# 507.https://stackoverflow.com/questions/70666107/is-it-possible-to-call-a-erc20-function-inside-a-erc721-contract

**T:**Is it possible to call a ERC20 function inside a ERC721 contract?

**Q:**What I am trying to achieve is calling a transferFrom from ERC20 contract inside an ERC721 contract like this:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
My ERC20 contract:  
  
pragma solidity ^0.7.0;import "../openzeppelin-contracts/contracts/token/ERC20/IERC20.sol";import "../openzeppelin-contracts/contracts/math/SafeMath.sol";contract ERC20Token is IERC20 { using SafeMath for uint256; bytes32[] public candidateList; uint public totalTokens; uint public balanceTokens; uint public tokenPrice; // what is the voter address? // total tokens purchased // tokens voted per candidate struct voter { address voterAddress; uint tokensBought; uint256[] tokensUsedPerCandidate; } mapping(address => voter) public voterInfo; mapping(bytes32 => uint256) public votesReceived; string public symbol; string public name; uint8 public decimals; mapping(address => uint256) balances; mapping(address => mapping(address => uint256)) allowed; constructor(uint256 \_totalTokens, uint256 \_tokenPrice, bytes32[] memory \_candidateNames) { symbol = "NCToken"; name = "NCSOFT TOKEN"; decimals = 0; totalTokens = \_totalTokens; balanceTokens = \_totalTokens; tokenPrice = \_tokenPrice; candidateList = \_candidateNames; emit Transfer(address(0), msg.sender, totalTokens); function transferFrom(address from, address to, uint256 tokens) public override returns (bool) { //This is the function I am trying to call from ERC721 contract balances[from] = SafeMath.sub(balances[from], tokens); allowed[from][msg.sender] = SafeMath.sub(allowed[from][msg.sender], tokens); balances[to] = SafeMath.add(balances[to], tokens); emit Transfer(from, to, tokens); return true; }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
My ERC721 contract:  
  
