# 380.https://stackoverflow.com/questions/71337704/how-to-send-already-minted-nft-using-alchemy

**T:**How to send already minted NFT using alchemy

**Q:**I have minted some NFTs on opensea. These are on Polygon Mumbai network. Now I want to transfer these to token to other addresses using alchemy web3. Here is the code I am using.  
  
Note: This is supposed to run in nodejs restful API, so there is no wallet available that why I am manually signing the transaction.  
  
async function main() { require('dotenv').config(); const { API\_URL,API\_URL\_TEST, PRIVATE\_KEY } = process.env; const { createAlchemyWeb3 } = require("@alch/alchemy-web3"); const web3 = createAlchemyWeb3(API\_URL\_TEST); const myAddress = '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*' const nonce = await web3.eth.getTransactionCount(myAddress, 'latest'); const transaction = { //I believe transaction object is not correct, and I dont know what to put here 'asset': { 'tokenId': '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*',//NFT token id in opensea }, 'gas': 53000, 'to': '\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*', //metamask address of the user which I want to send the NFT 'quantity': 1, 'nonce': nonce, } const signedTx = await web3.eth.accounts.signTransaction(transaction, PRIVATE\_KEY); web3.eth.sendSignedTransaction(signedTx.rawTransaction, function(error, hash) { if (!error) { console.log("🎉 The hash of your transaction is: ", hash, "\n Check Alchemy's Mempool to view the status of your transaction!"); } else { console.log("❗Something went wrong while submitting your transaction:", error) } });}main();  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

3 **Answer**

**A1:**Assumed that you have Metamask installed in your browser, and that the NFT smart contract follows ERC721 Standard  
  
const { API\_URL,API\_URL\_TEST, PRIVATE\_KEY } = process.env;const { createAlchemyWeb3 } = require("@alch/alchemy-web3");const {abi} = YOUR\_CONTRACT\_ABIconst contract\_address = CONTRACT ADDRESSrequire('dotenv').config();async function main() { const web3 = createAlchemyWeb3(API\_URL\_TEST); web3.eth.getAccounts().then(accounts => { const account = account[0] const nameContract = web3.eth.Contract(abi, contract\_address); nameContract.methods.transfer(account, ADDRESS\_OF\_WALLET\_YOU\_WANT\_TO\_SEND\_TO, TOKEN\_ID).send(); }).catch(e => console.log(e));}main();  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Will this work in backend, there is no browser. just nodejs

**C2:**It doent work in nodejs backend, It needs the wallet to be connected, That why I am signing the transaction with private key.

**A2:**Had the same problem, because there is no example on transferring NFT token.There is a well explained 3-parts-example on ethereum website to mint an NFT.  
  
In the first part, step 10, it explains how to write a contract and also mentions the existing methods in the contract object extended:  
  
After our import statements, we have our custom NFT smart contract, which is surprisingly short — it only contains a counter, a constructor, and single function! This is thanks to our inherited OpenZeppelin contracts, which implement most of the methods we need to create an NFT, such as ownerOf which returns the owner of the NFT, and transferFrom, which transfers ownership of the NFT from one account to another.  
  
So, with these informations, I made an NFT transfer transaction between two addresses with my metamask mobile app. Then I searched the JSON of this transaction through etherscan API.  
  
In this way, I was able to transfer tokens to other addresses using alchemy web3 with this script:  
  
require("dotenv").config()const API\_URL = process.env.API\_URL; //the alchemy app urlconst PUBLIC\_KEY = process.env.PUBLIC\_KEY; //my metamask public keyconst PRIVATE\_KEY = process.env.PRIVATE\_KEY;//my metamask private keyconst {createAlchemyWeb3} = require("@alch/alchemy-web3")const web3 = createAlchemyWeb3(API\_URL)const contract = require("../artifacts/contracts/MyNFT.sol/MyNFT.json")//this is the contract created from ethereum example siteconst contractAddress = "" // put here the contract addressconst nftContract = new web3.eth.Contract(contract.abi, contractAddress)/\*\* \* \* @param tokenID the token id we want to exchange \* @param to the metamask address will own the NFT \* @returns {Promise<void>} \*/async function exchange(tokenID, to) { const nonce = await web3.eth.getTransactionCount(PUBLIC\_KEY, 'latest');//the transactionconst tx = { 'from': PUBLIC\_KEY, 'to': contractAddress, 'nonce': nonce, 'gas': 500000, 'input': nftContract.methods.safeTransferFrom(PUBLIC\_KEY, to, tokenID).encodeABI() //I could use also transferFrom};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) })}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**A3:**I Had the same problem. I need to transfer NFT in node.js back-end.I use network provider to use Moralis NetworkWeb3Connector.  
  
here's my repository for example:https://github.com/HanJaeJoon/Web3API/blob/2e30e89e38b7b1f947f4977a0fe613c882099fbc/views/index.ejs#L259-L275  
  
 await Moralis.start({ serverUrl, appId, masterKey, }); await Moralis.enableWeb3({ // rinkeby chainId: 0x4, privateKey: process.env.PRIVATE\_KEY, provider: 'network', speedyNodeApiKey: process.env.MORALIS\_SPEEDY\_NODE\_API\_KEY, }); const options = { type, receiver, contractAddress, tokenId, amount: 1, }; try { await Moralis.transfer(options); } catch (error) { console.log(error); }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
you can get speed node api key inMoralis dashboad > Networks > Eth Rinkeby(in my case) > Settings  
  
screenshot