# 575.https://stackoverflow.com/questions/69841275/unhandled-rejection-typeerror-fs-createreadstream-is-not-a-function-error-at

**T:**Unhandled Rejection (TypeError): fs.createReadStream is not a function. Error at the time of image uploading to ipfs( pinata)

**Q:**I was trying to upload image to pinata ipfs cloud from my frontend app of react js. But it shows me error:Unhandled Rejection (TypeError): fs.createReadStream is not a function. .  
  
Here is my code snippet:  
  
require('dotenv').config();const fs = require('fs');const axios = require('axios');const FormData = require('form-data');const key = process.env.REACT\_APP\_PINATA\_KEY;const secret = process.env.REACT\_APP\_PINATA\_SECRET;export const pinFileToIPFS = async (filePath) => {const url = "https://api.pinata.cloud/pinning/pinFileToIPFS";//we gather a local file for this example, but any valid readStream source will work here.let data = new FormData();data.append('file',fs.createReadStream(filePath));return axios.post(url, data, { maxContentLength: 'Infinity', //this is needed to prevent axios from erroring out with large files headers: { 'Content-Type': `multipart/form-data; boundary=${data.\_boundary}`, 'pinata\_api\_key': key, 'pinata\_secret\_api\_key': secret } }) .then(function (response) { //handle response here return { success: true, pinataUrl: "https://gateway.pinata.cloud/ipfs/" + response.data.IpfsHash }; }) .catch(function (error) { //handle error here console.log(error) return { success: false, message: error.message, } });};  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
How to fix it Or Is there any way of doing this rather than using 'fs'?

1 **Answer**

**A1:** ● 'fs' package has been deemed as malicious software, from github security placeholder page: "This package contained malicious code and was removed from the registry by the npm security team...". I wouldn't recommend using it.  
  
 ● The error message is specifying (TypeError): fs.createReadStream is not a function.  
  
 ● While doing research, I found fs-extra npm package is a good replacement for fs.  
  
With fs-extra, you can use the readJson (one of many methods) method executing it with a callback, promises, or async/await e.g.:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
const fse = require('fs-extra');...try { const fileObject = await fse.readJson(filePath); data.append('file', fileObject); console.log(fileObject);} catch (err) { console.error(err);}...  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
If this is doesn't work, based on this node.js example attempt the following:  
  
 ● Create a variable with the createReadStream method:  
  
 // This line opens the file as a readable streamlet readStream = fs.createReadStream(filePath);  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
 ● Use the following as intended to retrieve the response and catch any errors  
  
 // This will wait until we know the readable stream is actually valid before piping readStream.on('open', function () { // This just pipes the read stream to the response object (which goes to the client) readStream.pipe(res); // in your case: data.append('file', res); }); // This catches any errors that happen while creating the readable stream (usually invalid names) readStream.on('error', function(err) { res.end(err); });  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Thanks for your answer. But after fixing according to you it shows : TypeError: Object prototype may only be an Object or null: undefined (anonymous function) node\_modules/graceful-fs/polyfills.js:139 136 | } 137 | 138 | // This ensures util.promisify works as it does for native fs.read. > 139 | if (Object.setPrototypeOf) Object.setPrototypeOf(read, fs$read) | ^ 140 | return read 141 | })(fs.read) 142 | .......

**C2:**'fs' package has been deemed as malicious software -- The fs package on npm is malicious, yes, but require('fs') ALWAYS imports the fs module that's built-in to Node.js. By the way, fs-extras just adds a few things to fs, so if you use fs-extras, you're using fs.

**C3:**@user17242583 I was not fully aware of it and just noticed the warning messages about the malicious software and though it wouldn't be a good idea but you bring up a good point. arefin: I will update my answer to the best of my knowledge resorting back to "fs".

**C4:**Guys. Use import … from "node:fs/promises";.