// SPDX-License-Identifier: MITpragma solidity ^0.7.0;import "../openzeppelin-contracts/contracts/token/ERC721/IERC721.sol";import "../openzeppelin-contracts/contracts/token/ERC721/ERC721.sol";import "../openzeppelin-contracts/contracts/token/ERC721/IERC721Receiver.sol";import "../openzeppelin-contracts/contracts/math/SafeMath.sol";import "../openzeppelin-contracts/contracts/utils/Address.sol";import "../openzeppelin-contracts/contracts/utils/Counters.sol";import "./ERC20Token.sol";contract NFTtoken is ERC721 { using SafeMath for uint256; using Address for address; using Counters for Counters.Counter; Counters.Counter private \_tokenIds; bytes32[] candidates = [bytes32('Rama'), bytes32('Nick'), bytes32('Jose')]; ERC20Token ERC20TokenContract = new ERC20Token(1000, 1, candidates); //instantiating an ERC20 contract address payable public owner; mapping(bytes4 => bool) supportedInterfaces; mapping(uint256 => address) tokenOwners; //a mapping from NFT ID to the address that owns it mapping(address => uint256) balances; //a mapping from NFT ID to the address that owns it mapping(uint256 => address) allowance; //a mapping from NFT ID to approved address mapping(address => mapping(address => bool)) operators; //Mapping from owner address to mapping of operator addresses. // mapping (uint256 => string) idToUri; uint8 public decimals; uint256[] public allValidTokenIds; mapping(uint256 => uint256) private allValidTokenIndex; string[] public allNFTNames; struct NFT { //uint NFTID; string name; address creator; } mapping(address => NFT) public nftInfo; constructor() ERC721("NC NFT example", "NCNFT") { owner = msg.sender; decimals = 0; } function mint(string calldata nftName) external payable { uint256 newItemId = \_tokenIds.current(); \_mint(msg.sender, newItemId); nftInfo[msg.sender].name = nftName; nftInfo[msg.sender].creator = msg.sender; allValidTokenIndex[newItemId] = allValidTokenIds.length; allValidTokenIds.push(newItemId); \_tokenIds.increment(); } function transferNFT(address from, address to, uint256 tokenId) public returns (bool){ transferFrom(from, to, tokenId); ERC20TokenContract.transferFrom(to, nftInfo[from].creator, 10); //<-----\*\*\*\*\*\*\*\*\*\*\*This is throwing an error! I am trying to call ERC20Token.transferFrom. } function allNFTs() public view returns (uint256[] memory) { return allValidTokenIds; }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Error Message when transferNFT from ERC721 is called.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
MetaMask - RPC Error: [ethjs-query] while formatting outputs from RPC '{"value":{"code":-32603,"data":{"message":"VM Exception while processing transaction: revert ERC721: operator query for nonexistent token","code":-32000,"data":{"0x0b5d04087c39a8caa2f730815e42f619d33c9d0c3b8682c8c01d3f1ecf0e7d0f":{"error":"revert","program\_counter":7889,"return":"0x08c379a00000000000000000000000000000000000000000000000000000000000000020000000000000000000000000000000000000000000000000000000000000002c4552433732313a206f70657261746f7220717565727920666f72206e6f6e6578697374656e7420746f6b656e0000000000000000000000000000000000000000","reason":"ERC721: operator query for nonexistent token"},"stack":"RuntimeError: VM Exception while processing transaction: revert ERC721: operator query for nonexistent token\n at Function.RuntimeError.fromResults (C:\\Program Files\\WindowsApps\\GanacheUI\_2.5.4.0\_x64\_\_5dg5pnz03psnj\\app\\resources\\static\\node\\node\_modules\\ganache-core\\lib\\utils\\runtimeerror.js:94:13)\n at BlockchainDouble.processBlock (C:\\Program Files\\WindowsApps\\GanacheUI\_2.5.4.0\_x64\_\_5dg5pnz03psnj\\app\\resources\\static\\node\\node\_modules\\ganache-core\\lib\\blockchain\_double.js:627:24)\n at runMicrotasks (<anonymous>)\n at processTicksAndRejections (internal/process/task\_queues.js:93:5)","name":"RuntimeError"}}}}' {code: -32603, message: `[ethjs-query] while formatting outputs from RPC '{…/task\_queues.js:93:5)","name":"RuntimeError"}}}}'`}code: -32603message: "[ethjs-query] while formatting outputs from RPC '{\"value\":{\"code\":-32603,\"data\":{\"message\":\"VM Exception while processing transaction: revert ERC721: operator query for nonexistent