Actividad 2 del complementario NodeJS

NOMBRE DEL INTEGRANTE

Valentina Hernández calderón

SERVICIO NACIONAL DE APRENDIZAJE SENA

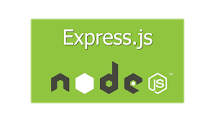
CENTRO AGROEMPRESARIAL Y DESARROLLO PECUARIO DEL HUILA

TECNOLOGO EN A.D.S.O

CIUDAD DE GARZÓN HUILA

AÑO DE 2023

¿Qué es Express y para que se implementa?

Express.js es un marco web de código abierto diseñado para Node.js que hace que sea fácil, rápido y eficiente crear aplicaciones web y API. Con Express, los desarrolladores pueden definir rutas para manejar diferentes URL y métodos HTTP, enviar y recibir datos fácilmente e implementar middleware para agregar funciones adicionales, como autenticación, compresión o registro de solicitudes. Con su enfoque minimalista y su gran comunidad de desarrolladores, Express.js se ha convertido en una opción popular para proyectos web de todos los tamaños y complejidades.

Express.js se utiliza principalmente para crear aplicaciones web y API en Node.js. Estos son algunos escenarios comunes para usar Express:

1. Servidores web: Express se usa para crear servidores web que pueden servir contenido estático, manejar solicitudes HTTP y responder a diferentes rutas y métodos.
2. API (interfaz de programación de aplicaciones): Express es una opción popular para crear API RESTful que permiten que otras aplicaciones interactúen con su aplicación y accedan a sus recursos y datos.
3. Aplicación de una sola página (SPA): Express se puede utilizar para atender el punto de entrada de la SPA y manejar las solicitudes del lado del servidor para el enrutamiento adecuado.
4. Aplicaciones web tradicionales: Express es adecuado para construir aplicaciones web tradicionales, donde se generan vistas dinámicas en el servidor y luego se envían al cliente.
5. Microservicios: para desarrollar microservicios en Node.js, Express proporciona un marco ligero y flexible para manejar solicitudes y respuestas entre varios componentes de un microservicio.
6. Aplicaciones híbridas: Express a veces se usa junto con otras tecnologías y marcos para crear aplicaciones híbridas que combinan la generación de vistas del lado del servidor con la interactividad del lado del cliente (por ejemplo, usando React o Angular).

En resumen, Express.js se implementa para facilitar el desarrollo de aplicaciones web y API en Node.js, proporcionando una base sólida para manejar el enrutamiento, las solicitudes y respuestas HTTP, y agregando middleware para ampliar la funcionalidad de la aplicación. solicitud. Su simplicidad y eficiencia lo convierten en una opción popular en el ecosistema Node.js.

¿Qué es una ruta absoluta y para que se implementa?

Una ruta absoluta es una forma completa y precisa de especificar la ubicación exacta de un archivo o directorio dentro del sistema de archivos. Se inicia desde la raíz del sistema o desde un punto de referencia conocido, y contiene todas las carpetas y subcarpetas necesarias para llegar al recurso deseado. En sistemas Unix o Linux, la ruta absoluta comienza con el caracter "/", mientras que en sistemas Windows puede comenzar con una letra de unidad (por ejemplo, "C:") o con el carácter "\" para referenciar la raíz de una unidad de red.

La implementación de rutas absolutas es útil en situaciones en las que se necesita acceder a recursos específicos de manera precisa y sin depender de la ubicación actual del proceso o la aplicación. Esto es especialmente relevante cuando se trabaja con archivos o directorios que se encuentran en ubicaciones fijas y no están sujetos a cambios frecuentes.

Al usar rutas absolutas, se asegura que el recurso se encuentre exactamente en la ubicación esperada, lo que garantiza que la aplicación pueda acceder a él sin importar en qué directorio se encuentre. Esto es especialmente importante en entornos donde la estructura del sistema de archivos es estricta o cuando se necesita garantizar un acceso confiable a recursos compartidos en red.

Por otro lado, las rutas relativas, en contraste, se definen en relación con la ubicación actual del proceso o la aplicación. Se utilizan cuando se desea referenciar recursos que están ubicados en el mismo directorio o en directorios relativos al punto actual.

La elección de usar una ruta absoluta o relativa depende del contexto y los requerimientos específicos del proyecto. Las rutas absolutas son más precisas y seguras, pero pueden resultar menos portátiles si se mueven a otro sistema o ubicación. En cambio, las rutas relativas son más flexibles, pero requieren que el recurso se encuentre en una ubicación relativa fija con respecto a la posición actual del proceso o la aplicación.

