# 589.https://stackoverflow.com/questions/69541323/continuously-listening-to-smart-contract-events

**T:**Continuously listening to smart contract events

**Q:**My NFT project architecture in such way that only when a minting event has occurred, a minted event is emitted and then I want to listen to that event so I trigger an upload of the JSON content that would be then visible to token owner (since anyone can potentially access the content of all the token by basically going to baseURI/tokenID and I want to keep the rarity of the tokens a secret until minted).  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
I want to create a function that continuously listens to new events from the minted event. I'm trying to access new events by using the following implementation, because I couldn't find an implementation that continuously listens to events from the contract.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
async function listener() { events = nftContract.getPastEvents("minted",{fromBlock: 1}, function(err,res) { console.log(res); })  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
What is the best way to create a function that continues to poll events from the contract?

**C1:**I refer you to my answer here: Getting all newly minted ERC721 contracts with an ethereum node

1 **Answer**

**A1:**The full code of the solution is so:  
  
const ethers = require('ethers');const CONTRACT\_ADDRESS = "0x10820dB......";const ABIJSON = JSON.parse('{"\_format": "hh-sol-artifact-1", "contractName": "testcontract", "....}')provider = new ethers.providers.AlchemyProvider("ropsten");const contract = new ethers.Contract(CONTRACT\_ADDRESS, ABIJSON.abi, provider);contract.on("eventName", ( caller,tokenID) => { //this section is called every time an event is emitted })  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]