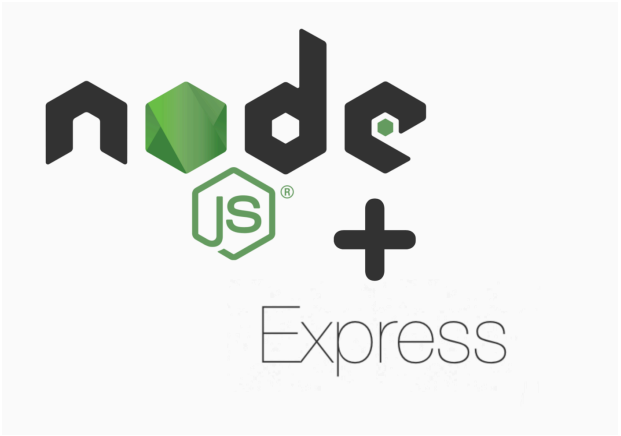


Ejercicio Node, Express y MongoDB

Evaluación del ejercicio

Partiendo del ejercicio de la semana anterior (Formularios y Drag&Drop), donde desarrollamos un formulario con su correspondiente verificación de campos, vamos a desarrollar un servidor en Express para la gestión de los archivos subidos. Podéis partir de vuestra implementación o de la solución de la semana pasada.

En concreto, si la verificación de campos es correcta, en lugar de enviar un mensaje por consola, vamos a subir estos ficheros al servidor (podemos convertir nuestros datos del formulario en un objeto que se puede enviar usando FormData). Para ello, usaremos una llamada Fetch API al servidor a /upload/files (revisad esto para saber cómo). En caso de que todo vaya bien, mostrará en el apartado de “Mensajes de estado” los siguientes datos:

Mensajes de estado

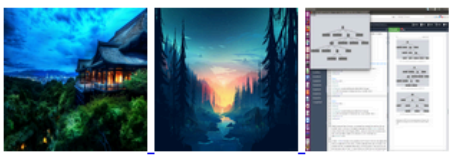
Datos del fichero: **3A0.jpg** Tipo: **image/jpeg** Tamaño: **1553742** bytes

Datos del fichero: **17010.jpg** Tipo: **image/jpeg** Tamaño: **979433** bytes

Datos del fichero: **BE.png** Tipo: **image/png** Tamaño: **367231** bytes

Resultados del formulario:

- Nombre: adrian
- Teléfono: 123456789
- Email: adrian@gmail.com
- Libro: Libro1
- Cantidad: 1
- Imágenes:



Además, si hacemos click en la imagen, esta se abrirá (hay que meter la imagen dentro de un enlace).

En el servidor, atenderemos la subida de archivos en un manejador de rutas para la ruta /upload/files. Usaremos multer para gestionar la subida de ficheros y guardarlos en /public/imgs/ (usad la opción *dest*). En la subida pondremos algunas condiciones:

- En total, la subida máxima será de 2MB. Para ello tendremos que usar la opción *limits*.

- Aceptaremos únicamente jpgs y pngs. Para ello se usa la opción *fileFilter*. En caso de no tener el formato correcto, mostrar un error por consola (en el servidor). Fijaos en la sección de "Error Handling" para ver cómo hacerlo.

La respuesta de vuelta contendrá un JSON con una respuesta booleana de éxito y con los datos del formulario y las rutas de los ficheros. La respuesta booleana la imprimiremos por consola en el cliente y, si la consulta fue un éxito, imprimimos los datos como hemos mencionado previamente.

Evaluación

Es necesario entregar los ficheros en esta entrega además de la IP+puerto. Si sólo se entrega uno de los dos o ninguno, la nota será de 0.

La puntuación será de 0 en caso de que el proyecto no esté dividido en carpetas correctamente. En concreto, se espera la siguiente estructura (no todos los elementos son necesarios):

- public/
|____ - index.html (si hubiese)
- js/
|____ - main.js
- views/ (si hubiese)
|____ - partials/ (si hubiese)
- xxx.ejs (si hubiese)
- imagenes/ (si hubiese)
- server.js (o como lo llaméis)
- package.json

Tarea	**Puntuación (sobre 10)**
Cliente: Petición fetch API	3
Cliente: mostrar respuesta servidor	2
Servidor: configuración de multer	2
Servidor: manejador de rutas /upload/files	3