# 504.https://stackoverflow.com/questions/70690702/how-to-display-nfts-of-torus-wallet

**T:**How to display NFTs of Torus wallet?

**Q:**I've successfully integrated the Torus wallet and now I am trying to display the NFTs that a person has in the wallet. I am doing it successfully with MetaMask but I am having troubles making it work with Torus. I am guessing I have a problem with the provider? I get the following errors on the browser:  
  
index.ts:225 Uncaught (in promise) Error: unsupported provider (argument="provider", value="[object Object]", code=INVALID\_ARGUMENT, version=providers/5.5.2) at Logger.makeError (index.ts:225) at Logger.throwError (index.ts:237) at Logger.throwArgumentError (index.ts:241) at new Web3Provider (web3-provider.ts:156) at loadNFTs (UsersNFTs.js:112)  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
and  
  
VM9765:2 Uncaught ReferenceError: process is not defined at Object.4043 (<anonymous>:2:13168) at r (<anonymous>:2:306599) at Object.8048 (<anonymous>:2:9496) at r (<anonymous>:2:306599) at Object.8641 (<anonymous>:2:1379) at r (<anonymous>:2:306599) at <anonymous>:2:315627 at <anonymous>:2:324225 at <anonymous>:2:324229 at HTMLIFrameElement.e.onload (index.js:1)  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
And this is my code:  
  
\*\*\* UsersNFTs.js \*\*\*  
  
 let [torusSdkInstance,setTorusSdkInstance] = useState() // Import dynamically torus wallet object useEffect(()=>{ const initialize = async () => { const torus = (await import("@toruslabs/torus-embed")).default; setTorusSdkInstance(new torus({})); } initialize(); }, [])const providerOptions = { "custom-Torus": { //Torus wallet display: { logo: 'https://miime.io/images/wallet-login-torus-logo.png', name: "Torus", description: "Connect to Torus Wallet" }, package: torusSdkInstance, options: { // apiKey: "EXAMPLE\_PROVIDER\_API\_KEY" }, connector: async (\_, options) => { await torusSdkInstance.init({ enableLogging: false, }); await torusSdkInstance.login(); const web3 = new Web3(torusSdkInstance.provider); return web3; } } }const web3Modal = new Web3Modal({ network: "mainnet", cacheProvider: true, providerOptions }); const connection = await web3Modal.connect() const provider = new ethers.providers.Web3Provider(connection) const signer = provider.getSigner() const marketContract = new ethers.Contract(nftmarketaddress, Market.abi, signer) const tokenContract = new ethers.Contract(nftaddress, NFT.abi, provider) const data = await marketContract.fetchMyNFTs()  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
\*\*\* fetchMyNFTs function in the smart contract \*\*\*  
  
 function fetchMyNFTs() public view returns (MarketItem[] memory) { uint totalItemCount = \_itemIds.current(); uint itemCount = 0; uint currentIndex = 0; for (uint i = 0; i < totalItemCount; i++) { if (idToMarketItem[i + 1].owner == msg.sender) { itemCount += 1; } } MarketItem[] memory items = new MarketItem[](itemCount); for (uint i = 0; i < totalItemCount; i++) { if (idToMarketItem[i + 1].owner == msg.sender) { uint currentId = idToMarketItem[i + 1].itemId; MarketItem storage currentItem = idToMarketItem[currentId]; items[currentIndex] = currentItem; currentIndex += 1; } } return items; }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

1 **Answer**

**A1:**I found the way to display the minted NFTs (not the purchased ones) of a wallet, so I want to share my solution!  
  
I have a fetchNFTs function in my smart contract that gets the tokens made available in the market place but not sold yet (address(0)):  
  
function fetchMarketItems() public view returns (MarketItem[] memory) {uint itemCount = \_itemIds.current();uint unsoldItemCount = \_itemIds.current() - \_itemsSold.current();uint currentIndex = 0;MarketItem[] memory items = new MarketItem[](unsoldItemCount);for (uint i = 0; i < itemCount; i++) { if (idToMarketItem[i + 1].owner == address(0)) { uint currentId = idToMarketItem[i + 1].itemId; MarketItem storage currentItem = idToMarketItem[currentId]; items[currentIndex] = currentItem; currentIndex += 1; }}return items;}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
And in the frontend after I call this function, I just had to map through the items and select the ones which the seller address equals the address of the connected wallet, so I get the NFTs that the address has minted but not sold yet.