**📁 File System Module (fs)**

The fs module in Node.js allows you to **interact with the file system** – reading, writing, updating files, etc.

✅ Use fs.promises or async/await for modern codebases. Below we cover both callback-based and Promise-based usage.

**🔹 1. Reading Files (Asynchronous)**

js

CopyEdit

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);

});

**🔹 2. Writing Files (Asynchronous)**

js

CopyEdit

fs.writeFile('output.txt', 'Hello, Node.js!', (err) => {

if (err) {

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

return;

}

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

});

**✅ Using fs.promises with async/await:**

js

CopyEdit

const fs = require('fs').promises;

async function readWriteDemo() {

try {

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

console.log('Read from file:', data);

await fs.writeFile('output.txt', data.toUpperCase());

console.log('Wrote modified content to output.txt');

} catch (err) {

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

}

}

readWriteDemo();

**🌐 HTTP Module - Creating a Basic Server**

The http module allows you to create an HTTP server in pure Node.js (without external frameworks like Express).

**🔹 1. Creating a Basic Server**

js

CopyEdit

const http = require('http');

const server = http.createServer((req, res) => {

res.writeHead(200, { 'Content-Type': 'text/plain' });

res.end('Hello from Node.js Server!\n');

});

server.listen(3000, () => {

console.log('Server running at http://localhost:3000/');

});

**🔹 2. Handling Simple HTTP Requests**

You can respond based on the URL path or method:

js

CopyEdit

const http = require('http');

const fs = require('fs');

const server = http.createServer((req, res) => {

if (req.url === '/' && req.method === 'GET') {

res.writeHead(200, { 'Content-Type': 'text/html' });

res.end('<h1>Welcome to My Node.js Server</h1>');

} else if (req.url === '/data' && req.method === 'GET') {

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

if (err) {

res.writeHead(500);

return res.end('Error reading file');

}

res.writeHead(200, { 'Content-Type': 'application/json' });

res.end(data);

});

} else {

res.writeHead(404);

res.end('Not Found');

}

});

server.listen(3000, () => {

console.log('Server listening on http://localhost:3000');

});

**🧠 Summary**

| **Feature** | **Description** |
| --- | --- |
| fs.readFile() | Reads file asynchronously using callback |
| fs.writeFile() | Writes to file asynchronously using callback |
| fs.promises | Modern Promise-based file operations |
| http.createServer() | Creates a basic HTTP server |
| Handling requests/responses | You can check req.url and req.method to serve different content |