# 2.https://stackoverflow.com/questions/72036381/deploy-erc721-contracts-from-a-simple-or-factory-contract

**T:**deploy erc721 contracts from a simple or factory contract

**Q:**I want to deploy erc721 contracts from a simple factory contract and i have this error when I try to upload NFT - "transfer to non ERC721Receiver implementer", "data"enter image description here  
  
// SPDX-License-Identifier: MITpragma solidity >=0.6.0 <0.8.0;import "./ezeynftFactory.sol";contract ezeNFT { uint256 public tokenCounter; constructor(){ tokenCounter = 201;} function \_mintNewNFT( string memory name, string memory symbol, string memory tokenUri)public{ uint256 newTokenId = tokenCounter; ezeynftFactory nfts = new ezeynftFactory(name,symbol,tokenUri,newTokenId); tokenCounter += 1; } }  
  
// SPDX-License-Identifier: MITpragma experimental ABIEncoderV2;pragma solidity >=0.6.0 <0.8.0;import "../ERC721/ERC721.sol";contract ezeynftFactory is ERC721 { constructor(string memory name, string memory symbol,string memory tokenURI,uint tokenID) ERC721(name,symbol) { \_safeMint(msg.sender, tokenID); \_setTokenURI(tokenID,tokenURI); }}

1 **Answer**

**A1:**The OpenZeppelin \_safeMint() function tries to invoke onERC721Received() on the receiver if the receiver is a contract (which your ezeNFT is). And reverts the transaction if it doesn't receive the expected response (which doesn't receive, so it reverts).  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
There are two solutions to your problem.  
  
 ● Either implement the onERC721Received() function on your ezeNFT contract (the token receiver), returning the expected response. See ERC721TokenReceiver interface in the ERC-721 standard definition for more info.  
  
contract ezeNFT { uint256 public tokenCounter; constructor(){ tokenCounter = 201; } function \_mintNewNFT( string memory name, string memory symbol, string memory tokenUri) public { uint256 newTokenId = tokenCounter; ezeynftFactory nfts = new ezeynftFactory(name,symbol,tokenUri,newTokenId); tokenCounter += 1; } function onERC721Received(address \_operator, address \_from, uint256 \_tokenId, bytes \_data) external returns(bytes4) { // TODO validate if you want to accept tokens only from certain collections // return the expected response return bytes4(keccak256("onERC721Received(address,address,uint256,bytes)")); }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
 ● Or use the \_mint() function, instead of \_safeMint(), which doesn't perform the onERC721Received() call on the token receiver. So you won't have to implement it on ezeNFT.  
  
contract ezeynftFactory is ERC721 { constructor(string memory name, string memory symbol,string memory tokenURI,uint tokenID) ERC721(name,symbol) { \_mint(msg.sender, tokenID); // `\_mint()` instead of `\_safeMint()` \_setTokenURI(tokenID,tokenURI); }}  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]