

HINTS

CREACIÓN DEL SERVIDOR Y CONCEPTOS

- Cuando crees tu servidor, y estés leyendo el contenido del nuevo objeto URL, creado con la sintaxis “new URL ()”, recuerda que siempre puedes hacer un `console.log` de dicho objeto para leer todos sus parámetros.
- Debido a que estos objetos son parte de Node, siempre puedes encontrar y leer la documentación en su página oficial.
- A pesar de que es necesario conocer cómo funciona un servidor desarrollado con Node js nativo, es decir, sin frameworks, no debes preocuparte por la complejidad de algunas líneas de código, ya que actualmente el estándar en la industria utilizando Node es el framework Express. Con éste notarás la gran cantidad de código que se puede reducir, y lo simple que es definir rutas y leer parámetros. Una de las razones para aprender a levantar un servidor con Node js nativo, es conocer la gran diferencia y todos los beneficios del uso de Express.

POSTMAN

Como hemos podido observar, Postman es una herramienta muy poderosa, la cual permite la prueba de APIs y rutas de manera muy simple. La gran cantidad de opciones que ofrece puede ser un poco intimidante al principio, pero éstas son parte de conceptos que siempre estarán presentes dentro de tu carrera como programador. Puedes repasar respecto a las peticiones HTTP, y toda la información que se envía a través de la petición.

Si la respuesta de una petición desde Postman se queda esperando o “colgada”, revisa que:

- Tu ruta sea la correcta tanto en Postman, como en tu servidor.
- El puerto en el cual se está escuchando tu aplicación.
- No exista ningún otro programa o instancia que se encuentre utilizando el mismo puerto.

ESCRIBIENDO Y MODIFICANDO OBJETOS POR ID

- Cada vez que escribas un nuevo objeto, asegúrate de que éste sea escrito con su respectivo id, y cuando necesites modificarlo, también verifica que el id sea el correcto. Toma en cuenta lo mismo para cuando desees borrar un objeto.
- Es posible que en algún momento cometas un error de sintaxis, y que sea escrito en tu archivo de texto. Revisa periódicamente que éste mantenga la estructura deseada, para así evitar perder tiempo buscando un posible error que no se encuentre específicamente en tu código.
- Cuando estés controlando errores, también puedes verificar que el tipo de dato que esperas sea el correcto. Por ejemplo: si esperamos que la cantidad de comics sea siempre un número, no hay nada que evite que un usuario escriba una cadena de texto.