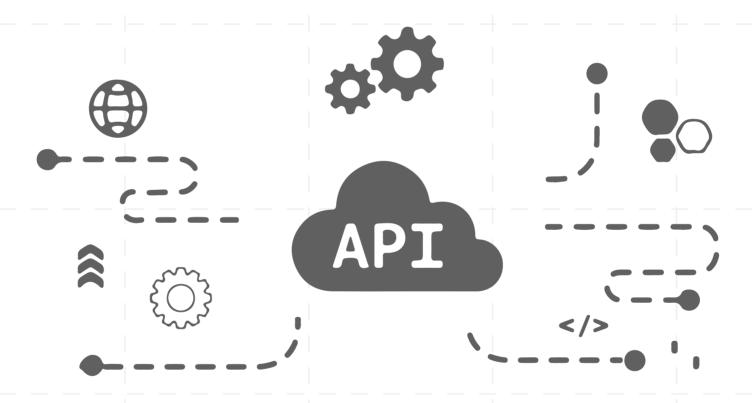


Ejemplo práctico de NodeJS



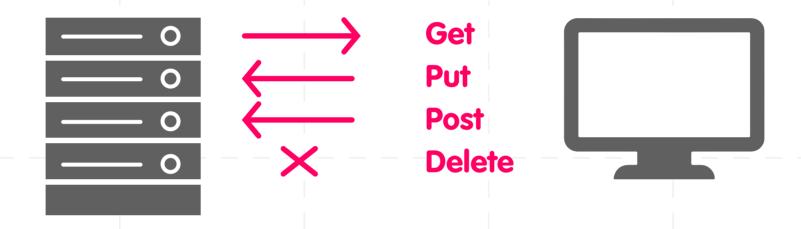
Hola

Con NodeJS en nuestros computadores podremos practicar un poco más, pero de una manera diferente a como lo hemos hecho. Ahora lo haremos pensando en un entorno web, para ello es necesario comprender lo que es una API, ya que en general las aplicaciones de Backend desarrolladas con NodeJS se presentan de esta manera. Una API (Application Programming Interface o interfaz de programación de aplicaciones) es un conjunto de funcionalidades expuestas a través de una serie de permisos definidos para cada una de ellas y permite a diversos componentes de software interactuar entre sí, reutilizando de una manera muy práctica las funcionalidades propias de los sistemas o generales para todos.



Cuando hablamos de una API web nos referimos a lo mismo, pero en esta ocasión las funcionalidades son expuestas mediante un servicio web, como lo es API REST. Los servicios REST son una interfaz que permite comunicar datos entre sistemas mediante el protocolo HTTP, aquel protocolo que utilizamos para navegar en internet. Los datos que se comunican a través de estos servicios REST se representan en formato JSON y XML. La especificación de estos dos formatos y sus variantes podrán verlo a profundidad en el material de esta semana.

Lo más importante en REST son los verbos que utiliza para referirse a diferentes acciones asociadas con las operaciones del CRUD en una base de datos. Los verbos que utiliza son los mismos del protocolo HTTP y con ellos se puede hacer: GET, POST, PUT y DELETE. REST es la solución a los problemas que presentaban en SOAP, una manera de exponer servicios web que tiene una mayor complejidad a la hora de desplegar y gestionar las acciones y solo corre sobre XML.



¿Por qué utilizar REST?

- Crea una petición HTTP que contiene toda la información necesaria en un objeto REQUEST. esta solicitud viaja desde un cliente hasta un servidor (Backend), este servidor atiende la solicitud y responde a través de un RESPONSE con la información solicitada.
- Las acciones de Rest se resumen en los métodos de HTTP:
 - Post: Crear.
 - Get: Obtener un listado completo o un dato específico.
 - Put: Actualizar.
 - Patch: Actualizar un dato específico.
 - Delete: Eliminar.

- El acceso a estas funcionalidades está disponible a través de una dirección URL, ya sea protegida por un certificado digital o no.
- Los servicios REST permiten separar la lógica del Backend del Frontend y de esta manera podemos tener diferentes accesos de Frontend, por ejemplo un computador, un dispositivo móvil, un televisor, etc.
- A partir de REST es muy sencillo diseñar una arquitectura de microservicios en diferentes tecnologías.
- 6 Esta tecnología es independiente del sistema operativo.
- Por último, y no menos importante, permite la escalabilidad horizontal y vertical, dándonos la opción de crecer en número de servidores o en características de un solo servidor.

Ya que tenemos claridad sobre lo que es una API y específicamente una API REST, podremos ir a la parte práctica para ver cómo desde NodeJS se implementa una serie de funcionalidades básicas.

Explicación video con ejemplo

Nodejs con paquetes de NPM:

- 1. Se requiere el **Visual Studio Code**.
- 2. Se muestra cómo integrar **Gitbash** dentro de **Visual Studio Code.**
- 3. Creación de una app en Node.
- 4. Instalación de paquetes necesarios para lanzar la API REST.
- 5. Función de listado de actividades.
- 6. Función de búsqueda de una actividad por ID.

De esta manera finalizamos el contenido de esta primera semana, espero que lo hubiesen disfrutado. Recuerden que después de cada sesión de clase deben realizar las actividades asincrónicas para profundizar en la temática.

Gracias por acompañarnos.

77.00



