# 573.https://stackoverflow.com/questions/69854414/why-and-when-is-setapprovalforall-called

**T:**why and when is setApprovalForAll called?

**Q:**I'm having a hard time understanding why and when SetApprovalForAll is called on NFT generating contracts.  
  
Take Cool Cats, for example. And for convenience there's a screen shot of relevant transactions, below.  
  
I've noticed that for this contract and others, Set Approval For All is repeatedly called. In my own limited experience with a contract that allows minting, via \_safeMint, I'm seeing a SetApprovalForAll transaction in the logs too. It's not called directly in my code.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Why is it being called? Are wallets doing it as part of the minting process?

**C1:**Hi, i can i ask you something related to your question? in the setApprovalForAll it is asking me to pass an operator (which is supposed to be an address). does this mean i should pass all the wallet addresses manual?

2 **Answer**

**A1:**When you sell NFTs on a Marketplace, you need to authorize this marketplace to transfer sold items from your address to the buyer's address.  
  
This is what SetApprovalForAll is used for: because you trust the marketplace, you "approve" it to sell your NFTs (not all your NFTs, but the NFTs you own in the context of this contract).  
  
The marketplace is what is called the "operator" in the context of this API.

**C1:**Why do you need to authorise the marketplace to transfer ALL your NFTs (in the context of that contract) even when you only want to list a single NFT for sale? I tested this on NFTrade. I went to list one NFT that I own, and it wants to call setApprovalForAll(). Even if I do trust them, why should I give them more permissions than needed? They should be able to send an NFT for me by just having me call approve() on a single token ID. Am I missing something?

**C2:**There is a cost associated with setApprovalForAll, so maybe they thought it was more cost effective to set it once per collection contract

**A2:**this is openzeppelin ERC721.sol  
  
setApprovalForAll calls this function:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
// Approve `operator` to operate on all of `owner` tokensfunction \_setApprovalForAll( address owner, address operator, bool approved ) internal virtual { require(owner != operator, "ERC721: approve to caller"); \_operatorApprovals[owner][operator] = approved; emit ApprovalForAll(owner, operator, approved); }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
\_operatorApprovals is a mapping:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
// owner address allows operator addressmapping(address => mapping(address => bool)) private \_operatorApprovals;  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
An operator is a third party that is trusted and can be approved by anowner to have access to all the NFTs owned by them. If an owner givespermission to an operator, that operator is allowed to withdraw anyNFT from that owner's address and they are allowed to transfer it tothemselves or others.