



Práctica profesionalizante IX

Clase 5 dgp-dizp-mhp



Clase de hoy

- **Back-end**
- **Simplificación de api**

Aclaraciones

Para hacer el trabajo más realista con bases de datos, no utilizaremos **localStorage**, pero explicaremos de qué se trata de todas formas.

LocalStorage se basa en el concepto de almacenar información en formato JSON, como hemos visto. El JSON puede ser guardado dentro del caché del navegador.

¿Cuál es el problema? El caché es un almacenamiento **local**, lo que presenta una limitación. Si subo información, esta solo se reflejará en mi computadora, y no será visible para otros usuarios o dispositivos.

Que es una API

Una **API** (Application Programming Interface, o Interfaz de Programación de Aplicaciones) es un conjunto de reglas y herramientas que permiten que dos aplicaciones o sistemas se comuniquen entre sí. En términos simples, una API es un "intermediario" que permite que tu aplicación hable con otro software o servicio sin que tengas que preocuparte por los detalles internos de cómo funciona ese otro sistema.

Uso de APIs

Uso de APIs (como JSONBin):

Qué es: Un servicio en la nube que permite que varias personas (o dispositivos) puedan acceder a los mismos datos. En este caso, JSONBin almacena las tareas de manera centralizada.

Ventaja: Los datos son accesibles desde cualquier dispositivo conectado a Internet, ya que están guardados en un servidor externo. Esto lo hace ideal para aplicaciones que requieren colaboración o donde se necesita acceso desde múltiples ubicaciones.

Desventaja: Necesita una conexión a Internet para funcionar.

Relacionándolo con el Proyecto

En este proyecto, en lugar de guardar las tareas localmente (como lo harías con **LocalStorage**), estás usando una **API** (JSONBin) para guardar los datos de manera remota. Esto hace que la información esté disponible desde cualquier dispositivo conectado a Internet, lo que es más realista y similar a cómo funcionan las aplicaciones web modernas que utilizan bases de datos.

Diferencias entre ambos

LocalStorage es útil para guardar datos en el navegador, pero esos datos sólo estarán disponibles en esa computadora.

Usar una **API** como JSONBin permite que varios dispositivos puedan acceder a los mismos datos, haciendo que la aplicación funcione en cualquier parte.

JSONBin es como una base de datos en la nube, y usamos la API para comunicarnos con ella. En vez de guardar la información solo en tu navegador, la estamos guardando en un servidor, lo que permite que cualquier persona con acceso pueda ver los mismos datos.

Creación de cuenta jsonBin

Visitar: [JSONBin.io](https://jsonbin.io)

Registrarse y luego crear un nuevo "bin" (almacén de datos JSON).

Copiar la **API Key** desde la configuración de tu cuenta. Esta clave permitirá realizar operaciones en tu bin desde el código.

Otra cosa a tener en consideración

Al subir imágenes a una base de datos, lo ideal es **no** almacenarlas directamente en esta. En su lugar, se suben a un servidor en la nube que está diseñado específicamente para almacenar archivos grandes, como imágenes.

Este concepto es similar a lo que mencionamos sobre el almacenamiento local. **No usamos almacenamiento local** porque si guardamos las imágenes o datos en un dispositivo específico, solo estarán accesibles desde ese dispositivo. Al almacenarlas en un servidor en la nube, **todos** los usuarios pueden acceder a las imágenes solicitando la información a ese servidor, independientemente del dispositivo desde el cual se conecten

Una forma fácil de subir imagenes

La opción más fácil es : **Subir Imágenes a Imgur**

Visita [Imgur](#) y crea una cuenta (o usa la API para subir imágenes).

Podes subir imágenes directamente desde la página y obtener un **enlace de la imagen**.

Con esto en mente creemos nuestro primer proyecto

{

"tasks": [

{ "id": 1,

"task": "Comprar leche",

"image_url": "<https://i.imgur.com/example.jpg>",

"checked": false },

{ "id": 2,

"task": "Terminar el proyecto",

"image_url": "https://i.imgur.com/example2.jpg",

"checked": false }

] }

Lista de tareas


[Link de ejemplo](#)

Podrán ver cómo funciona accediendo al link, en el cual encontrarán también el repositorio con dicho código.

Tareas a seguir



En base al proyecto visto tendremos que crear un nuevo proyecto para que cada equipo pueda compartirlo, y tener una lista de tareas.

¡Hasta la próxima!

