# 651.https://stackoverflow.com/questions/68468072/unable-to-deploy-nft-in-terminal

**T:**Unable to deploy NFT in terminal

**Q:**I already deployed my smart contract to my wallet and connected it to my Alchemy account.  
  
Here are my codings (Note that my contract address, PUBLIC\_KEY, PRIVATE\_KEY, API\_URL and alchemy address are edited for security purposes).  
  
mint-nft.js  
  
 require('dotenv').config();const API\_URL = process.env.API\_URL;const PUBLIC\_KEY = process.env.PUBLIC\_KEY;const PRIVATE\_KEY = process.env.PRIVATE\_KEY;const { createAlchemyWeb3 } = require("@alch/alchemy-web3");const web3 = createAlchemyWeb3(API\_URL);const contract = require("../artifacts/contracts/MyNFT.sol/MyNFT.json");const contractAddress = "My\_Contract\_Adress";const nftContract = new web3.eth.Contract(contract.abi, contractAddress);async function mintNFT(tokenURI) { const nonce = await web3.eth.getTransactionCount(PUBLIC\_KEY, 'latest'); //get latest nonce //the transaction const tx = { 'from': PUBLIC\_KEY, 'to': contractAddress, 'nonce': nonce, 'gas': 500000, 'data': nftContract.methods.mintNFT(PUBLIC\_KEY, tokenURI).encodeABI() }; const signPromise = web3.eth.accounts.signTransaction(tx, PRIVATE\_KEY); signPromise.then((signedTx) => { web3.eth.sendSignedTransaction(signedTx.rawTransaction, function(err, hash) { if (!err) { console.log("The hash of your transaction is: ", hash, "\nCheck Alchemy's Mempool to view the status of your transaction!"); } else { console.log("Something went wrong when submitting your transaction:", err) } }); }).catch((err) => { console.log(" Promise failed:", err); });}mintNFT("https://gateway.pinata.cloud/ipfs/My\_NFT\_Picture\_Hash");  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
alchemyContext.js  
  
"use strict";Object.defineProperty(exports, "\_\_esModule", { value: true });exports.makeAlchemyContext = void 0;var tslib\_1 = require("tslib");var sturdy\_websocket\_1 = tslib\_1.\_\_importDefault(require("sturdy-websocket"));var websocket\_1 = require("websocket");var jsonRpc\_1 = require("../util/jsonRpc");var version\_1 = require("../version");var alchemySendHttp\_1 = require("./alchemySendHttp");var alchemySendWebSocket\_1 = require("./alchemySendWebSocket");var httpProvider\_1 = require("./httpProvider");var sendPayload\_1 = require("./sendPayload");var webSocketProvider\_1 = require("./webSocketProvider");var NODE\_MAX\_WS\_FRAME\_SIZE = 100 \* 1024 \* 1024; // 100 MBfunction makeAlchemyContext(url, config) { var makePayload = jsonRpc\_1.makePayloadFactory(); if (/^https?:\/\//.test(url)) { var alchemySend = alchemySendHttp\_1.makeHttpSender(url); var \_a = sendPayload\_1.makePayloadSender(alchemySend, config), sendPayload = \_a.sendPayload, setWriteProvider = \_a.setWriteProvider; var senders = jsonRpc\_1.makeSenders(sendPayload, makePayload); var provider = httpProvider\_1.makeAlchemyHttpProvider(sendPayload); return { provider: provider, senders: senders, setWriteProvider: setWriteProvider }; } else if (/^wss?:\/\//.test(url)) { var protocol = isAlchemyUrl(url) ? "alchemy-web3-" + version\_1.VERSION : undefined; var ws = new sturdy\_websocket\_1.default(url, protocol, { wsConstructor: getWebSocketConstructor(), }); var alchemySend = alchemySendWebSocket\_1.makeWebSocketSender(ws); var \_b = sendPayload\_1.makePayloadSender(alchemySend, config), sendPayload = \_b.sendPayload, setWriteProvider = \_b.setWriteProvider; var senders = jsonRpc\_1.makeSenders(sendPayload, makePayload); var provider = new webSocketProvider\_1.AlchemyWebSocketProvider(ws, sendPayload, senders); return { provider: provider, senders: senders, setWriteProvider: setWriteProvider }; } else { throw new Error("Alchemy URL protocol must be one of http, https, ws, or wss. Recieved: " + url); }}exports.makeAlchemyContext = makeAlchemyContext;function getWebSocketConstructor() { return isNodeEnvironment() ? function (url, protocols) { return new websocket\_1.w3cwebsocket(url, protocols, undefined, undefined, undefined, { maxReceivedMessageSize: NODE\_MAX\_WS\_FRAME\_SIZE, maxReceivedFrameSize: NODE\_MAX\_WS\_FRAME\_SIZE, }); } : WebSocket;}function isNodeEnvironment() { return (typeof process !== "undefined" && process != null && process.versions != null && process.versions.node != null);}function isAlchemyUrl(url) { return url.indexOf("alchemyapi.io") >= 0;}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
.env  
  
API\_URL = "https://eth-rinkeby.alchemyapi.io/v2/KEY"PRIVATE\_KEY = "MY\_PRIVATE\_KEY"PUBLIC\_KEY = "MY\_PUBLIC\_KEY"  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
But then I was trying to deploy my NFT with metadata and nft-mint.js, I got these error.  
  
Can anyone please tell me what was the error about?

1 **Answer**

**A1:**Your issue may be with dotenv not reading in the values in your .env.If you add console.log(API\_URL), is it correct or is it undefined?If it is undefined, I was able to resolve the issue by adding the path to my .env like so:  
  
require('dotenv').config({path:"../.env"});  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
(In my case my mint-nft.js was in scripts/mint-nft.jsand .env is in the root directory.)