y práctico ya visto, el cual nos indica que una API es una especificación formal que **establece cómo un módulo de un software se comunica o interactúa con otro** para cumplir una o muchas funciones.

¡Buena suerte! 😎👍✨

Micro desafío - Paso 1:

Muchas veces les va a tocar trabajar con código existente, especialmente si entran a un equipo de desarrollo que ya tiene trabajo en progreso. Este es uno de esos casos, donde les toca hacer funcionar el proyecto de alguien más.

Utilizando de base el siguiente **proyecto creado con express** (Instalar todas las dependencias del proyecto, ejecutar la instrucción **npm install** 😉), además utilizaremos la base de datos **movies_db**.

Si se fijan, tenemos un controlador **(genresController.js).** En el mismo se encuentran varios métodos como lo son: **list, detail** estos efectúan diferentes consultas a la base de datos. Este controlador **(genresController.js)** le va a prestar servicio a nuestra aplicación pero no permite proveer endpoints para que podamos usarlo como una API.

Entonces su tarea será la de crear **dos (2) endpoints** que entreguen esos mismos datos. Les dejo un paso a paso y así ayudarles en la tarea 🤓👌

- Crear el controlador de movies de la api → **/controllers/api/genresController.js**
- Crear las rutas de movies de la api → **/routes/api/genresRouter.js**
- Agregar las rutas al archivo **app.js**

Para facilitar la tarea les pasamos la estructura de cómo deberías entregar los datos de los géneros:.

```
▼ meta:
    status:          200
    total:           12
    url:             "api/genres"
▼ data:
  ▼ 0:
      id:            1
      name:          "Comedia"
      ranking:       1
      active:        true
      created_at:    "2016-07-04T03:00:00.000Z"
      updated_at:    null
```


Si llegastes hasta aquí, significa que ya tienen los endpoints listos, ahora pueden llamarlos desde la url y verificar que efectivamente están devolviendo los datos de las películas:

📌 Micro desafío - Paso 2:

Si se fijan, tenemos un controlador **(moviesController.js).** En este controlador se encuentran varios métodos que efectúan diferentes acciones en la base de datos.

Su tarea será la de crear **dos (2) endpoints , uno por el método POST (create) y otro por el método DELETE (destroy)** que se encarguen de efectuar las tareas de insertar una nueva película y otra de borrar la misma. Les dejo un paso a paso y así ayudarles en la tarea 🤓👌

- Crear el controlador de movies de la api → **/controllers/api/moviesController.js**
- Crear las rutas de movies de la api → **/routes/api/moviesRouter.js**
- Agregar las rutas al archivo **app.js**

No se te olvide que al crear o borrar una película, la API, debe devolver el registro creado o el registro eliminado.

Es importante probar cada endpoint que creen antes de implementarlo para detectar cualquier falla, ahí es donde los clientes REST, como (Postman o Insomnia) van a brillar ✨🤖✨.

Bonus Track:

Si lograron realizar toda la práctica, una buena idea es replicar el proceso pero con el **modelo Actors**

Conclusión

Tal vez no lo vean ahora, pero las APIs son una manera muy poderosa de conectar aplicaciones. Ya hablamos muchas veces de reutilizar código ¿verdad 🤔?

Supongamos que desarrollamos una API de productos completa... entonces la lógica de creación, listado, búsqueda, edición y borrado ya está encapsulada. Ahora esa misma funcionalidad la pueden usar:

- El frontend que ve el cliente
- El backend que ve el administrador
- [Spoiler alert 🚨] El dashboard que vamos a desarrollar con React
- Una APP para dispositivos móviles
- Otros sitios que quieran listar nuestros productos

Así que les toca ponerse a pensar qué APIs van a desarrollar en su proyecto integrador 🤓🚀.

¡Hasta la próxima!