

Resumen de Definiciones de Node.js

1. Arquitectura de Node.js

Node.js es un entorno de ejecución JavaScript basado en una arquitectura de un solo hilo y un modelo orientado a eventos. Utiliza el motor V8 para ejecutar código JavaScript en el servidor y libuv para manejar operaciones de entrada/salida (E/S) de manera no bloqueante.

2. V8

V8 es el motor JavaScript desarrollado por Google que convierte el código JavaScript en código máquina. Está encapsulado en Node.js y es responsable de ejecutar el código JavaScript en el servidor con alto rendimiento.

3. Bindings de Node.js

Los bindings son enlaces entre el código JavaScript y el código C/C++ subyacente que se ejecuta en Node.js. Estos permiten acceder a las funciones del sistema operativo y otras características de bajo nivel, manteniendo a Node.js rápido y eficiente.

4. libuv

libuv es una biblioteca en C que maneja las operaciones de E/S no bloqueantes y multiplataforma en Node.js. Es fundamental para el funcionamiento del bucle de eventos, ya que permite delegar operaciones intensivas en E/S a un pool de hilos.

5. Bucle de Eventos (Event Loop)

El bucle de eventos es un componente central en Node.js que permite manejar operaciones asíncronas sin bloquear el hilo principal. Revisa constantemente la cola de eventos y ejecuta las tareas a medida que se completan.

6. Cola de Eventos (Event Queue)

La cola de eventos es una estructura de datos donde se almacenan las tareas asíncronas que necesitan ser procesadas por el bucle de eventos. Las operaciones que finalizan, como la lectura de archivos, colocan eventos en esta cola para ser procesados.

7. APIs (Application Programming Interfaces)

Las APIs en Node.js son conjuntos de funciones que permiten interactuar con el sistema operativo, realizar operaciones de red, manejar archivos, entre otras tareas. Las APIs nativas de Node.js proporcionan acceso directo a estas funcionalidades.

8. Worker Threads

Los worker threads son hilos de trabajo que permiten ejecutar tareas intensivas en CPU de manera paralela al hilo principal. Son útiles para evitar que estas tareas bloqueen la ejecución del resto del código en Node.js.

9. Callbacks

Un callback es una función que se pasa como argumento a otra función y se ejecuta una vez que esa función ha completado su operación. Los callbacks son esenciales en Node.js para manejar operaciones asíncronas como la lectura de archivos o las solicitudes HTTP sin bloquear el flujo del programa.