# 562.https://stackoverflow.com/questions/70014669/error-enoent-no-such-file-or-directory-undefined

**T:**Error: ENOENT, No such file or directory 'undefined'

**Q:**PLease help! I've been looking everywhere to find a solution, including asking in the official Moralis forums as well and so far no one has been able to give me a solution. I've been at this for 3 days now and still in the same spot. Let's just say I've spent so much time looking at this code that I'm actually starting to understand it and I'm not even a programmer... So far I've seen other people have a similar "ENOENT" error but theirs all are followed by a certain file that's missing or something along the lines. My issue however simply just says "Undefined" and it doesn't tell me what's undefined, or how I should define it. I'm in the root folder as far as I know, I've installed all the dependencies, done "npm install", installed "canvas", did "npm init", cleaned the cache, deleted canvas, reinstalled canvas, deleted the "node\_modules" folder, uninstalled npm, and then re-installed npm. No matter what I keep getting the same error.  
  
To add some context, this is the original tutorial video I followed directly from the Moralis YouTube page.  
  
Generate NFTs with this simple code (this could make you millions) PART 1  
  
This is the official bone stock code from their GitHub and I've only changed the few couple of lines that it says to change in the video and on the GitHub to my own links. Here's a link to the GitHub  
  
Moralis Mutants NFT Collection | Generative Art Engine  
  
Here is the output I'm getting when I run "node index.js"  
  
PS C:\Users\Eli\Documents\moralis-mutants-nft-engine-main> node index.js################### Generative Art ## - Generating your NFT collection##################-----------------Mutating 1 of 10000- rarity: rare- dna: 01--02-04-02-01-010-03-03-04C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\node\_modules\canvas\lib\image.js:91 SetSource.call(img, src); ^Error: ENOENT, No such file or directory 'undefined' at setSource (C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\node\_modules\canvas\lib\image.js:91:13) at Image.set (C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\node\_modules\canvas\lib\image.js:62:9) at C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\node\_modules\canvas\index.js:34:15 at new Promise (<anonymous>) at loadImage (C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\node\_modules\canvas\index.js:23:10) at C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\index.js:83:25 at new Promise (<anonymous>) at loadLayerImg (C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\index.js:82:10) at C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\index.js:344:29 at Array.forEach (<anonymous>) at saveFile (C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\index.js:343:15) at handleFinal (C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\index.js:389:31) at startCreating (C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\index.js:392:11) at Object.<anonymous> (C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\index.js:448:1) at Module.\_compile (node:internal/modules/cjs/loader:1097:14) at Object.Module.\_extensions..js (node:internal/modules/cjs/loader:1149:10) { errno: 2, code: 'ENOENT', path: 'undefined', syscall: 'fopen'}Node.js v17.1.0  
  
I've also been trying to debug for hours now and this is what I get when I run the default debugger in Visual Studio Code  
  
C:\Program Files\nodejs\node.exe .\index.js ################## index.js:298# Generative Art # index.js:299# - Generating your NFT collection index.js:300################## index.js:301----------------- index.js:316Mutating 1 of 10000 index.js:317- rarity: rare index.js:323- dna: 01--02-04-01-02-05-04-04-04 index.js:335Uncaught Error: ENOENT, No such file or directory 'undefined' Process exited with code 1

**C1:**If we could see the file C:\Users\Eli\Documents\moralis-mutants-nft-engine-main\index.js, we could begin to follow the stack trace and see what code is involved. The issue is likely some bad parameter on a function call that goes a whole bunch of layers deep before an error occurs or in rarer cases, a bad module installation.

**C2:**@jfriend00 I included a link to the github that has the original index.js file in it from Moralis. I'll post it here again though, this is a more direct link. My index file is almost exactly the same if not exactly the same. The onlyy file I made changes to is the config.js file and those were the few changes that the video specifically says to change, which are the same directions on the github. github.com/ashbeech/moralis-mutants-nft-engine/blob/main/… This link has all the code.

2 **Answer**

**A1:**While debugging with you, I'm putting this into an answer only because I need to communicate multiple lines of code which don't really work in a comment.  
  
It looks like the problem originates in this sequence of code in https://github.com/ashbeech/moralis-mutants-nft-engine/blob/main/index.js:  
  
 // propagate information about required layer contained within config into a mapping object // = prepare for drawing let results = constructLayerToDna(newDna, layers, rarity); let loadedElements = []; // load all images to be used by canvas results.forEach((layer) => { loadedElements.push(loadLayerImg(layer)); });  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
When it calls loadLayerImg(layer), it appears that layer is faulty and does not contain some properties that it should. layer comes from results which comes from the call to constructLayerToDna() which, according to the comment is reading information from your config. So, it appears that something is missing or incorrect in the config that causes a faulty results array.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**I actually just figured it out, I'll post it in an answer, but I think what you just pointed out in the code makes sense considering how I fixed the problem.

**A2:**I found the culprit! So what was happening was, in one of my layer folders for “Base\_head” I had 5 folders for the 5 different rarity levels, however I only have 2 base head options so I had an image in “common”, an image in “legendary” and left the other 3 rarity folders empty for the “Base\_head” layer since I had no other base heads. I thought it didn’t matter if a folder for a certain layer rarity was empty. So I tested to see if that was an issue by filling the other 3 rarity folders with the same image for the “common” rarity “base head” -> so rarities “common, uncommon, rare, and super\_rare” get the same image and then I kept the 2nd image in the legendary rarity folder. This seems to fix it. I tried a few different things, tried lowering the resolution of my images thinking that maybe it's some kind of bottleneck in terms of how large of an image the program can process and pump out, however I can now use all the original 4320x4320px images and their respective folders and it’s pumping out NFTs finally at that resolution.  
  
In short, if you have your layer folders, with various rarity folders in each of them, then you need to make sure you have at least one image in every folder. You can’t leave any folder empty or else it won’t run due to not being able to find an image in said folder.  
  
This actually aligns with what you pointed out @jfriend00. You mentioned that "When it calls loadLayerImg(layer), it appears that layer is faulty and does not contain some properties that it should."  
  
Well, if the layer folder is empty, or the rarity folder within the layer folder is empty then the program will conclude that the "layer does not contain some properties" i.e. an image lol. Spent days on this and it ended up being a simple fix that was hard to find. Thanks everyone, hopefully this helps people into the future!