# 559.https://stackoverflow.com/questions/70036366/how-to-transfer-erc1155-token-using-web3js

**T:**How to transfer ERC1155 token using web3js?

**Q:**I am trying to transfer ERC1155 tokens to another account. I have deployed the contract and have put down my contract code below.The problem I am facing is I get the below error when I call getMyToken function from my frontend web3js code.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
{code: -32603, message: 'execution reverted: ERC1155: caller is not owner nor approved', data: {…}}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Not sure what is happening. I am calling the function as below from frontend. The isApproved function is returning true.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
const selectedAccountIsApproved = await monkeysContract.methods.isApproved(selectedAccount).call();console.log('IS APPROVED RESULT: ', selectedAccountIsApproved);if(selectedAccountIsApproved) { const result = await monkeysContract.methods.mintMyMonkey(selectedAccount, 0).call(); console.log('Transfer Result: ', result);} else { console.log('Account not approved by owner for transfers');}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
NOTE:  
  
 ● I have called the setApprovalForAll method and approved the receiver account.  
  
 ● I am able to call isApprovedForAll and verify that the account is approved.  
  
 ● I am able to transfer tokens to the receiver through Etherscan contract interface.  
  
Contract Code:  
  
// SPDX-License-Identifier: MITpragma solidity ^0.6.0;import "@openzeppelin/contracts/token/ERC1155/ERC1155.sol";import "@openzeppelin/contracts/access/Ownable.sol";contract Monkeys is ERC1155, Ownable { uint256 public constant CC1 = 0; uint256 public constant CC2 = 1; mapping (uint256 => string) private \_uris; constructor() public ERC1155("https://game.example/api/item/{id}.json") { \_mint(msg.sender, CC1, 1000, ""); \_mint(msg.sender, CC2, 1000, ""); } function uri(uint256 tokenId) override public view returns (string memory) { return(\_uris[tokenId]); } function setTokenUri(uint256 id, string memory url) public onlyOwner { require(bytes(\_uris[id]).length == 0, "Cannot set uri twice"); \_uris[id] = url; } function transfer(address from, address to, uint256 id, uint256 amount, bytes memory data) public { safeTransferFrom(from, to, id, amount, data); } function isApproved(address user) public view virtual returns (bool) { return isApprovedForAll(owner(), user); } function getMyToken(address to, uint256 id) public { transfer(owner(), to, id, 1, 'test'); }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Is the selectedAccount the same with the address you used to send the call? The transfer method check if the msg.sender is the same with the from address or from list of approved for owner()

0 **Answer**