### File Handling in Node.js:

Node.js provides built-in modules for file system operations, allowing you to read from and write to files on your system. Here's a breakdown of the file handling process:

#### Reading Files:

* Using fs.readFile: This function reads the entire contents of a file asynchronously.

const fs = require('fs');

fs.readFile('example.txt', 'utf8', (err, data) => {

 if (err) {

 console.error('Error reading file:', err);

 return;

 }

 console.log('File content:', data);

});

* Using fs.readFileSync: This function reads the entire contents of a file synchronously.

const fs = require('fs');

try {

 const data = fs.readFileSync('example.txt', 'utf8');

 console.log('File content:', data);

} catch (err) {

 console.error('Error reading file:', err);

}

#### Writing to Files:

* Using fs.writeFile: This function writes data to a file asynchronously, replacing the file if it already exists.

const fs = require('fs');

fs.writeFile('example.txt', 'Hello, World!', 'utf8', (err) => {

 if (err) {

 console.error('Error writing to file:', err);

 return;

 }

 console.log('File written successfully.');

});

* Using fs.writeFileSync: This function writes data to a file synchronously, replacing the file if it already exists.

const fs = require('fs');

try {

 fs.writeFileSync('example.txt', 'Hello, World!', 'utf8');

 console.log('File written successfully.');

} catch (err) {

 console.error('Error writing to file:', err);

}

#### Other Operations:

* File Existence: You can check if a file exists using fs.existsSync.
* Reading Directories: Use fs.readdir to read the contents of a directory.
* File Stats: Get information about a file using fs.stat.