

Bootcamp de Full Stack

Bienvenidos a la clase N°55

- Framework Express
- Postman
- Parámetros de API
- Métodos HTTP
- Proyecto Integrador II - Consultas

Backend

Express JS

¿Qué es Express JS?

Express.js es un marco de desarrollo web para Node.js que facilita la creación de aplicaciones web y APIs. Node.js es un entorno de ejecución de JavaScript del lado del servidor, y Express.js proporciona una capa adicional sobre Node.js para simplificar y agilizar el desarrollo de aplicaciones web.

Consigna:

- Instalar Express JS y configurar EsLint
- Crear un servidor HTTP con Express JS
- Demostrar el uso de algunos códigos de estado HTTP y formatos de transferencia de datos:
 - Texto
 - HTML
 - JSON

Backend

Postman

¿Qué es Postman?

Postman es una plataforma de colaboración para el desarrollo de API (Interfaz de Programación de Aplicaciones). Proporciona herramientas para que los desarrolladores diseñen, prueben, documenten, monitoreen y compartan API de manera más eficiente. Postman se utiliza comúnmente en el proceso de desarrollo de software para simplificar y mejorar la gestión de las API.

Consigna:

- Instalar Postman. Se puede descargar desde aquí: <https://www.postman.com/downloads/>
- Registrarse e iniciar sesión en Postman
- Crear collection para el backend
- Realizar pruebas de endpoints

Backend

Parámetros de API

¿Qué son los parámetros de API?

En el contexto de una API, los parámetros son elementos que se utilizan para enviar información adicional a un servidor cuando se realiza una solicitud. Los parámetros permiten a los clientes especificar detalles sobre la operación que desean realizar o sobre cómo desean que se maneje la respuesta del servidor.

Tipos de parámetros:

- **Path Params:** Son los parámetros de la ruta. Estos se incluyen en la URL y generalmente representan variables en la ruta del recurso. Por ejemplo: `/productos/:id`
- **Query Params:** Son los parámetros de consulta. Estos se incluyen en la URL después de un signo de interrogación y se utilizan para refinar la respuesta. Por ejemplo: `/productos?color=rojo&origen=china`
- **Header Params:** Son los parámetros de encabezado: Son parte de la cabecera de la solicitud HTTP y se utilizan para proporcionar información adicional, como la autenticación o preferencias del cliente HTTP.
- **Request Body:** Es el cuerpo de la solicitud. Los datos se envían en el cuerpo de la solicitud en un formato específico (JSON o XML). Se aplica en POST, PUT y PATCH.

Backend

Métodos HTTP

¿Qué es un método HTTP?

Un método HTTP es un comando que indica la acción que se debe realizar en un recurso (endpoint) específico cuando se realiza una solicitud a través de HTTP. Cada solicitud HTTP que se envía desde un cliente a un servidor, incluye un método que especifica la acción que se desea realizar.

Métodos HTTP más utilizados:

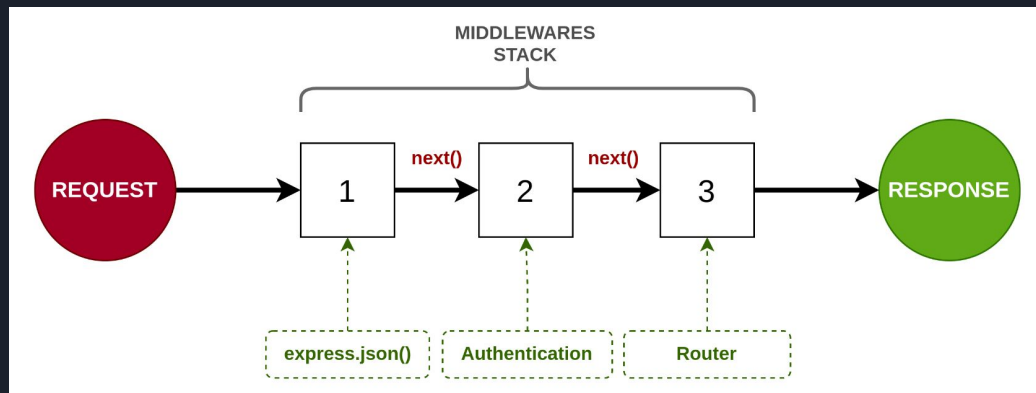
- **GET:** Solicita un recurso específico. No debe tener efectos secundarios en el servidor y se utiliza principalmente para recuperar información.
- **POST:** Envía datos al servidor para que sean procesados como parte del recurso identificado por la URI. Se utiliza comúnmente para enviar datos de formularios al servidor.
- **PUT:** Actualiza la totalidad de un recurso específico en la URI.
- **DELETE:** Elimina el recurso especificado en la URI.
- **PATCH:** Aplica modificaciones parciales a un recurso.

Backend

Métodos HTTP

¿Qué es un Middleware?

En Node.js, un middleware es una función que se ejecuta durante el ciclo de solicitud-respuesta del servidor. Se sitúa entre el cliente que realiza la solicitud y la lógica principal de manejo de la solicitud en el servidor. Su función principal es realizar operaciones en la solicitud entrante, la respuesta saliente o ambos, y luego pasar el control al siguiente middleware en la cadena o, en el caso del último middleware, a la lógica principal de manejo de la solicitud.



BREAK

Descansemos 15 minutos



Backend

Métodos HTTP

Consigna:

- Agregar los métodos POST, PUT, PATCH y DELETE
- Implementar con `use` el middleware `express.json()`
- Demostrar el uso de parámetros: Path Params, Query Params y Request Body

Proyecto Integrador II

Espacios de Consultas



CIERRE DE CLASE

Continuaremos en la próxima clase

