# 4.https://stackoverflow.com/questions/67803090/how-to-get-erc-721-tokenid

**T:**How to get ERC-721 tokenID?

**Q:**I have created a ERC-721 contract deployed on ropston network.Using contract I'm creating NFT's and its totally working fine.  
  
Now for the transfer part I need to get tokenID of any NFT and transfer to to other address but I'm not able get the tokenID whenever I fetch transaction details from etherscan or using web3.  
  
I want to store the tokenID in DB so it can be utilized while transferring to other address.  
  
  
  
I have encircled the exact tokenID required in above image.  
  
I'm using following code:  
  
window.ethereum .request({ method: 'eth\_sendTransaction', params: [ { from: fromAddress, to: contractAddress, gas: '50000', data: nftContract.methods.transferFrom(fromAddress, toAddress, tokenNumber).encodeABI() }, ], })  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
I just want to get tokenID when NFT was created and store into DB for reference and perform business logic.  
  
function mintNFT(address recipient, string memory tokenURI) public onlyOwner returns (uint256) { \_tokenIds.increment(); uint256 newItemId = \_tokenIds.current(); \_mint(recipient, newItemId); \_setTokenURI(newItemId, tokenURI); return newItemId; }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Above is the solidity function responsible for creating the NFT.

**C1:**now i need to fetch newItemID and store into my DB as part of storing reference to my NFT to transfer any point later to other address.

3 **Answer**

**A1:**Your mintNFT() function doesn't emit any event containing the newItemId.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
solidity is using the standard definition of transfer  
  
There's no "standard definition", the ERC-721 standard only defines an interface and few other rules - and the actual implementation (of the interface) is on each developer. However I'm assuming that by the "standard definition" you mean the OpenZeppelin implementation, which is a widely used implementation of the ERC-721 standard and is used by many people who start coding in Solidity.  
  
You can see in the linked implementation, that the OZ \_mint() function emits the Transfer() event, where the 3rd argument is the minted token ID.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
So executing your mintNFT() function effectively emits the Transfer() event that contains the newly minted token ID as a value of the 3rd parameter.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
After you've executed the mintNFT() contract function from your JS code, it returns a PromiEvent object, that you can use to catch its receipt event.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
The receipt contains the emited logs, where you can find the Transfer() log as well.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
const tx = nftContract.methods.mintNFT(...).send({from: ...});tx.on('receipt', function(receipt){ console.log(receipt.logs[0].topics[3]); // this prints the hex value of the tokenId // you can use `web3.utils.hexToNumber()` to convert it to decimal});  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
If you want to get the token ID from an already existing transaction (using the tx hash), you can use this snippet:  
  
web3.eth.getTransactionReceipt('0x258a6d35445814d091ae67ec01cf60f87a4a58fa5ac1de25d0746edf8472f189').then(function(data){ let transaction = data; let logs = data.logs; console.log(logs); console.log(web3.utils.hexToNumber(logs[0].topics[3]));});  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
You can find more details in the web3 docs for the send() method and the receipt.

**C1:**Thankyou for this detailed information , please let me know the way forward or resource that particularly help to solve this issue. i got the idea whatever you described above but emitting event and getting there value on application side is kind of grey area for me without any documentation reference.

**C2:**@Omar I've updated my answer with an example of getting the event log data from a JS code (using web3).

**C3:**Thank you , I tried the solution but converting always gives 0 as a result which is of course not the tokenID im looking for : code that i tried : web3.eth.getTransactionReceipt('0x3216a1abc2a955c4323180a0d76a333631e823f39a1ebe82746aed8f9e8f9f73').then(function(data){ console.log(web3.utils.hexToNumber(data.logs[0].data)); })

**C4:**@Volatil3 The ERC-721 standard defines the Transfer() event with 3 indexed arguments. topics[1] is the address sender, topics[2] is the address recipient, and topics[3] is the uint256 token ID. (And topics[0] is the event signature)... Mind that this is the ERC-721 for NFTs. If you want a "regular" ERC-20 token, there's only 2 indexed topics.

**C5:**@TamásSengel Yes, thank you for pointing that out. I expected the code in the answer to be self-explanatory, but the actual code should be receipt.logs[0].topics[3]. Just updated the answer.

**A2:**You can try:  
  
const receipt = await web3.eth.getTransactionReceipt(hash)const tokenId = Web3.utils.hexToNumber(receipt.logs[0].topics[3])  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
I check hash from ropsten testnet:https://ropsten.etherscan.io/tx/0x59928012c3e0605b9346215c24654e84be29f2bf47949a2284aecf9991996a28  
  
and output is 11

**A3:**Try this if receipt.logs[0].topics[3] doesn't work:  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
receipt.events.Transfer.returnValues.tokenId  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]