

TEXT CLASS REVIEW

TEMAS A TRATAR EN EL CUE

- Recibiendo archivos.
- Obteniendo y modificando el nombre de archivo.
- Retornando mensajes de estado al cliente.
- Obteniendo atributos propios del archivo.
- Eliminando archivos del servidor.

RECIBIENDO ARCHIVOS

Al realizar una consulta al servidor, se hace con un método **GET** y para obtener los archivos, el método **fs.readdir()**, que se utiliza para leer de forma asíncrona el contenido de un directorio determinado. La devolución de llamada de este método devuelve una matriz de todos los nombres de archivo en el directorio.

La sintaxis es:

```
1 fs.readdirSync( path, options )
```

Ejemplo de forma de uso:

```
1 app.get('/files', async (req, res) => {
2     const directoryPath = "./files/";
3
4     fs.readdir(directoryPath, function(err, files) {
5         if (err) {
6
7             // Código de Gestión de errores en la búsqueda
8         }
9
10        // Código para la captura de la matriz de todos los
11        archivos
12
13        // Respuesta al cliente
14        res.status(200).send(listaArchivos);
15    })
16 });
```

OBTENIENDO Y MODIFICANDO EL NOMBRE DE ARCHIVO

El método **fs.renameSync()** se usa para cambiar el nombre de un archivo en la ruta antigua, a la nueva ruta dada de forma síncrona. Sobrescribirá el archivo de destino si ya existe.

Sintaxis:

```
1 fs.renameSync (rutaAntigua, rutaNueva)
```

Valores de propiedad:

rutaAntigua: contiene la ruta del archivo que se debe renombrar. Puede ser: una cadena, un búfer, o una URL.

rutaNueva: contiene la nueva ruta a la que se debe renombrar el archivo. Puede ser: una cadena, un búfer, o una URL.

```
1 // fs.renameSync() method
2
3 // importando el modulo
4 const fs = require('fs');
5
6 // Listando los archivos antes del cambio
7 getCurrentFileNames();
8
9 // renombrando el archivo
10 fs.renameSync('hello.txt', 'world.txt', () => {
11     console.log("\nArchivo Renombrado!\n");
12
13     // Listando los archivos luego del cambio
14     getCurrentFileNames();
15 });
16
17 // Función de listado de archivos en el directorio
18 function getCurrentFileNames() {
19     console.log("Archivos del sistema:");
20     fs.readdirSync(__dirname).forEach(file => {
21         console.log(file);
22     });
23 }
```

OBTENIENDO ATRIBUTOS PROPIOS DEL ARCHIVO:

El middleware express-fileupload hace que los archivos cargados sean accesibles desde la propiedad `req.files`. Por ejemplo: si está cargando un archivo llamado `info.txt`, y su nombre de campo es `fileName`, puede acceder a él a través de `req.files.fileName`. El objeto `fileName` contendrá la siguiente información:

fileName.name: el nombre del archivo cargado, es decir: mi `info.txt`.

fileName.mv: función para mover el archivo a otra parte del servidor.

fileName.mimetype: el tipo mime del archivo.

fileName.size: el tamaño del archivo en bytes.

fileName.data: una representación de búfer del archivo cargado.

Por ejemplo, la salida de la consola log:

```
1  ...
2  let fileReceived = req.files.fileName;
3  console.log(fileReceived); ...
4
5  La salida en la consola del terminal:
6
7      {
8        name: 'info.txt',
9        data: < Buffer 69 6 e 66 6 f 0 a > ,
10       size: 5,
11       encoding: '7bit',
12       tempFilePath: '',
13       truncated: false,
14       mimetype: 'text/plain',
15       md5: 'efd7c8e9aa906a97022dbed05369bf6a',
16     }
```

ELIMINANDO ARCHIVOS DEL SERVIDOR

Para esto hacemos uso del comando **fs.unlinkSync()**, que elimina un archivo, y espera hasta que se termine la operación para seguir ejecutando el código; también se puede usar el comando **fs.unlink()**, que ejecuta dicha operación de forma asíncrona.

La forma general de eliminar un archivo del servidor, siguiendo el método HTTP **DELETE**, es:

```
1 app.delete('/files/:name', async (req, res) => {
2   const fileName = req.params.name;
3   const directoryPath = "./files/";
4   try {
5     fs.unlinkSync(directoryPath + fileName);
6     console.log('File removed')
7     res.status(200).send("Archivo Eliminado Satisfactoriamente ");
8   } catch (err) {
9     console.error('ocurrió algo incorrecto al eliminar el archivo',
10 err)
11   }
12 });
```