

Pedidos asincrónicos con JavaScript

Práctica integradora

Objetivo

En ejercicios anteriores vimos cómo podemos consumir las APIs desde el back-end con node-fetch. En este módulo estamos aprendiendo cómo hacerlo desde el front-end ya que es muy común conectar nuestro front-end con el back-end, ya sea uno propio o de terceros. Es por ello que vamos a practicar cómo consumir una API Rest de manera asíncrona desde el front-end usando fetch API.

¡Buena suerte!   



Micro desafío - Paso 1:

La idea principal es conectar [el siguiente front-end](#) con el back-end que creamos en clases anteriores, donde codificamos nuestras propias APIs. Si no lo tenemos, podemos utilizar el siguiente [proyecto creado con Express](#) como back-end (recordemos instalar todas las dependencias del proyecto, ejecutando el comando **npm install** 😊).

Además, aprovecharemos la base de datos [movies db](#) (no olvidemos activar el servicio de MySQL en nuestro equipo). De esta manera, todo funcionará correctamente.

Una vez realizados todos los pasos anteriores, debemos hacer lo siguiente en el proyecto front-end entregado:

- En **home.html**: listar todas las películas que nos trae el endpoint de listado de películas: `router.get('/', moviesAPIController.list);`
- En **formulario.html**: cargar los datos de una película en particular, para luego poder modificarla y/o eliminarla utilizando los endpoints:
 - Detalle de una película: `router.get('/:id', moviesAPIController.detail);`
 - Modificar una película: `router.put('/update/:id', moviesAPIController.update);`
- Utilizar el mismo formulario anterior para poder crear una película por medio del endpoint: `router.post('/create', moviesAPIController.create);`



Micro desafío - Paso 2:

Vamos con un desafío más complejo en nuestro **home.html**. En cada película veremos que tenemos una estrellita. Esta permitirá guardar las películas favoritas mediante un clic.

- Tendremos que capturar el evento click sobre cada estrellita (ojo que puede haber muchas estrellitas).
- Al hacer clic en la estrellita debe almacenar en el storage el id de la película seleccionada.

- Nótese que puede haber muchas películas favoritas. Dado esto, ¿qué tipo de dato debemos almacenar en storage? Luego tendremos una página llamada **favoritos.html**, donde debemos:
 - Listar las películas que han sido marcadas como favoritas por el usuario.
 - Si aún no tuviese ninguna película, esta página debe indicar que aún no tiene nada.
 - Agregar en el encabezado de home.html un botón a “Mis películas favoritas”, Este botón solo debe aparecer si el usuario ya tiene películas favoritas.

Conclusión

Aún cuando actualmente se están utilizando diferentes tecnologías para el desarrollo de proyectos de front-end independientes —como por ejemplo Vue.js, Angular, React, incluso JavaScript puro— es importante que aprendamos cómo se realizan las diferentes peticiones cuando deseamos consumir una API Rest usando Javascript puro. Esto nos brindará las bases necesarias para comprender cómo debemos conectarnos a las diferentes API Rest. De esta manera, luego podremos aplicarlos en las diferentes tecnologías con tan solo averiguar detalles particulares de implementación que tenga cada una de las tecnologías mencionadas.

¡Hasta la próxima!