# 368.https://stackoverflow.com/questions/71374833/react-contract-error-when-click-mint-button

**T:**React contract error when click Mint button

**Q:**I use Ropsten network for testing.Connecting with the Metamask wallet working well, but when I click the Mint button I get this error:  
  
Uncaught (in promise) Error: invalid address (argument="address", value=null, code=INVALID\_ARGUMENT, version=address/5.5.0) (argument="\_to", value=null, code=INVALID\_ARGUMENT, version=abi/5.0.7) at Logger.makeError (index.ts:225:1) at Logger.throwError (index.ts:237:1) at Logger.throwArgumentError (index.ts:241:1) at AddressCoder.\_throwError (abstract-coder.ts:68:1) at AddressCoder.encode (address.ts:18:1) at array.ts:71:1 at Array.forEach (<anonymous>) at pack (array.ts:54:1) at TupleCoder.encode (tuple.ts:23:1) at AbiCoder.encode (abi-coder.ts:106:1) at ABICoder.push../node\_modules/web3-eth-abi/lib/index.js.ABICoder.encodeParameters (index.js:120:1) at index.js:438:1 at Array.map (<anonymous>) at Object.\_encodeMethodABI (index.js:437:1) at Object.\_processExecuteArguments (index.js:700:1) at Object.\_executeMethod (index.js:719:1) at mint (App.js:655:1)  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Here is my ABI  
  
const ABI = [{ "inputs": [], "stateMutability": "nonpayable", "type": "constructor"},{ "anonymous": false, "inputs": [ { "indexed": true, "internalType": "address", "name": "owner", "type": "address" }, { "indexed": true, "internalType": "address", "name": "approved", "type": "address" }, { "indexed": true, "internalType": "uint256", "name": "tokenId", "type": "uint256" } ], "name": "Approval", "type": "event"},{ "anonymous": false, "inputs": [ { "indexed": true, "internalType": "address", "name": "owner", "type": "address" }, { "indexed": true, "internalType": "address", "name": "operator", "type": "address" }, { "indexed": false, "internalType": "bool", "name": "approved", "type": "bool" } ], "name": "ApprovalForAll", "type": "event"},{ "anonymous": false, "inputs": [ { "indexed": true, "internalType": "address", "name": "previousOwner", "type": "address" }, { "indexed": true, "internalType": "address", "name": "newOwner", "type": "address" } ], "name": "OwnershipTransferred", "type": "event"},{ "anonymous": false, "inputs": [ { "indexed": true, "internalType": "address", "name": "from", "type": "address" }, { "indexed": true, "internalType": "address", "name": "to", "type": "address" }, { "indexed": true, "internalType": "uint256", "name": "tokenId", "type": "uint256" } ], "name": "Transfer", "type": "event"},{ "inputs": [ { "internalType": "address", "name": "to", "type": "address" }, { "internalType": "uint256", "name": "tokenId", "type": "uint256" } ], "name": "approve", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [ { "internalType": "address", "name": "owner", "type": "address" } ], "name": "balanceOf", "outputs": [ { "internalType": "uint256", "name": "", "type": "uint256" } ], "stateMutability": "view", "type": "function"},{ "inputs": [], "name": "baseExtension", "outputs": [ { "internalType": "string", "name": "", "type": "string" } ], "stateMutability": "view", "type": "function"},{ "inputs": [], "name": "baseURI", "outputs": [ { "internalType": "string", "name": "", "type": "string" } ], "stateMutability": "view", "type": "function"},{ "inputs": [], "name": "cost", "outputs": [ { "internalType": "uint256", "name": "", "type": "uint256" } ], "stateMutability": "view", "type": "function"},{ "inputs": [ { "internalType": "uint256", "name": "tokenId", "type": "uint256" } ], "name": "getApproved", "outputs": [ { "internalType": "address", "name": "", "type": "address" } ], "stateMutability": "view", "type": "function"},{ "inputs": [ { "internalType": "address", "name": "owner", "type": "address" }, { "internalType": "address", "name": "operator", "type": "address" } ], "name": "isApprovedForAll", "outputs": [ { "internalType": "bool", "name": "", "type": "bool" } ], "stateMutability": "view", "type": "function"},{ "inputs": [], "name": "maxMintAmount", "outputs": [ { "internalType": "uint256", "name": "", "type": "uint256" } ], "stateMutability": "view", "type": "function"},{ "inputs": [], "name": "maxSupply", "outputs": [ { "internalType": "uint256", "name": "", "type": "uint256" } ], "stateMutability": "view", "type": "function"},{ "inputs": [ { "internalType": "address", "name": "\_to", "type": "address" }, { "internalType": "uint256", "name": "\_mintAmount", "type": "uint256" } ], "name": "mint", "outputs": [], "stateMutability": "payable", "type": "function"},{ "inputs": [], "name": "name", "outputs": [ { "internalType": "string", "name": "", "type": "string" } ], "stateMutability": "view", "type": "function"},{ "inputs": [], "name": "owner", "outputs": [ { "internalType": "address", "name": "", "type": "address" } ], "stateMutability": "view", "type": "function"},{ "inputs": [ { "internalType": "uint256", "name": "tokenId", "type": "uint256" } ], "name": "ownerOf", "outputs": [ { "internalType": "address", "name": "", "type": "address" } ], "stateMutability": "view", "type": "function"},{ "inputs": [ { "internalType": "bool", "name": "\_state", "type": "bool" } ], "name": "pause", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [], "name": "paused", "outputs": [ { "internalType": "bool", "name": "", "type": "bool" } ], "stateMutability": "view", "type": "function"},{ "inputs": [], "name": "renounceOwnership", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [ { "internalType": "address", "name": "from", "type": "address" }, { "internalType": "address", "name": "to", "type": "address" }, { "internalType": "uint256", "name": "tokenId", "type": "uint256" } ], "name": "safeTransferFrom", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [ { "internalType": "address", "name": "from", "type": "address" }, { "internalType": "address", "name": "to", "type": "address" }, { "internalType": "uint256", "name": "tokenId", "type": "uint256" }, { "internalType": "bytes", "name": "\_data", "type": "bytes" } ], "name": "safeTransferFrom", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [ { "internalType": "address", "name": "operator", "type": "address" }, { "internalType": "bool", "name": "approved", "type": "bool" } ], "name": "setApprovalForAll", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [ { "internalType": "string", "name": "\_newBaseExtension", "type": "string" } ], "name": "setBaseExtension", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [ { "internalType": "string", "name": "\_newBaseURI", "type": "string" } ], "name": "setBaseURI", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [ { "internalType": "uint256", "name": "\_newmaxMintAmount", "type": "uint256" } ], "name": "setmaxMintAmount", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [ { "internalType": "bytes4", "name": "interfaceId", "type": "bytes4" } ], "name": "supportsInterface", "outputs": [ { "internalType": "bool", "name": "", "type": "bool" } ], "stateMutability": "view", "type": "function"},{ "inputs": [], "name": "symbol", "outputs": [ { "internalType": "string", "name": "", "type": "string" } ], "stateMutability": "view", "type": "function"},{ "inputs": [ { "internalType": "uint256", "name": "index", "type": "uint256" } ], "name": "tokenByIndex", "outputs": [ { "internalType": "uint256", "name": "", "type": "uint256" } ], "stateMutability": "view", "type": "function"},{ "inputs": [ { "internalType": "address", "name": "owner", "type": "address" }, { "internalType": "uint256", "name": "index", "type": "uint256" } ], "name": "tokenOfOwnerByIndex", "outputs": [ { "internalType": "uint256", "name": "", "type": "uint256" } ], "stateMutability": "view", "type": "function"},{ "inputs": [ { "internalType": "uint256", "name": "tokenId", "type": "uint256" } ], "name": "tokenURI", "outputs": [ { "internalType": "string", "name": "", "type": "string" } ], "stateMutability": "view", "type": "function"},{ "inputs": [], "name": "totalSupply", "outputs": [ { "internalType": "uint256", "name": "", "type": "uint256" } ], "stateMutability": "view", "type": "function"},{ "inputs": [ { "internalType": "address", "name": "from", "type": "address" }, { "internalType": "address", "name": "to", "type": "address" }, { "internalType": "uint256", "name": "tokenId", "type": "uint256" } ], "name": "transferFrom", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [ { "internalType": "address", "name": "newOwner", "type": "address" } ], "name": "transferOwnership", "outputs": [], "stateMutability": "nonpayable", "type": "function"},{ "inputs": [ { "internalType": "address", "name": "\_owner", "type": "address" } ], "name": "walletOfOwner", "outputs": [ { "internalType": "uint256[]", "name": "", "type": "uint256[]" } ], "stateMutability": "view", "type": "function"},{ "inputs": [], "name": "withdraw", "outputs": [], "stateMutability": "payable", "type": "function"}]  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Here is my JS  
  
