# 510.https://stackoverflow.com/questions/70650791/how-to-connect-a-erc-721-smart-contract-with-another-smart-contract

**T:**How to connect a ERC-721 smart contract with another smart contract

**Q:**I was wondering if it is possibile to connect the ERC-721 contract with another smart contract. Based on the data included into the smart contract, I'd like the ERC-721 contract to automatically mint and deliver the nft. For this reason, I was wondering if it is possibile for the ERC-721 contract to connect and retrieve the specific data from the smart contract, like some sort of oracle.  
  
I am new to programming, so thanks in advance.  
  
Giulia

2 **Answer**

**A1:**The action always needs to originate from a transaction - e.g. to the other contract. So the NFT contract can't just react to any situation. But apart from that, it's possible.  
  
// deployed on address 0x123contract OtherContract { function mintNFT() external { // invoke the NFT's function `mint()` NFT(0x456).mint(); }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
// deployed on address 0x456contract NFT { function mint() external { // only executable from the `OtherContract` address require(msg.sender == address(0x123)); }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
When the user executes the OtherContract.mintNFT() function, it effectively invokes the NFT.mint() as well.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]

**A2:**Last I've checked, in order for your contract to receive an ERC721 token your contract needs to include an onERC721Received function.  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
import "@openzeppelin/contracts/token/ERC721/IERC721Receiver.sol";function onERC721Received( address, address, uint256, bytes memory ) external view override returns (bytes4) { //additional logic (optional) return IERC721Receiver.onERC721Received.selector; }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
https://docs.openzeppelin.com/contracts/4.x/api/token/erc721#IERC721Receiver  
  
I don't really think it is necessary to use an oracle to get data that's already stored on chain.  
  
As far as accessing data from that smart contract, there are multiple way of going about it.  
  
You could import the contract, initialize it, and access its functions via an interface, such as:  
  
import "@openzeppelin/contracts/token/ERC721/IERC721.sol";constructor(address \_nft) { nft = IERC721(\_nft);}function thenYouCould() public { nft.functionName(parameters);  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
or you could use the external call method with the abi.encodedSignature, such as:  
  
(bool success, bytes memory data) = contractAddress.call{/\*optional values\*/ value: msg.value, gas: 5000}( abi.encodeWithSignature("functionName(string,uint256)", "call foo", 123)  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
https://solidity-by-example.org/call/