DigitalHouse > Coding School

Desarrollo Web Full Stack Node

Ejercitación - M07C02

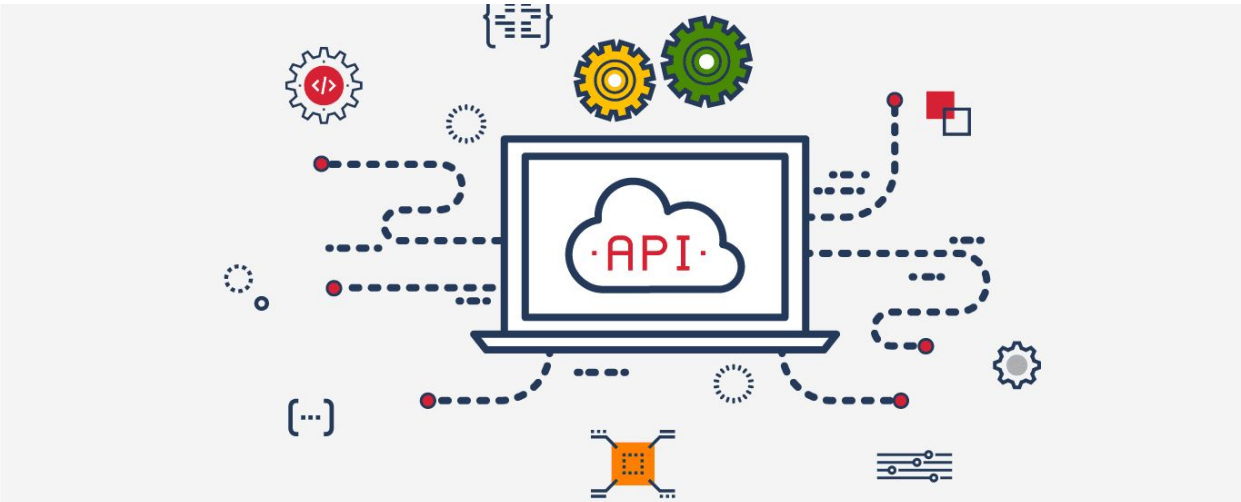
Ejercitación - APIs 2

Introducción

En la anterior ejercitación vimos cómo crear una API y repasamos cómo usar esos datos desde el frontend, pero... ¿qué pasa si necesitamos consumir una API desde el backend 🤔?

Además de eso, ¿qué pasa si lo que queremos es enviarle datos a la API en lugar de recibirlos 🤔🤔? ¿y si queremos usar métodos como PUT, PATCH y DELETE 🤔🤔🤔?

Pues para eso estamos aquí reunidos con nuestra querida librería amiga Axios que nos va a dar una mano con todo esto 🤖👌.



Consignas

A modo de calentamiento, vamos a arrancar con un simple ejemplo de consumo de APIs desde el backend, donde le pediremos un GIF determinado a Giphy.

Para seguir y como ejercitación del día 🏋️🧠🏋️ , seguiremos con el querido **Mercado Liebre** ®, esta vez trabajando en el carrito de compras.

Su desafío sería el crear los endpoints de la API del carrito para permitir que este se modifique sin necesitar recargar la página cada vez que se agrega, quita o modifica alguno de sus ítems.

Esta vez sí van a necesitar de algún cliente REST como Postman o Insomnia.

Algunas sugerencias de cómo encarar el ejercicio:

- Recuerden que por lo general la lógica de las APIs será muy similar a la que ya está programada en las rutas y controladores existentes.
- Es importante probar cada endpoint que creen antes de implementarlo para detectar cualquier falla, ahí es donde los clientes REST van a brillar ✨🤖✨.
- Vayan paso a paso, una acción a la vez.

Esperamos que estén listas y listos, porque ahí les van esos desafíos 🤓👆✨.

Desafío 1 — La entrada en calor

En el proyecto que les vamos a pasar todas las URLs llevan al mismo lugar… y ese agujero negro 🕳️ es nuestro famoso error 404 🤷‍♀️.

Su objetivo será utilizar la API de Giphy para traer un gif relacionado con este error y mostrarlo en la página.

