



EXPRESS

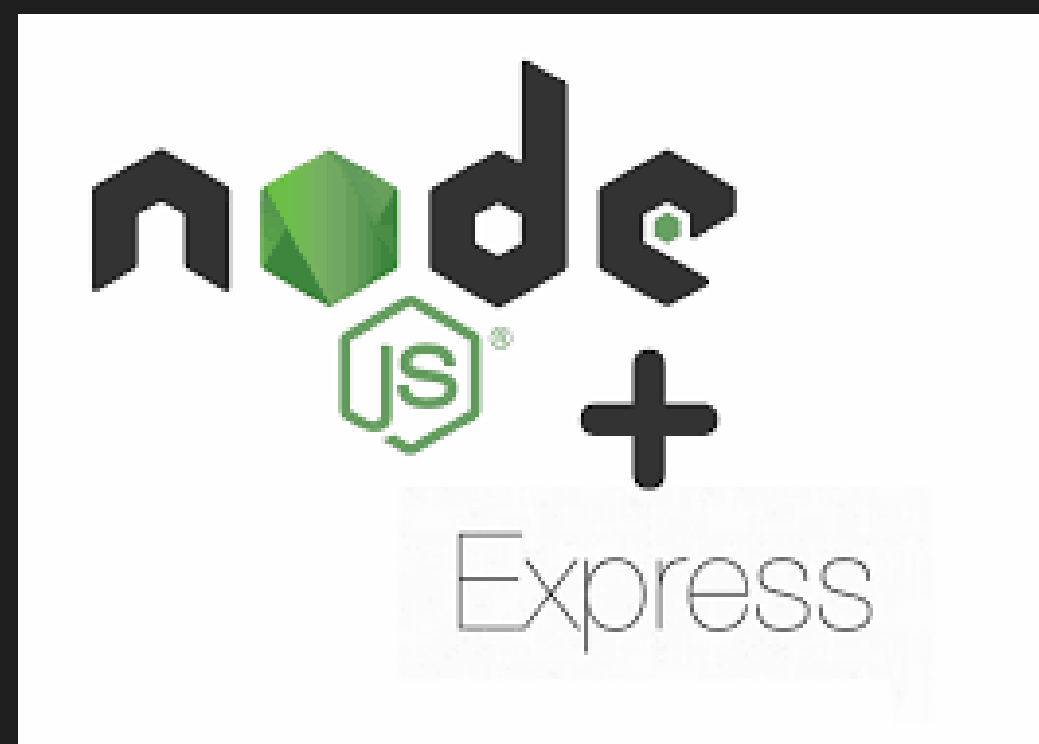
Express





Express

Es un framework web para Node.js, diseñado para simplificar el proceso de desarrollo de aplicaciones web y APIs.





Framework

Es una estructura o plataforma que proporciona un conjunto de herramientas, bibliotecas, y convenciones para facilitar y estandarizar el desarrollo de aplicaciones de software. Los frameworks son ampliamente utilizados en la programación debido a su capacidad para simplificar tareas comunes, mejorar la organización del código y promover buenas prácticas.