var acc = null;var contract = null;const ADDRESS = "0x00000000000000000000000000000000"; // hidding async function connwallet(){ if(window.ethereum){ var web3 = new Web3(window.ethereum); await window.ethereum.send('eth\_requestAccounts'); var accounts = await web3.eth.getAccounts(); var acc = accounts[0]; document.getElementById('wallet-address').textContent = acc; contract = new web3.eth.Contract(ABI, ADDRESS); }}async function mint(){ if(window.ethereum){ console.log("IF"); var \_mintAmount = Number(document.querySelector("[name=amount]").value); var mintRate = Number(await contract.methods.cost().call()); var totalAmount = mintRate \* \_mintAmount; contract.methods.mint(acc, \_mintAmount).send({from: acc, value: String(totalAmount)}).then((res) => { console.log(res); }).catch((err) => { console.log(err); }); }} function App() { return ( <div className="App"> <div className='container'> <div className='row'> <form class="gradient col-lg-5 mt-5" style={{borderRadius:"25px",boxShadow:"1px 1px 15px #000000"}}> <h4 style={{color:"#FFFFFF"}}>Mint Portal</h4> <h5 style={{color:"#FFFFFF"}}>Please connect your wallet</h5> <Button onClick={connwallet} style={{marginBottom:"5px", color:"#FFFFFF"}}>Connect Wallet</Button> <div class="card" id='wallet-address' style={{marginTop:"3px", boxShadow:"1px 1px 4px #000000"}}> <label for="floatingInput">Wallet Address</label> </div> <div class="card" style={{marginTop:"3px", boxShadow:"1px 1px 4px #000000"}}> <input type="number" name="amount" defaultValue="1" min="1" max="5"/> <label>Please select the amount of NFTs to mint.</label> <Button onClick={mint}>Mint</Button> </div> <label> Price 0.01 ETH </label> </form> </div> </div> </div> );}export default App;  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
What versions I have:  
  
"bootstrap": "^5.1.3","ethers": "^5.5.4","react": "^17.0.2","react-bootstrap": "^2.1.2","react-dom": "^17.0.2","react-scripts": "5.0.0","stream": "^0.0.2","web-vitals": "^2.1.4","web3": "^1.7.0","webpack": "^4.46.0"  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
I tried lower versions of react-script, stream, web3 and webpack but the same error.The error is in line 655, so it mean here:  
  
contract.methods.mint(acc, \_mintAmount).send({from: acc, value: String(totalAmount)}).then((res) => {  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
Please help

0 **Answer**