token\",\"code\":-32000,\"data\":{\"0x0b5d04087c39a8caa2f730815e42f619d33c9d0c3b8682c8c01d3f1ecf0e7d0f\":{\"error\":\"revert\",\"program\_counter\":7889,\"return\":\"0x08c379a00000000000000000000000000000000000000000000000000000000000000020000000000000000000000000000000000000000000000000000000000000002c4552433732313a206f70657261746f7220717565727920666f72206e6f6e6578697374656e7420746f6b656e0000000000000000000000000000000000000000\",\"reason\":\"ERC721: operator query for nonexistent token\"},\"stack\":\"RuntimeError: VM Exception while processing transaction: revert ERC721: operator query for nonexistent token\\n at Function.RuntimeError.fromResults (C:\\\\Program Files\\\\WindowsApps\\\\GanacheUI\_2.5.4.0\_x64\_\_5dg5pnz03psnj\\\\app\\\\resources\\\\static\\\\node\\\\node\_modules\\\\ganache-core\\\\lib\\\\utils\\\\runtimeerror.js:94:13)\\n at BlockchainDouble.processBlock (C:\\\\Program Files\\\\WindowsApps\\\\GanacheUI\_2.5.4.0\_x64\_\_5dg5pnz03psnj\\\\app\\\\resources\\\\static\\\\node\\\\node\_modules\\\\ganache-core\\\\lib\\\\blockchain\_double.js:627:24)\\n at runMicrotasks (<anonymous>)\\n at processTicksAndRejections (internal/process/task\_queues.js:93:5)\",\"name\":\"RuntimeError\"}}}}'"[[Prototype]]: Object  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
localhost/:1 Uncaught (in promise) {code: -32603, message: `[ethjs-query] while formatting outputs from RPC '{…/task\_queues.js:93:5)","name":"RuntimeError"}}}}'`, stack: '{\n "code": -32603,\n "message": "[ethjs-query] wh…gaeaoehlefnkodbefgpgknn/background-0.js:1:216902)'}code: -32603message: "[ethjs-query] while formatting outputs from RPC '{\"value\":{\"code\":-32603,\"data\":{\"message\":\"VM Exception while processing transaction: revert ERC721: operator query for nonexistent token\",\"code\":-32000,\"data\":{\"0x0b5d04087c39a8caa2f730815e42f619d33c9d0c3b8682c8c01d3f1ecf0e7d0f\":{\"error\":\"revert\",\"program\_counter\":7889,\"return\":\"0x08c379a00000000000000000000000000000000000000000000000000000000000000020000000000000000000000000000000000000000000000000000000000000002c4552433732313a206f70657261746f7220717565727920666f72206e6f6e6578697374656e7420746f6b656e0000000000000000000000000000000000000000\",\"reason\":\"ERC721: operator query for nonexistent token\"},\"stack\":\"RuntimeError: VM Exception while processing transaction: revert ERC721: operator query for nonexistent token\\n at Function.RuntimeError.fromResults (C:\\\\Program Files\\\\WindowsApps\\\\GanacheUI\_2.5.4.0\_x64\_\_5dg5pnz03psnj\\\\app\\\\resources\\\\static\\\\node\\\\node\_modules\\\\ganache-core\\\\lib\\\\utils\\\\runtimeerror.js:94:13)\\n at BlockchainDouble.processBlock (C:\\\\Program Files\\\\WindowsApps\\\\GanacheUI\_2.5.4.0\_x64\_\_5dg5pnz03psnj\\\\app\\\\resources\\\\static\\\\node\\\\node\_modules\\\\ganache-core\\\\lib\\\\blockchain\_double.js:627:24)\\n at runMicrotasks (<anonymous>)\\n at processTicksAndRejections (internal/process/task\_queues.js:93:5)\",\"name\":\"RuntimeError\"}}}}'"stack: "{\n \"code\": -32603,\n \"message\": \"[ethjs-query] while formatting outputs from RPC '{\\\"value\\\":{\\\"code\\\":-32603,\\\"data\\\":{\\\"message\\\":\\\"VM Exception while processing transaction: revert ERC721: operator query for nonexistent token\\\",\\\"code\\\":-32000,\\\"data\\\":{\\\"0x0b5d04087c39a8caa2f730815e42f619d33c9d0c3b8682c8c01d3f1ecf0e7d0f\\\":{\\\"error\\\":\\\"revert\\\",\\\"program\_counter\\\":7889,\\\"return\\\":\\\"0x08c379a00000000000000000000000000000000000000000000000000000000000000020000000000000000000000000000000000000000000000000000000000000002c4552433732313a206f70657261746f7220717565727920666f72206e6f6e6578697374656e7420746f6b656e0000000000000000000000000000000000000000\\\",\\\"reason\\\":\\\"ERC721: operator query for nonexistent token\\\"},\\\"stack\\\":\\\"RuntimeError: VM Exception while processing transaction: revert ERC721: operator query for nonexistent token\\\\n at Function.RuntimeError.fromResults (C:\\\\\\\\Program Files\\\\\\\\WindowsApps\\\\\\\\GanacheUI\_2.5.4.0\_x64\_\_5dg5pnz03psnj\\\\\\\\app\\\\\\\\resources\\\\\\\\static\\\\\\\\node\\\\\\\\node\_modules\\\\\\\\ganache-core\\\\\\\\lib\\\\\\\\utils\\\\\\\\runtimeerror.js:94:13)\\\\n at BlockchainDouble.processBlock (C:\\\\\\\\Program