# 251.https://stackoverflow.com/questions/72006961/problem-creating-nft-erc721-name-symbol

**T:**Problem creating NFT "erc721" (name, symbol)

**Q:**I am trying to create an NFT creation site (ERC721) where any user could upload an NFT he wants he just needs to fill in details of "name","symbol" and "url".I'm having a problem I can't change the "name" and "symbol". After the first creation each user who uploads NFT is given the "name" and "symbol" of the first user filled in. I use the library of "openzeppelin"  
  
// SPDX-License-Identifier: MITpragma experimental ABIEncoderV2;pragma solidity >=0.6.0 <0.8.0;import "../ERC721/ERC721.sol";contract newNFT is ERC721 { uint internal counterNFT; constructor () public{ counterNFT = 1; } function mint(string memory uri,string memory name,string memory symbol) public { name\_symbol(name,symbol); \_safeMint(msg.sender, counterNFT); \_setTokenURI(counterNFT, uri); counterNFT++; }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
\*\* openzeppelin \*\*  
  
 constructor () public { // register the supported interfaces to conform to ERC721 via ERC165 \_registerInterface(\_INTERFACE\_ID\_ERC721); \_registerInterface(\_INTERFACE\_ID\_ERC721\_METADATA); \_registerInterface(\_INTERFACE\_ID\_ERC721\_ENUMERABLE); } function name\_symbol(string memory nameNFT,string memory symbolNFT) internal virtual{ \_name = nameNFT; \_symbol = symbolNFT; }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
I want to create a smart contract with which any user can easily upload NFTBy "name" "symbol" and url he chooses. The problem is that the "name" and "symbol" do not change for me after the first user uploaded the NFT with his name and icon. Any user who comes after him and fills in the details will not change the "name" and "symbol" in "polygonscan" in the transaction.  
  
For example the first user on a site that uploads NFTname = dragonBallZsymbol = Dragonurl = wwwww.exmaple.co.il  
  
In the first example everything will work well you can see in polygonscan the "name" and "symbol" that the user has selected.  
  
Second user of the site:name = pokemonsymbol = PKurl = wwwww.exmaple.co.il  
  
In the second example after a user has filled in details entering polygonscan only the name and icon will still be as in the first example they will not change that way any subsequent user will get the name and icon of the first example.And this is the problem I can not solve.

2 **Answer**

**A1:**You are fully missunderstanding how name and symbol are supposed to work. They should identify the collection or the one NFT they will represent.  
  
Say for example Bored Apes: name="Bored Apes" and symbol = "BAYC".Then if I have the monkey smoking a cigar it's gonna be part of the metadata (the tokenURI) of that specific token ( for example Bored Ape #1344 with tokenURI="{smokes: true, smokeName: "cigar", hairColor:"black"}")  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
So you shouldn't be chaning name and symbol for every mint.

**C1:**I edited the post I hope my explanation is more understandable now

**A2:**I can not comment so sorry this must be an answer, I have no clue if you are creating an NFT or creating a new contract for said NFT with it with the goal of creating a Name and Symbol but if you are creating NFTs upon your contract you must have your solidity a function that sets a tokenURI and mints an nft with that tokenURI and then in your javascript create the metadata that then uploads to say IPFS and mints the NFT using that metadata, here is some code snippets if this is your goal  
  
Solidity Contract similar to this this one is simple and you can check eddrop.io to see how this one works  
  
contract EdDrop is ERC721, ERC721URIStorage, Ownable { using Counters for Counters.Counter; Counters.Counter private \_tokenIdCounter; constructor() ERC721("EdDrop", "EDN") {} //No Set Mint Cost, Educators just pay a Gas fee. function safeMint(address to, string memory uri) public { uint256 tokenId = \_tokenIdCounter.current(); \_tokenIdCounter.increment(); \_safeMint(to, tokenId); \_setTokenURI(tokenId, uri); } // The following functions are overrides required by Solidity. // function \_burn(uint256 tokenId) internal override(ERC721, ERC721URIStorage) { super.\_burn(tokenId); } //Sets the tokenURI the pinata link created from input on page. function tokenURI(uint256 tokenId) public view override(ERC721, ERC721URIStorage) returns (string memory) { return super.tokenURI(tokenId); }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Javascript to pass data to function  
  
 const quantityAsNumber = +quantity for (let i=1; i < quantity; ++i) { await mintNFT( name, instructor, platform, description, dates, content, skills, roster, external\_url, score, image, ); }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
The Actual Mint Function  
  
 export const mintNFT = async(name,instructor,platform, description, dates, content, skills, roster, external\_url, score, image, quanitity) => { //error handling //make metadata const metadata = new Object(); metadata.name = name; metadata.description = description; metadata.external\_url = external\_url; metadata.attributes = { instructor, platform, dates, content, skills, roster, score, } metadata.image= image; //pinata pin request const pinataResponse = await pinJSONToIPFS(metadata); console.log(pinataResponse) if (!pinataResponse.success) { return { success: false, status: "😢 Something went wrong while uploading your tokenURI.", } } const tokenURI = pinataResponse.pinataUrl; //load smart contract window.contract = new web3.eth.Contract(contractABI, contractAddress);//loadContract(); //set up your Ethereum transaction const transactionParameters = { to: contractAddress, // Required except during contract publications. from: window.ethereum.selectedAddress, // must match user's active address. 'data': window.contract.methods.safeMint(window.ethereum.selectedAddress, tokenURI).encodeABI() //make call to NFT smart contract };  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**I edited the post I hope my explanation is more understandable now

**C2:**@asalefalena I am sorry I don't believe I fully understand what you are trying to achieve, I can tell you that from what I understand all NFTs lying on one contract use the name and symbol belonging to that contract and you could change it with a function similar to what you are using but it will change for every NFT on that contract.

**C3:**What function give me details please?