# 369.https://stackoverflow.com/questions/71374163/how-to-request-transfer-of-nft-using-solana-web3-js

**T:**How to request transfer of NFT using @solana/web3.js

**Q:**I am working on a web application where a NFT owner can stake their NFT and earn rewards.In my application I need to create a transfer request of the NFT from the users wallet to my applications wallet.  
  
Code Example:  
  
import { Connection, PublicKey } from "@solana/web3.js";import \* as anchor from "@project-serum/anchor";import { web3 } from "@project-serum/anchor";import { Token, TOKEN\_PROGRAM\_ID, ASSOCIATED\_TOKEN\_PROGRAM\_ID,} from "@solana/spl-token";const doNFTTransfer = async function (mint: string, from: Wallet, to: string) { let connection = new Connection("https://api.devnet.solana.com"); const mintPublicKey = new web3.PublicKey(mint);// Mint is the Mint address found in the NFT metadata const ownerPublicKey = from.publicKey; const destPublicKey = new web3.PublicKey("MY\_APPS\_WALLET\_ADDRESS"); const mintToken = new Token( connection, mintPublicKey, TOKEN\_PROGRAM\_ID, from.payer ); // GET SOURCE ASSOCIATED ACCOUNT const associatedSourceTokenAddr = await Token.getAssociatedTokenAddress( mintToken.associatedProgramId, mintToken.programId, mintPublicKey, ownerPublicKey ); // GET DESTINATION ASSOCIATED ACCOUNT const associatedDestinationTokenAddr = await Token.getAssociatedTokenAddress( mintToken.associatedProgramId, mintToken.programId, mintPublicKey, destPublicKey ); const receiverAccount = await connection.getAccountInfo( associatedDestinationTokenAddr ); const instructions = []; if (receiverAccount === null) { console.log("receiver account is null!"); instructions.push( Token.createAssociatedTokenAccountInstruction( mintToken.associatedProgramId, mintToken.programId, mintPublicKey, associatedDestinationTokenAddr, destPublicKey, ownerPublicKey ) ); } instructions.push( Token.createTransferInstruction( TOKEN\_PROGRAM\_ID, associatedSourceTokenAddr, associatedDestinationTokenAddr, ownerPublicKey, [], 1 ) ); // This transaction is sending the tokens let transaction = null; for (let i = 0; i < instructions.length; i++) { transaction = new web3.Transaction().add(instructions[i]); } if (transaction) { let response = await from.sendTransaction(transaction, connection); console.log("response: ", response); } else { console.log("Transaction error: transaction data is null"); }};export default doNFTTransfer;  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
However when I run the code. The user is prompted to accept the transaction after approving the transaction I receive the following error.  
  
next-dev.js?3515:32 Transaction simulation failed: Error processing Instruction 0: invalid account data for instruction Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA invoke [1] Program log: Instruction: Transfer Program log: Error: InvalidAccountData Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA consumed 1781 of 200000 compute units Program TokenkegQfeZyiNwAJbNbGKPFXCWuBvf9Ss623VQ5DA failed: invalid account data for instruction  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
I've also tried swapping  
  
 ● mintToken.associatedProgramID with ASSOCIATED\_TOKEN\_PROGRAM\_ID  
  
 ● mintToken.programI with TOKEN\_PROGRAM\_ID  
  
imported from @solana/spl-token  
  
Anyone know what might be my issue ?

1 **Answer**

**A1:**I think this section is your problem:  
  
 // This transaction is sending the tokens let transaction = null; for (let i = 0; i < instructions.length; i++) { transaction = new web3.Transaction().add(instructions[i]); }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Every time the loop runs, it overwrites the entire transaction. This means that only the last transaction you made gets sent (createTransferInstruction). The following should work better!  
  
 // This transaction is sending the tokens let transaction = new web3.Transaction(); for (let i = 0; i < instructions.length; i++) { transaction.add(instructions[i]); }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Oh man, thank you for catching that! I totally overlooked that bit of code. Much appreciated!