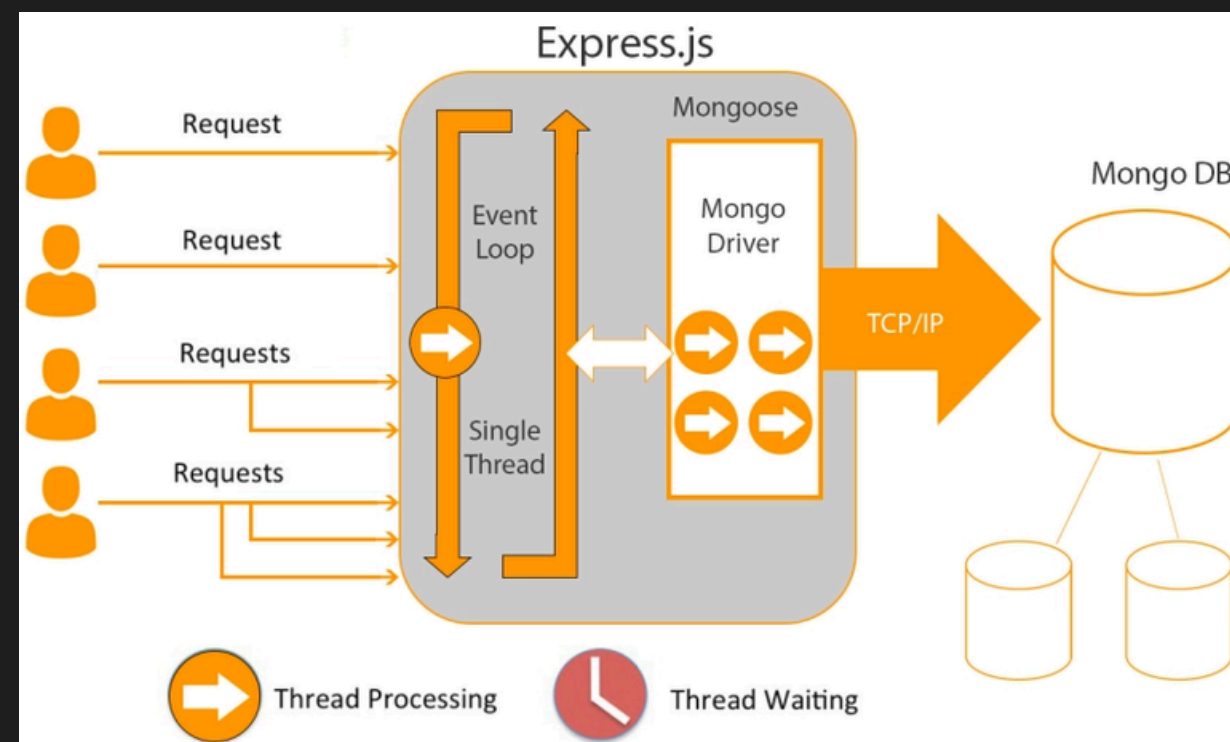
Express

Es un framework backend Node.js minimalista, rápido y similar a Sinatra, que proporciona características y herramientas robustas para desarrollar aplicaciones de backend escalables. Te ofrece el sistema de enrutamiento y características simplificadas para ampliar el framework con componentes y partes más potentes en función de los casos de uso de tu aplicación.



Express

El framework proporciona un conjunto de herramientas para aplicaciones web, peticiones y respuestas HTTP, enrutamiento y middleware para construir y desplegar aplicaciones a gran escala y preparadas para la empresa.





Solicitud HTTP

Una petición HTTP (HTTP request) es un mensaje enviado por un cliente (como un navegador web o una aplicación) a un servidor web para solicitar recursos o realizar alguna acción en el servidor.

Las peticiones HTTP pueden ser de varios tipos, incluyendo:

- GET
- POST
- PUT
- DELETE



Respuesta HTTP

Una respuesta HTTP (HTTP response) es el mensaje enviado por un servidor web al cliente en respuesta a una petición HTTP. Las respuestas HTTP también incluyen varios componentes:

