# 7.https://stackoverflow.com/questions/71666255/how-to-transfer-a-nft-to-another-smart-contract

**T:**How to transfer a NFT to another smart contract?

**Q:**I am trying to setup an escrow contract which takes a deposit as a NFT into the contract. I am struggling to deposit the NFT into the contract.  
  
I start by having a simple NFT that gets minted to the address with an id.  
  
contract MyEpicNFT is ERC721URIStorage { // Magic given to us by OpenZeppelin to help us keep track of tokenIds. using Counters for Counters.Counter; Counters.Counter private \_tokenIds; // We need to pass the name of our NFTs token and its symbol. constructor() ERC721 ("SquareNFT", "SQUARE") { console.log("This is my NFT contract. Woah!"); console.log(address(this)); } // A function our user will hit to get their NFT. function makeNft() public { uint256 newItemId = \_tokenIds.current(); \_safeMint(msg.sender, newItemId); \_setTokenURI(newItemId, "https://jsonkeeper.com/b/1LHH"); console.log("An NFT w/ ID %s has been minted to %s", newItemId, msg.sender); \_tokenIds.increment(); }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
An NFT w/ ID 0 has been minted to the msg.sender of the contract in this case: 0xdD870fA1b7C4700F2BD7f44238821C26f7392148  
  
I know now I need to approve the Escrow contract to be allowed to "spend" this token. To do this on MyEpicNft contract I call the approve function with the contract address of the MyEpicNft and the Id of the NFT that was minted.  
  
console.log: This is my NFT contract. Woah! 0x5fc7Dc95b4Bb48dbC9894fCaE417482cb8A6A45a  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
In my case I call the approve with (0x5fc7Dc95b4Bb48dbC9894fCaE417482cb8A6A45a, 0) - this seems to work. Now what I need to do in the NftEscrow contract is to deposit the NFT.  
  
The approve function i am using is from the ERC721 lib  
  
approve(address to, uint256 tokenId)  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
There is a depositNFT function which takes the NFT address - in our case 0x5fc7Dc95b4Bb48dbC9894fCaE417482cb8A6A45a and 0. But when i do this i get the error message:  
  
The transaction has been reverted to the initial state.Reason provided by the contract: "ERC721: transfer caller is not owner nor approved".Debug the transaction to get more information.contract NftEscrow is IERC721Receiver { enum ProjectState {newEscrow, nftDeposited, cancelNFT, ethDeposited, canceledBeforeDelivery, deliveryInitiated, delivered} address payable public sellerAddress; address payable public buyerAddress; address public nftAddress; uint256 tokenID; bool buyerCancel = false; bool sellerCancel = false; ProjectState public projectState; receive() external payable { } constructor(){ sellerAddress = payable(msg.sender); projectState = ProjectState.newEscrow; } function onERC721Received( address , address , uint256 , bytes calldata ) public pure override returns (bytes4) { return this.onERC721Received.selector; } function depositNFT(address \_NFTAddress, uint256 \_TokenID) public onlySeller { nftAddress = \_NFTAddress; tokenID = \_TokenID; ERC721(nftAddress).safeTransferFrom(msg.sender, address(this), tokenID); projectState = ProjectState.nftDeposited; } function depositEth() public payable { buyerAddress = payable(msg.sender); projectState = ProjectState.ethDeposited; } function confirmDelivery()public payable { ERC721(nftAddress).safeTransferFrom(address(this), buyerAddress, tokenID); sellerAddress.transfer(address(this).balance); } modifier onlySeller() { require(msg.sender == sellerAddress); \_; }}   
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Not sure what I am doing wrong and been stuck on this for a day. Why can't i deposit the NFT to the smart contract?

1 **Answer**

**A1:**Your escrow contract will revert with "ERC721: transfer caller is not owner nor approved" if none of the following are true:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
 ● msg.sender is the owner of contract  
  
 ● msg.sender is approved for all for the owner  
  
 ● msg.sender is approved for that token by the owner  
  
In your situation, we are opting for the 3rd option.  
  
What we need to check is  
  
 ● Did you really approve correct address?  
  
 ● Did approve function revert?  
  
When the escrow contract calls the NFT contract  
  
ERC721(nftAddress).safeTransferFrom(msg.sender, address(this), tokenID);  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
in NFT contract's point of view, the msg.sender is the escrow contract. So the problem is 99% with your approval process. Please share more information about that.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
EDIT:  
  
We are trying to get the following condition true, where the spender is the escrow contract's adress  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
return (spender == owner || isApprovedForAll(owner, spender) || getApproved(tokenId) == spender);  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
If getApproved is really returns the contract address, try adding following function to your nft contract:  
  
 /\*\* \* @dev Returns whether `spender` is allowed to manage `tokenId`. \* \* Requirements: \* \* - `tokenId` must exist. \*/ function \_isApprovedOrOwner(address spender, uint256 tokenId) internal view virtual override returns (bool) { require(\_exists(tokenId), "ERC721: operator query for nonexistent token"); address owner = ERC721.ownerOf(tokenId); console.log("spender %s", spender); console.log("getApproved %s", getApproved(tokenId); return (spender == owner || isApprovedForAll(owner, spender) || getApproved(tokenId) == spender); }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**I am simply using the ERC721 approve function which takes the NFT contract address and the token id? docs.openzeppelin.com/contracts/2.x/api/token/…

**C2:**what does it say when you call getApproved(0)

**C3:**I get the address of the contract of the NFT :

**C4:**``` constructor() ERC721 ("SquareNFT", "SQUARE") { console.log("This is my NFT contract. Woah!"); console.log(address(this)); } ``` which is address(this)

**C5:**@marcuspetty edited, please check