# 288.https://stackoverflow.com/questions/71774868/how-to-mint-erc1155-with-my-own-erc20-token

**T:**How to Mint ERC1155 with my own ERC20 token

**Q:**I'm tryin to mint my ERC1155 nfts with my own ERC20 so i did the following :  
  
1- created a variable of ERC20   
  
ERC20 public FCG = FCG<br>   
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
and in the constructor i pass the address of the ERC20 token contract  
  
FCG = ERC20(0x0fC5025C764cE34df352757e82f7B5c4Df39A836);  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
2- in my mint function i did the following code  
  
function mintCharacterPublic(address to, uint256 id, uint256 amount) public payable returns(bool){ require(to != address(0), "ERC1155: mint to the zero address"); require(FCG.approve(address(this), MINT\_PRICE), "can't approve your token"); require(uint256(FCG.allowance(msg.sender, address(this))) >= MINT\_PRICE, "Not enough of tokens"); FCG.transferFrom(msg.sender, address(this), MINT\_PRICE); mintCharacter(to, id, amount); \_balances[id][to] +=amount; return true;}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
but when i try to mint it does not apply and it throws "Not enough of tokens"so i think that the approve function does not function correctly.here is my smart contract for the token  
  
 // SPDX-License-Identifier: MITimport "@openzeppelin/contracts/token/ERC20/ERC20.sol";import "@openzeppelin/contracts/access/Ownable.sol";pragma solidity ^0.8.2;contract Token { mapping(address => uint) public balances; mapping(address => mapping(address => uint)) public allowance; uint public totalSupply = 1000000 \* 10 \*\* 18; string public name = "FightClub Token"; string public symbol = "FCT"; uint public decimals = 18; event Transfer(address indexed from, address indexed to, uint value); event Approval(address indexed owner, address indexed spender, uint value); constructor() { balances[msg.sender] = totalSupply; } function balanceOf(address owner) public view returns(uint) { return balances[owner]; } function transfer(address to, uint value) public returns(bool) { require(balanceOf(msg.sender) >= value, 'balance too low'); balances[to] += value; balances[msg.sender] -= value; emit Transfer(msg.sender, to, value); return true; } function transferFrom(address from, address to, uint value) public returns(bool) { require(balanceOf(from) >= value, 'balance too low'); require(allowance[from][msg.sender] >= value, 'allowance too low'); balances[to] += value; balances[from] -= value; emit Transfer(from, to, value); return true; } function approve(address spender, uint value) public returns (bool) { allowance[msg.sender][spender] = value; emit Approval(msg.sender, spender, value); return true; }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

1 **Answer**

**A1:**This must be a two-step process:  
  
 ● The user authorizes your contract to spend the token.  
  
 ● The user calls mintCharacterPublic.  
  
Your contract cannot grant itself permission to spend the originator's tokens. require(FCG.approve(address(this), MINT\_PRICE), "can't approve your token"); means that the eip-721 contract sets its own approval to MINT\_PRICE, not the end-user's.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]