- Line de estado
- Cabecera
- Cuerpo



Rutas

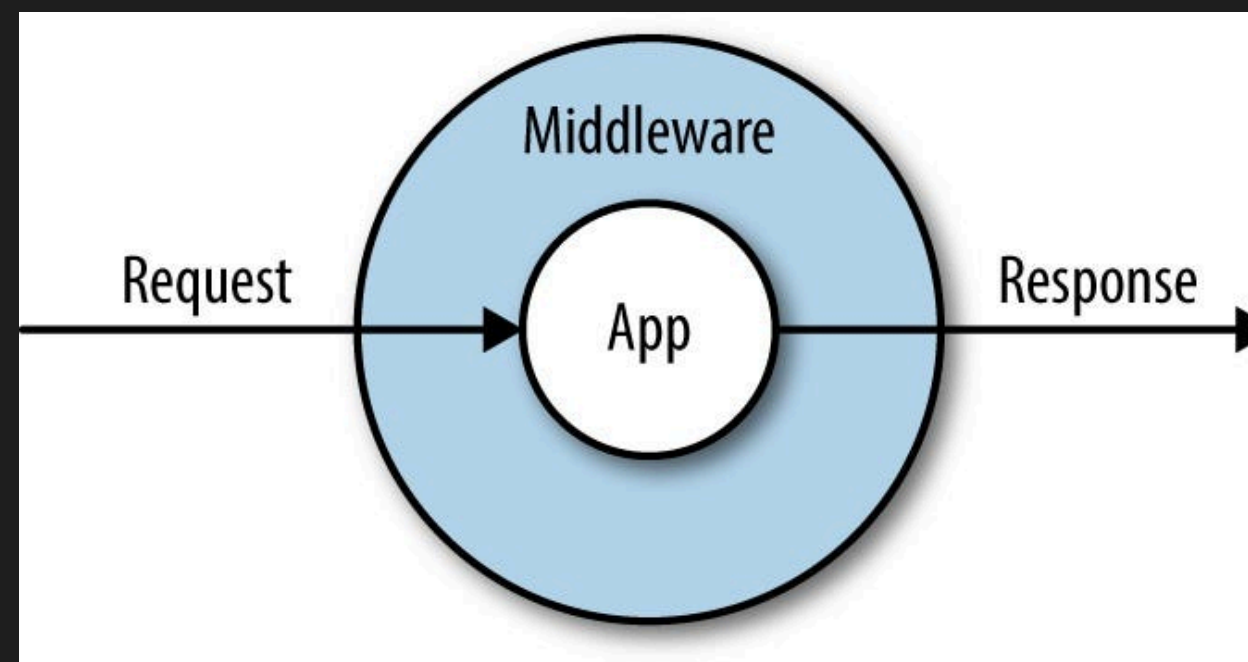
El enrutamiento (routing) es el proceso de definir cómo responde una aplicación a las peticiones del cliente a una URL específica o un patrón de URL. En Express.js, se utiliza el sistema de rutas para manejar las peticiones HTTP de manera específica. Cada ruta puede estar asociada a una función que se ejecutará cuando se acceda a esa ruta.

```
app.get('/', (req, res) => {  
  res.send('¡Hola, mundo!');  
});
```




Middleware

El middleware es una función que tiene acceso al objeto de la petición (req), al objeto de la respuesta (res), y a la siguiente función de middleware en el ciclo de solicitud/respuesta de una aplicación. En Express.js, el middleware se utiliza para realizar tareas comunes como análisis de cuerpos de peticiones, manejo de sesiones, y manejo de errores.





Middleware

Los middlewares son códigos que se ejecutan antes de que una petición HTTP llegue al manejador de rutas o antes de que un cliente reciba una respuesta, lo que da al framework la capacidad de ejecutar un script típico antes o después de la petición de un cliente.



Middleware

Con los middlewares, los desarrolladores pueden introducir scripts para interceptar el flujo de la aplicación; por ejemplo, los desarrolladores pueden utilizar los middlewares para comprobar si un usuario se ha conectado o desconectado correctamente.



Aplicaciones comunes de express

- APIs RESTful: Express es ampliamente utilizado para crear APIs RESTful debido a su capacidad para manejar diferentes tipos de peticiones HTTP y su sistema de ruteo flexible.
- Aplicaciones web: Con el soporte para motores de plantillas y middleware, Express es una excelente opción para desarrollar aplicaciones web completas.
- Aplicaciones en tiempo real: Al combinar Express con tecnologías como WebSockets, es posible desarrollar aplicaciones en tiempo real, como chats y notificaciones en vivo.