Recuerden los pasos que ya vimos durante las clases:

- Descargar el proyecto → Link de descarga ⏬
- Instalar las dependencias de Node → Iniciar el proyecto
- [Registrarse en Giphy] → Crear una APP → Obtener una key
- Crear el archivo del recurso → **/requests/giphyRequests.js**
- Crear un método con **Axios** que traiga un gif buscando el texto “not found”
- Ejecutarlo en el controlador principal para enviarle el gif a la vista

Desafío 2 — Puesta en marcha de Mercado Liebre 🐇

Si hicieron la ejercitación del día uno, todo esto ya debería estar listo 🤓👆✨

En caso contrario les toca hacer funcionar el proyecto. Recuerden los pasos que ya vimos durante las clases:

- Descargar el proyecto → Link de descarga ⏬
- Instalar las dependencias de Node
- Crear la base de datos
- Configurar el acceso a la base de datos (ojo que este proyecto usa archivos **.env**).
- Iniciar el proyecto

Desafío 3 — Endpoint: agregar un ítem al carrito 🛒 ✚ 🛍️

Si se fijan, las acciones del carrito en este caso están en el controlador de usuarios. Si seguimos el camino de app.js → rutas → controlador → vista podemos decir lo siguiente del proceso de agregado:

- Ruta → Los datos del producto se envían por **POST**
- Ruta → Middlewares → El usuario debe estar autenticado para agregar un producto
- Ruta → Validaciones → El el producto a agregarse debe tener una cantidad mayor a 1
- Vista → Los datos enviados son **quantity** y **productId**
- Entidad → En todos los casos estamos manejando la tabla de **items**

Teniendo esto en cuenta su tarea será la de crear el juego de ruta + controlador + middleware para que el producto pueda ser agregado a través de un endpoint 🤓👌.

Vamos con algunos pasos que les van a aclarar el camino 🗺️.

- Crear el controlador de ítems de la api → **/controllers/api/itemsController.js**
- Crear el ruteador de ítems de la api → **/routes/api/itemsRouter.js**
- Agregar el ruteador al archivo **app.js**
- Programar el endpoint agregar un item → **/api/items [POST]**
- Probar que el endpoint funcione correctamente

Para facilitarles la tarea les pasamos la estructura de cómo deberían devolver los datos del ítem creado.


```
{
  "meta": {
       "status": 201,
       "message": "Product added to cart"
  },
  "data": {
       "item": {
       "id": 126,
       "salePrice": 4968,
       "quantity": "1",
       "subTotal": 4968,
       "state": 1,
       "userId": 102,
       "sellerId": 101,
       "productId": 221,
       "updatedAt": "2020-10-01T19:36:24.590Z",
       "createdAt": "2020-10-01T19:36:24.590Z"
       }
  }
}
```


Desafío 4 — Agregando productos con el endpoint

Ya tenemos el endpoint listo y funcionando, ahora queda implementarlo.

Para esto vamos a tener que hacer un pedido desde el listado de productos y en función de los datos que nos lleguen mostrar el carrito actualizado.

Vamos con otro paso a paso para ayudarlos en la tarea 🤓👌

- Agregar la librería Axios a la página de detalle, pueden usar este código
  `<script src="https://unpkg.com/axios/dist/axios.min.js"></script>`
- Agregar un script a la página de detalle → **public/js/addItem.js**
- Dentro de ese script seguir los siguientes pasos
  - Capturar el formulario (pueden que tengan que identificarlo de alguna manera)
  - Capturar los datos del formulario
  - Iniciar un pedido de Axios por **POST** a nuestro endpoint **/api/items**
  - Si nos llega un status **201**, redirigir al usuario a la página del carrito.

Desafío 5 — Endpoint: quitar un ítem del carrito 🛒 ✚ 🛍️

En este caso te pedimos que repitas todo lo anterior para la acción de quitar un ítem del carrito.

Vamos con ese paso a paso para este nivel final 🤓👌

Creando el endpoint

- Agregar la nueva ruta en el ruteador → **/routes/api/itemsRouter.js**
- Agregar la nueva acción en el controlador → **/controllers/api/itemsController.js**
- Programar el endpoint de últimos productos → **/api/items [DELETE]**
- Probar que el endpoint funcione correctamente

Consumiendo el endpoint

- Agregar la librería Axios a la página de carito, pueden usar este código
  `<script src="https://unpkg.com/axios/dist/axios.min.js"></script>`
- Agregar un script a la página de carito → **public/js/removeItem.js**
- Dentro de ese script seguir los siguientes pasos
  - Capturar el formulario (pueden que tengan que identificarlo de alguna manera)
  - Capturar los datos del formulario
  - Iniciar un pedido de Axios por **DELETE** a nuestro endpoint **/api/items**
  - Si nos llega un status **200**, capturar el elemento correspondiente de la vista y eliminarlo.

Cierre

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
- [Inserte infinitos ejemplos aquí]

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