# 328.https://stackoverflow.com/questions/71576780/i-want-to-mint-and-transfer-1-erc20custom-to-the-minter-itself-just-to-track

**T:**I want to mint and transfer 1 erc20(custom) to the minter itself, just to track erc20 transaction

**Q:**This is the code used, I am using polygon testnet for testing, the approve function is working fine but transferFrom is not working(error: -32000)  
  
// SPDX-License-Identifier: MITpragma solidity 0.8.0;import "@0xcert/ethereum-erc721/src/contracts/tokens/nf-token-metadata.sol";import "@0xcert/ethereum-erc721/src/contracts/ownership/ownable.sol";import "@openzeppelin/contracts/token/ERC20/ERC20.sol";contract newNFT is NFTokenMetadata, Ownable { ERC20 KOOLToken; constructor() { KOOLToken=ERC20(0xxxxxxxxxxxxxxxxxxxxxxxxxxx); nftName = "Test NFT 123"; nftSymbol = "TNFT321"; } function approve() public onlyOwner { KOOLToken.approve(address(msg.sender), 1); }function transferFrom() public onlyOwner{ KOOLToken.transferFrom(msg.sender,msg.sender, 1);}function mint(address \_to, uint256 \_tokenId, string calldata \_uri) public onlyOwner { super.\_mint(\_to, \_tokenId); super.\_setTokenUri(\_tokenId, \_uri);}}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Can you explain better your problem? What mean 'transfer 1 erc20(custom) to the minter itself'?

**C2:**I mean I have my own erc20 token, so I want to send 1 token to the minter itself, eg if address A mints an nft then A has to pay 1 token to itself, just that an erc20 transaction showes up in the blockchain explorer

**C3:**in simple terms, I want to send my erc20 token to the minter with the help of approve and transferfrom function

1 **Answer**

**A1:**Your -32000 error is caused by a revert in the ERC20 contract, because msg.sender does't have enough allowance to spend.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
According to EIP20, transferFrom provides:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
The transferFrom method is used for a withdraw workflow, allowing contracts to transfer tokens on your behalf. This can be used for example to allow a contract to transfer tokens on your behalf and/or to charge fees in sub-currencies. The function SHOULD throw unless the \_from account has deliberately authorized the sender of the message via some mechanism.  
  
Which means,even though the msg.sender is the owner here, it is not a valid source address for transferFrom unless you call approve on it before.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
You can approve the msg.sender, or simply use transfer function.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]