**3) Create a Node.js file that opens an empty file using the File**

**Ans.-**

**Asynchronously Ensure a File**

This asynchronous way of creating an empty file uses the helper library [fs-extra](https://github.com/jprichardson/node-fs-extra). **fs-extra** provides the same API as the Node.js **fs** module and additional convenience methods.

The **fs-extra** package includes the extra **ensureFile(path)** method which creates the file at **path**:

const Fs = require('fs-extra')

async function ensureFile (path) {

await Fs.ensureFile(path)

}

A nice benefit of **ensureFile** is that it creates all missing directories along the path. It handles the creation of missing directories automatically. Sweet!

Also, it won’t override an existing file.

**Synchronously Ensure a File**

You can implement the synchronous approach with modules already integrated into the **fs** module from Node.js:

const Fs = require('fs')

function ensureFileSync (path) {

Fs.closeSync(

Fs.openSync(path, 'a')

)

}

**Notice:** the synchronous way will block the Node.js event loop while traversing the file system and creating the file. Please consider using the asynchronous approach.

**openSync(path)**returns a file descriptor:

const Fs = require('fs')

function ensureFileSync (path) {

const fileDescriptor = Fs.openSync(path, 'a')

return fileDescriptor

}

If you need the file descriptor for further processing, remember to close it when you finish all your handling. This won’t leave created file descriptors open.