# 448.https://stackoverflow.com/questions/70947724/erc721-minted-nft-not-showing-on-opensea-io-testnet

**T:**ERC721 Minted NFT not showing on Opensea.io testnet

**Q:**I run my code on rinkeby etherscan network and it works perfectly. But the image and descriptions are not showing on opensea testnet. I run the /validate/ url it shows Valid: "false".  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Here is what I found when I force update on opensea, https://testnets-api.opensea.io/api/v1/asset/0x668D179B933af761e4732B084290D32B3235C22b/0/?force\_update=true  
  
Here is my code:  
  
 // SPDX-License-Identifier: MITpragma solidity ^0.8.2;import "@openzeppelin/contracts/token/ERC721/extensions/ERC721URIStorage.sol";import "@openzeppelin/contracts/token/ERC721/extensions/ERC721Enumerable.sol";import "@openzeppelin/contracts/access/Ownable.sol";import "@openzeppelin/contracts/utils/Counters.sol";import "./ERC2981ContractWideRoyalties.sol";import "hardhat/console.sol";contract SmoothApe is ERC721URIStorage, Ownable, ERC2981ContractWideRoyalties { using Counters for Counters.Counter; using Strings for uint256; Counters.Counter private \_tokenIdCounter; event EmitNFT(address sender, uint256 tokenId); uint256 public preSaleTokenPrice = 0.01 ether; uint256 public costPrice = 0.02 ether; uint256 public maxSupply = 10000; uint256 public maxMintNft = 10; string public baseURI; string public baseExtension = ".json"; string public notRevealedUri; bool public paused = false; bool public revealed = true; bool public presale = false; // set 0x7350243981aB92E2A3646e377EBbFC28e9DE96C1 as payable admn wallet address payable public adminWallet = payable(0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2); address payable public communityWallet = payable(0x4B20993Bc481177ec7E8f571ceCaE8A9e22C02db); address payable public t1Wallet = payable(0x78731D3Ca6b7E34aC0F824c42a7cC18A495cabaB); address payable public t2Wallet = payable(0x617F2E2fD72FD9D5503197092aC168c91465E7f2); address payable public t3Wallet = payable(0x17F6AD8Ef982297579C203069C1DbfFE4348c372); // Mapping from token ID to owner address mapping(uint256 => address) private \_owners; constructor(string memory \_name, string memory \_symbol,string memory \_initBaseURI, string memory \_initNotRevealedUri) ERC721(\_name, \_symbol) { setBaseURI(\_initBaseURI); setNotRevealedUri(\_initNotRevealedUri); } // inherit and override erc165 function supportsInterface(bytes4 interfaceId) public view virtual override (ERC721, ERC2981Base) returns (bool) { return super.supportsInterface(interfaceId); } // set Royalty value (between 0 and 10000) function setRoyalties(address recipient, uint256 value) public onlyOwner { \_setRoyalties(recipient, value); } function \_baseURI() internal view override returns (string memory) { return baseURI; } function mintApe() public payable { uint256 tokenId = \_tokenIdCounter.current(); require(!paused, "Minting is paused"); require(msg.sender != owner(), "You can't mint ape to admin"); // require(\_mintAmount > 0, "Mint amount must be greater than 0"); // require(\_mintAmount <= maxSupply, "Mint amount must be less than or equal to max supply"); if(presale == true) { preSaleTokenPrice = costPrice; } require(msg.value >= costPrice, string(abi.encodePacked("Must send at least ", costPrice.toString(), " ether to purchase"))); require(balanceOf(msg.sender) < maxMintNft, "You can only own 11 ape at a time"); // require(\_mintAmount + balanceOf(msg.sender) <= maxMintNft, "You can only own 10 ape at a time"); // adminWallet.transfer(msg.value); payable(owner()).transfer(msg.value); emit EmitNFT(msg.sender, tokenId); // for(uint256 i = 1; i <= \_mintAmount; i++){ \_safeMint(msg.sender, tokenId); // \_setTokenURI(tokenId, \_baseURI()); \_tokenIdCounter.increment(); // } } // admin mint ape to wallet function mintApeTo(address \_to, uint256 \_mintAmount) public onlyOwner { require(!paused, "Minting is paused"); require(\_mintAmount > 0, "Mint amount must be greater than 0"); require(\_mintAmount <= maxSupply, "Mint amount must be less than or equal to max supply"); for(uint256 i = 1; i <= \_mintAmount; i++){ uint256 tokenId = \_tokenIdCounter.current(); \_safeMint(\_to, tokenId); // \_setTokenURI(tokenId, \_baseURI()); \_tokenIdCounter.increment(); } } function getCostPrice() public view virtual returns (uint256) { return costPrice; } // function to set cost of ape function setCost(uint256 \_preSaleTokenPrice, uint256 \_publicTokenPrice) public onlyOwner { preSaleTokenPrice = \_preSaleTokenPrice; costPrice = \_publicTokenPrice; } // set presale to true function setPresale(bool \_presale) public onlyOwner { presale = \_presale; } function reveal(bool \_reveal) public onlyOwner { revealed = \_reveal; } // pause function function pause() public onlyOwner { paused = true; } // set function for setBaseURI and setNotRevealed function function setBaseURI(string memory \_newBaseURI) public onlyOwner { baseURI = \_newBaseURI; } function setTokenURI(uint256 \_tokenId, string memory \_tokenURI) public onlyOwner { \_setTokenURI(\_tokenId, \_tokenURI); } // override tokenURI function function tokenURI(uint256 tokenId) public view virtual override returns (string memory) { require(\_exists(tokenId), "ERC721Metadata: URI query for nonexistent token"); require(tokenId < \_tokenIdCounter.current(), "Token ID must be less than the total supply"); if(!revealed) { return notRevealedUri; } string memory currentBaseURI = \_baseURI(); return bytes(currentBaseURI).length > 0 ? string( abi.encodePacked( currentBaseURI, tokenId.toString(), baseExtension)) : ""; }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Here is a sample of NFT i mint:https://testnets.opensea.io/assets/0x668D179B933af761e4732B084290D32B3235C22b/0  
  
my ipfs CID:ipfs://QmaoMZ19zhpC6T4id6jdBP1Qz5dQSmRZMkQZU7Zt8hyFNQ/

**C1:**They are showing correctly for me, sir

**C2:**I viewed it on another browser and it's working fine. I don't know what could have caused the problem but it's all fine now

2 **Answer**

**A1:**as you can see in the right up corner there is a "reload" button, you need to click it to reload the ipfs on opensea

**A2:**I had the same problem. Turned out that I wasn't uploading the images to ipfs correctly because it was visible only to me on my machine. So try to see if the ipfs metadata are correctly formated and are accesible from different devices.  
  
Also you shouldn't just paste your entire contract in. Tt is really hard to answer your question then. Just write out the important sections of the code.