# 108.https://stackoverflow.com/questions/72853874/trying-to-implement-erc1155openseagassless-but-not-able-to-understand-the-struct

**T:**Trying to implement ERC1155OpenSeaGassLess but not able to understand the structure of the contract

**Q:**// SPDX-License-Identifier: MIT pragma solidity ^0.8.13; import "@openzeppelin/contracts/token/ERC1155/ERC1155.sol"; abstract contract ERC1155OpenSeaGassLess is ERC1155 { address public \_openSeaProxyRegistryAddress; constructor(address openSeaProxyRegistryAddress\_) { \_openSeaProxyRegistryAddress = openSeaProxyRegistryAddress\_; } function \_setOpenSeaProxyRegistryAddress(address openSeaProxyRegistryAddress\_) internal virtual { \_openSeaProxyRegistryAddress = openSeaProxyRegistryAddress\_; } function isApprovedForAll(address \_owner, address operator) public view virtual override returns (bool) { if (\_openSeaProxyRegistryAddress != address(0)) { \*\*OpenSeaProxyRegistry proxyRegistry = OpenSeaProxyRegistry(\_openSeaProxyRegistryAddress); if (address(proxyRegistry.proxies(\_owner)) == operator)\*\* { return true; } } return super.isApprovedForAll(\_owner, operator); } } contract OwnableDelegateProxy {} contract OpenSeaProxyRegistry { mapping(address => OwnableDelegateProxy) public proxies; }  
  
WARN: THIS PARAGRAPH CONTAINS TAG: [CODE]   
  
in the above code can someone explain me a few things  
  
1.  
  
OpenSeaProxyRegistry proxyRegistry = OpenSeaProxyRegistry(\_openSeaProxyRegistryAddress);if (address(proxyRegistry.proxies(\_owner)) == operator)  
  
what this part is trying to do...  
  
2.  
  
contract OwnableDelegateProxy {}  
  
what does this contract do?  
  
3.  
  
contract OpenSeaProxyRegistry {mapping(address => OwnableDelegateProxy) public proxies;}  
  
what does this contract do?  
  
will be highly thankful if someone explains the full contract flow

0 **Answer**