# 433.https://stackoverflow.com/questions/71027923/how-to-retrieve-all-addresses-with-their-relevant-tokenids

**T:**How to retrieve all addresses with their relevant tokenIds

**Q:**I've got an NFT collection with a maxSupply of 1000.All the NFTs have been minted where totalSupply == maxSupply.  
  
What is the best way to retrieve all 1000 addresses along with their relevant tokenIds?  
  
I've tried the below, but the result is not working as expected. Where am I going wrong?  
  
Any help is very much appreciated!  
  
 mapping(uint256 => address) private getMintedAddress; function getAllAddresses() public view returns (address[] memory) { uint256 totalMinted = totalSupply(); address[] memory tokenAddresses = new address[](totalMinted); for (uint256 i = 1; i < totalMinted; i++) { tokenAddresses[i] = getMintedAddress[i]; } return tokenAddresses; }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**what error are you getting or why is the outcome is not as expected? Are you sure you are using the right i indexes? Can u share the full contract code

**C2:**I've updated the above code. It no longer returns an error. The only problem is that when the getAllAddresses() function is called, it returns this: 0: address[]: 0x0000000000000000000000000000000000000000 It doesn't return the actual address with the tokenIds.

1 **Answer**

**A1:**In solidity mappings, all values exist. In javascript, if we try to access a key that does not exist, objectName['keyThatDoesNotExist'] we get undefined. But in solidity, we get default value based on the values. If all values are strings, default value is empty string '', if all values are integers, default value is 0, if all values are booleans default value is false and if all values are address types, default values is 0x0000000000000000000000000000000000000000 which is the same as address(0);. That means getMintedAddress[i] does not exist so solidity is returning default address(0)  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
When you create an nft, update the  
  
// since i did not initalize default would be 0uint public indexForSenders;  
  
function createNFT() public returns(bytes32){ // setting id is important. easy way to start create a top level state // first item will start from 0: 0 -->address getMintedAddress[indexForSenders]=msg.sender; indexForSenders+=1 ..... }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**C1:**Thank you Yilmaz for answering my question. I'm still learning how all this works, and your answer was very well thought out. If that is the case, how do I properly map the getMintedAddress to the actual address that own these tokenIds?

**C2:**@JennyFromtheBlock you probably did not set the getMintedAddress. I mean did u add any item in your code

**C3:**Oh I see. All that needs to be done is set the address to the getMintedAddress. But I was wanting to retrieve the current address of a NFT holder. So if an address transfers an NFT to another address, I wanted to retrieve the current address. So like a snapshot.

**C4:**@JennyFromtheBlock that is a different story. You should ask a different question with related code and a detailed explanation.