Files\\\\\\\\WindowsApps\\\\\\\\GanacheUI\_2.5.4.0\_x64\_\_5dg5pnz03psnj\\\\\\\\app\\\\\\\\resources\\\\\\\\static\\\\\\\\node\\\\\\\\node\_modules\\\\\\\\ganache-core\\\\\\\\lib\\\\\\\\blockchain\_double.js:627:24)\\\\n at runMicrotasks (<anonymous>)\\\\n at processTicksAndRejections (internal/process/task\_queues.js:93:5)\\\",\\\"name\\\":\\\"RuntimeError\\\"}}}}'\",\n \"stack\": \"Error: [ethjs-query] while formatting outputs from RPC '{\\\"value\\\":{\\\"code\\\":-32603,\\\"data\\\":{\\\"message\\\":\\\"VM Exception while processing transaction: revert ERC721: operator query for nonexistent token\\\",\\\"code\\\":-32000,\\\"data\\\":{\\\"0x0b5d04087c39a8caa2f730815e42f619d33c9d0c3b8682c8c01d3f1ecf0e7d0f\\\":{\\\"error\\\":\\\"revert\\\",\\\"program\_counter\\\":7889,\\\"return\\\":\\\"0x08c379a00000000000000000000000000000000000000000000000000000000000000020000000000000000000000000000000000000000000000000000000000000002c4552433732313a206f70657261746f7220717565727920666f72206e6f6e6578697374656e7420746f6b656e0000000000000000000000000000000000000000\\\",\\\"reason\\\":\\\"ERC721: operator query for nonexistent token\\\"},\\\"stack\\\":\\\"RuntimeError: VM Exception while processing transaction: revert ERC721: operator query for nonexistent token\\\\n at Function.RuntimeError.fromResults (C:\\\\\\\\Program Files\\\\\\\\WindowsApps\\\\\\\\GanacheUI\_2.5.4.0\_x64\_\_5dg5pnz03psnj\\\\\\\\app\\\\\\\\resources\\\\\\\\static\\\\\\\\node\\\\\\\\node\_modules\\\\\\\\ganache-core\\\\\\\\lib\\\\\\\\utils\\\\\\\\runtimeerror.js:94:13)\\\\n at BlockchainDouble.processBlock (C:\\\\\\\\Program Files\\\\\\\\WindowsApps\\\\\\\\GanacheUI\_2.5.4.0\_x64\_\_5dg5pnz03psnj\\\\\\\\app\\\\\\\\resources\\\\\\\\static\\\\\\\\node\\\\\\\\node\_modules\\\\\\\\ganache-core\\\\\\\\lib\\\\\\\\blockchain\_double.js:627:24)\\\\n at runMicrotasks (<anonymous>)\\\\n at processTicksAndRejections (internal/process/task\_queues.js:93:5)\\\",\\\"name\\\":\\\"RuntimeError\\\"}}}}'\\n at new i (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:18:148782)\\n at s (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:18:146325)\\n at Object.internal (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:18:146935)\\n at y.<anonymous> (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-0.js:1:210928)\\n at Object.h (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:18:35204)\\n at u (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:1:117610)\\n at y.a.emit (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:1:118146)\\n at y.\_setTransactionStatus (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-1.js:1:46740)\\n at y.setTxStatusFailed (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-1.js:1:45972)\\n at B.\_failTransaction (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-0.js:1:225990)\\n at B.approveTransaction (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-0.js:1:217659)\\n at async B.updateAndApproveTransaction (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-0.js:1:216902)\"\n}\n at new i (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:18:148782)\n at s (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:18:146325)\n at Object.internal (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:18:146935)\n at y.<anonymous> (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-0.js:1:210928)\n at Object.h (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:18:35204)\n at u (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:1:117610)\n at y.a.emit (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/common-0.js:1:118146)\n at y.\_setTransactionStatus (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-1.js:1:46740)\n at y.setTxStatusFailed (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-1.js:1:45972)\n at B.\_failTransaction (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-0.js:1:225990)\n at B.approveTransaction (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-0.js:1:217659)\n at async B.updateAndApproveTransaction (chrome-extension://nkbihfbeogaeaoehlefnkodbefgpgknn/background-0.js:1:216902)"[[Prototype]]: ObjectsetTimeout (async) (anonymous) @ inpage.js:1write @ inpage.js:1  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
This is my javascript file calling the functions from both contracts.  
  
