# 2.https://stackoverflow.com/questions/73727734/i-am-not-be-able-to-transfer-nft-using-hashgraph-sdk-hedera-npm-how-to-transfe

**T:**I am not be able to transfer NFT using @hashgraph/sdk hedera npm. How to transfer nft using @hashgraph/sdk's in built method?

**Q:**I am not able to transfer NFT using @hashgraph/SDK hedera npm. How to transfer nft using @hashgraph/SDK's inbuilt method?  
  
Also, Nft is already associated with the seller account.  
  
Also, I am transferring token which is available in hashpack nft section.  
  
I am getting the below error on calling the TransferTransaction method of @hashgraph/sdk@2.17.1.Code:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
 const TransferNft = async (tokenId, sellerAccount, sellerId, buyerAccount, buyerId) => { try { const client = await getClient(); tokenId = TokenId.fromString(tokenId); let sellerKey = await getPrivateKey(sellerId); sellerKey = PrivateKey.fromString(sellerKey); let buyerKey = await getPrivateKey(buyerId); buyerKey = PrivateKey.fromString(buyerKey); // 2nd NFT TRANSFER NFT Alice->Bob let tokenTransferTx2 = await new TransferTransaction() .addNftTransfer(tokenId, 2, sellerAccount, buyerAccount) .addHbarTransfer(sellerAccount, Hbar.fromTinybars(100)) .addHbarTransfer(buyerAccount, Hbar.fromTinybars(-100)) .freezeWith(client) .sign(sellerKey); let tokenTransferTx2Sign = await tokenTransferTx2.sign(buyerKey); let tokenTransferSubmit2 = await tokenTransferTx2Sign.execute(client); let tokenTransferRx2 = await tokenTransferSubmit2.getReceipt(client); console.log(`\n NFT transfer Alice->Bob status: ${tokenTransferRx2.status} \n`); return tokenTransferRx2.status; } catch (error) { console.log('Error in HederaToken/TransferNft/TransferNft: \n', error) }};  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
receipt for transaction 0.0.40217130@1663228521.315536859 contained error status TOKEN\_NOT\_ASSOCIATED\_TO\_ACCOUNT  
  
Error:

1 **Answer**

**A1:**Before an account that is not the treasury for an HTS token can receive or send a specific token ID, they must be “associated” with the token — this helps reduce unwanted spam and other concerns from users that don’t want to be associated with any of the variety of tokens that are created on the Hedera network.  
  
This association between an account and a token ID can be done in two ways, manually or automatically. Note that automatic associations can be done for both existing and newly created accounts.  
  
Here's how you would do an auto-association for an existing account via an account update:  
  
 // AUTO-ASSOCIATION FOR ALICE'S ACCOUNT let associateTx = await new AccountUpdateTransaction() .setAccountId(aliceId) .setMaxAutomaticTokenAssociations(100) .freezeWith(client) .sign(aliceKey); let associateTxSubmit = await associateTx.execute(client); let associateRx = await associateTxSubmit.getReceipt(client); console.log(`Alice NFT Auto-Association: ${associateRx.status} \n`);  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Here's how you would do a manual association for an existing account:  
  
// MANUAL ASSOCIATION FOR BOB'S ACCOUNTlet associateBobTx = await new TokenAssociateTransaction() .setAccountId(bobId) .setTokenIds([tokenId]) .freezeWith(client) .sign(bobKey);let associateBobTxSubmit = await associateBobTx.execute(client);let associateBobRx = await associateBobTxSubmit.getReceipt(client);console.log(`Bob NFT Manual Association: ${associateBobRx.status} \n`);  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]