

Módulo 6: Node.js

¿Qué es Node.js?

Se introduce Node.js como un entorno de ejecución para JavaScript en el lado del servidor. Se explica su importancia en el desarrollo moderno, sus ventajas y cómo se diferencia de la ejecución en el navegador.

Consola de Node.js

Se detalla el uso de la consola de Node.js para ejecutar comandos y scripts, explicando cómo interactuar directamente con el entorno para realizar pruebas y ejecutar tareas de manera rápida.

Ejecutando archivos JavaScript

Esta sección muestra cómo ejecutar archivos JavaScript desde Node.js. Se revisa el proceso para correr scripts y se discute la integración de Node.js en el flujo de trabajo del desarrollo backend.

File System (Manipular archivos)

Se explica cómo utilizar el módulo File System de Node.js para leer, escribir y manipular archivos en el sistema. Esto permite desarrollar aplicaciones que interactúan con el sistema de archivos, gestionando datos de forma dinámica.

npm init & package.json

Se abordan los conceptos básicos de npm (Node Package Manager), comenzando con la inicialización de un proyecto mediante el comando "npm init". Se explica la estructura y la importancia del archivo package.json para la gestión de dependencias y scripts en el proyecto.

Nodemon (npm)

Se presenta nodemon como una herramienta para el desarrollo, que permite reiniciar automáticamente el servidor cuando se detectan cambios en el código, facilitando el proceso de prueba y depuración durante el desarrollo.

Importaciones y exportaciones

En esta sección se profundiza en el uso de módulos en Node.js. Se explica cómo dividir el código en diferentes archivos mediante importaciones y exportaciones, y se revisa la sintaxis utilizada tanto en el sistema de módulos CommonJS como en el de ES6.

Variables de entorno

Se introduce el concepto de variables de entorno, explicando su utilidad para almacenar configuraciones sensibles o específicas del entorno (como claves de API o datos de conexión a bases de datos) sin incluirlos directamente en el código.

Herramientas para las prácticas

Se presentan diversas herramientas y utilidades que serán empleadas en las prácticas del módulo. Se incluyen recomendaciones sobre editores, terminales y otras aplicaciones que facilitan el desarrollo en Node.js.

PRÁCTICA 1

json-server

Se explica el uso de json-server para crear una API REST simulada de forma rápida. Esta herramienta permite levantar un servidor que sirve datos en formato JSON, facilitando la creación de prototipos y pruebas de frontend.

Postman

Se introduce Postman como herramienta para probar y documentar APIs. Se revisa cómo enviar peticiones HTTP, inspeccionar respuestas y gestionar colecciones de solicitudes para facilitar el desarrollo y la depuración.

Subir Repositorio a Github

Se detalla el proceso para subir el proyecto a Github, enfatizando la importancia del control de versiones y la colaboración mediante repositorios remotos.

Códigos HTTP

Se revisan los principales códigos de estado HTTP, explicando su significado y cómo se utilizan para comunicar el resultado de una petición en una API REST.

PRÁCTICA 2

Web Server

Se desarrolla una práctica orientada a la creación de un servidor web básico utilizando Node.js. Se explica la configuración inicial y cómo gestionar peticiones y respuestas.

Express.js

Se introduce Express.js, un framework para Node.js que simplifica la creación de aplicaciones web y APIs. Se abordan sus principales características y cómo integrarlo en un proyecto para manejar rutas y middleware.

Migrar Web Server a ES6

Se muestra cómo actualizar el servidor web creado a una sintaxis moderna utilizando ES6, aprovechando las mejoras en la organización del código y la utilización de importaciones y exportaciones.

Editar README.md con StackEdit

Se explica la importancia del archivo README.md en un proyecto, y se ofrece una práctica para editarlo utilizando StackEdit, de forma que se documente adecuadamente el proyecto y se faciliten futuras colaboraciones.

PRÁCTICA 3

API REST (CRUD) con Mongo

En esta práctica se construye una API REST completa utilizando Node.js y MongoDB. Se aplican los conceptos de creación, lectura, actualización y eliminación (CRUD) de datos, integrando la base de datos en el flujo de la aplicación.

Mongoose

Se introduce Mongoose, una biblioteca de modelado de datos para MongoDB en Node.js. Se explica cómo definir esquemas y modelos para gestionar de forma estructurada la información en la base de datos.

Middleware propio

Se aborda la creación de middleware personalizado en Express.js, mostrando cómo interceptar y procesar peticiones para agregar funcionalidades o validar datos antes de llegar a la lógica principal.

Body Parser

Se revisa el uso de body-parser (o su integración en Express.js) para analizar el cuerpo de las peticiones HTTP, permitiendo acceder a los datos enviados en formato JSON u otros tipos.

MongoDB

Se detalla la instalación, configuración y uso básico de MongoDB como sistema de gestión de bases de datos NoSQL, enfatizando su integración con Node.js a través de Mongoose y otros módulos.

Docker

Se introduce Docker como una herramienta para contenerizar aplicaciones. Se explica brevemente cómo crear y utilizar contenedores para desplegar el backend y la base de datos, facilitando el entorno de desarrollo y producción.

Mongo Compass

Se presenta Mongo Compass como una herramienta visual para gestionar y explorar bases de datos MongoDB, facilitando la administración, la realización de consultas y el análisis de los datos.

Probando la API REST

Se explica cómo probar la API REST creada, integrando herramientas como Postman y pruebas unitarias para asegurar el correcto funcionamiento de todas las rutas y operaciones.

Despliegue de backend y base de datos en Railway

Finalmente, se aborda el despliegue de la aplicación backend y la base de datos en Railway, una plataforma que permite alojar aplicaciones de forma sencilla y escalable. Se explica el proceso de configuración y despliegue para hacer la aplicación accesible en línea.