// Import libraries we need.import { default as Web3} from 'web3';import { default as contract } from 'truffle-contract'import voting\_artifacts from '../../build/contracts/Voting.json'import voting\_artifacts2 from '../../build/contracts/DeedToken.json'window.App = { start: function() { var self = this; self.transferNFT(); },transferNFT: function() { NFTContract.deployed().then(function(contractInstance) { let toAddress = $("#to-address").val(); let NFTid\_temp = $("#nft-id").val(); let NFTid = NFTid\_temp.substring(7); contractInstance.transferFrom(web3.currentProvider.selectedAddress, toAddress, NFTid, {gas: 140000, from: web3.eth.accounts[0]}); })}}window.addEventListener('load', async function() { if (window.ethereum) { await window.ethereum.send('eth\_requestAccounts'); window.web3 = new Web3(window.ethereum); } else { console.warn("No web3 detected."); window.web3 = new Web3(new Web3.providers.HttpProvider("http://127.0.0.1:8545")); } App.start();});  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Does this answer your question? Call a function in another contract - Solidity

**C2:**@Yilmaz, I'm using openzeppelin ERC20/721 standards. Inheriting ERC20 contract from ERC721 will have so many duplicate function clashes.

2 **Answer**

**A1:**You could use the interface if you do not want the full functionality of the contract.  
  
openzeppelin IERC20  
  
You copy full code in your project. Maybe create an interfaces directory. Then import it to your contract:  
  
import "../interfaces/IERC20.sol";  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
in order to call a contract method, you always need ABI. Since you imported the file into the contract, IERC20 will be available in your contract. In order to call transferFrom you need to pass 3 arguments. "from address", "to address", and amount. An example of how to use it will be like this  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
function stakeTokens(uint256 \_amount,address \_token) public{ // add require staments // IERC20(\_token) this will initialize the contract IERC20(\_token).transferFrom(msg.sender,address(this),\_amount); }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**thank you so much for your input. I took your advice and successfully compile and migrated. I found out that in order for user to trade erc20 tokens, the frontend should approve the use of erc20 tokens as showns below in javascript. The following is throwing an error, could you help me with ERC20 approve? Voting.deployed().then(function(votingcontractInstance) { votingcontractInstance.approve(contractInstance.address, votingcontractInstance.balanceOf(web3.eth.accounts[1]))});

**C2:**Voting is ABI of ERC20Token.sol

**C3:**@Nayana, please ask your next issue in another question with a clear explanation

**A2:**In order to interact with an ERC20 token, you have to create an instance of it from the desired contract. You would need to import ERC20 to your nfts contracts, and then create an ERC20 token instance pointing to your token. It would be something like this:  
  
// Inside the nfts contractERC20 token = ERC20("your token address here");  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
And then you will be able to interact with that token as:  
  
token.transferFrom("args");  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Hope you find this information useful :)

**C1:**Thank you for your answer. But I am still getting the same error. I've updated my code and error messages, I'd appreciate it if you could provide some other solutions.

**C2:**On a side note, I wonder if instantiating an ERC20 token would trigger its contructor again which mints erc20 tokens.

**C3:**Maybe you need to approve the transfer. A brand new instance of an ERC20 token is only created when calling it with the "new" keyword

**C4:**thank you so much for your input. Do you know any reference sources for this? I did find some people mentioning having users approve the transfer, but I am not quite sure what it means and how to go about implementing this.

**C5:**Basically, in order to let someone else to spend tokens from another's wallet, you need first to approve the spender. Because transferFrom function lets you transfer tokens from one account to another, without needing to be the owner of the account. Here you have further information: ethereum.stackexchange.com/questions/98892/…