Arquitectura de proyecto

La arquitectura de un proyecto se refiere a la estructura organizativa del software, describiendo cómo se organizan y se comunican los diferentes componentes del sistema. En el contexto de una aplicación web, la arquitectura del proyecto define cómo se dividen y organizan los archivos y directorios, cómo se estructuran los datos y las responsabilidades de cada componente, y cómo se comunican entre sí.



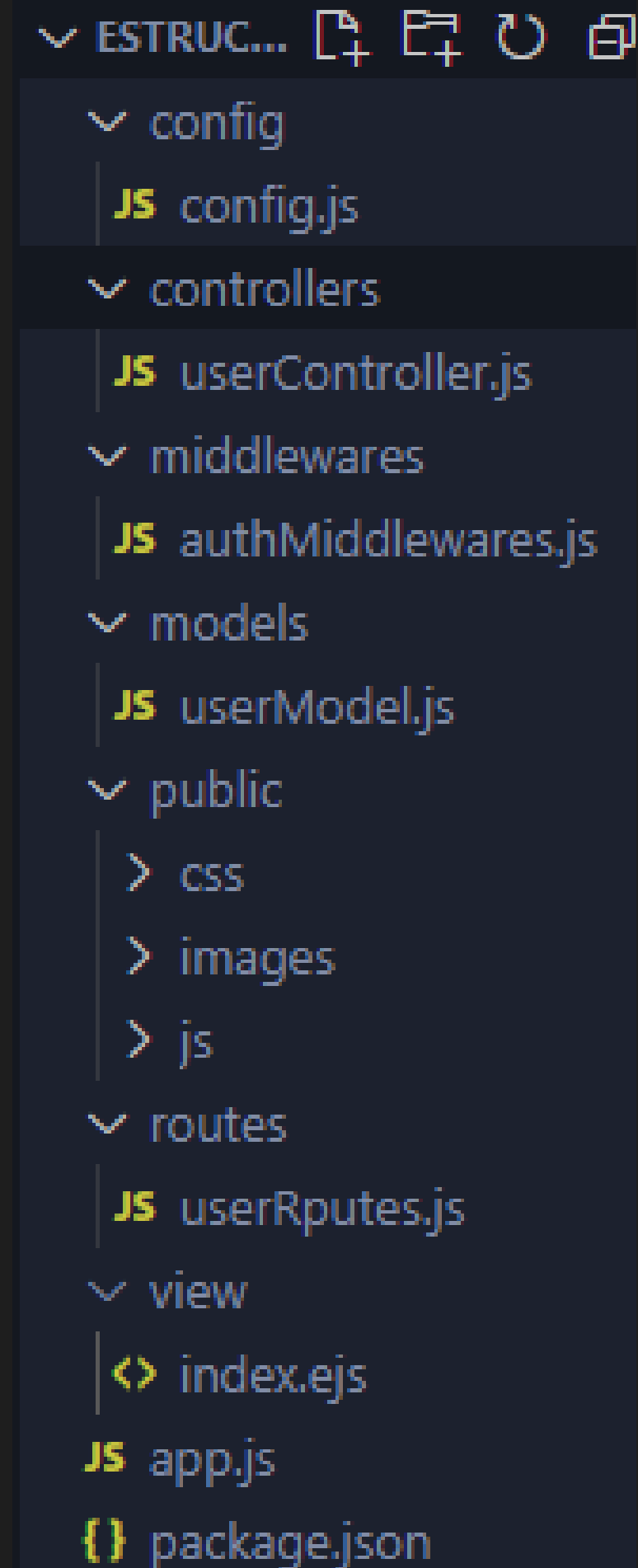
Organizacion de archivos y directorios

Directorio Raíz: Contiene archivos de configuración, scripts, y otros archivos esenciales.

Subdirectorios: Organizados por funcionalidad (e.g., routes, controllers, models, views, middlewares, etc.).



Ejemplo de arquitectura de proyecto.





Preguntas

