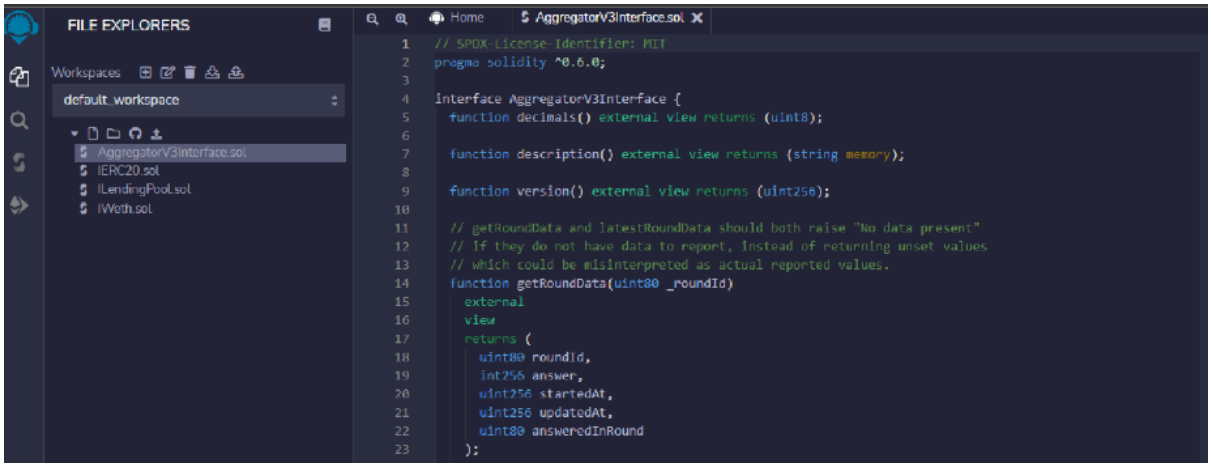
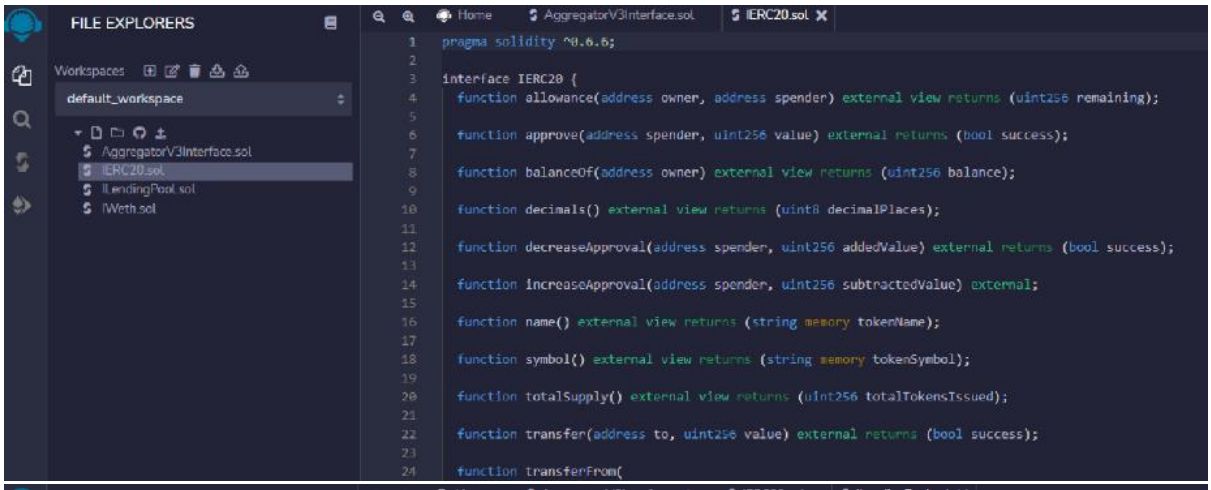
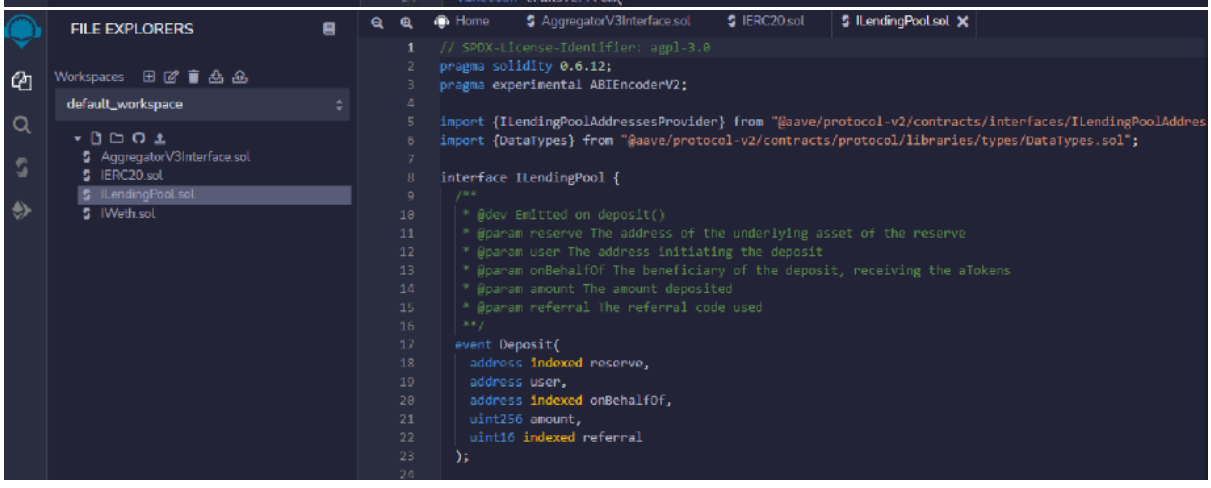

FILE EXPLORERS
Workspaces
default_workspace
AggregatorV3Interface.sol
IERC20.sol
ILendingPool.sol
IWeth.sol
Home
AggregatorV3Interface.sol
// SPDX-License-Identifier: MIT
pragma solidity ^0.6.0;
interface AggregatorV3Interface {
function decimals() external view returns (uint8);
function description() external view returns (string memory);
function version() external view returns (uint256);
// getRoundData and latestRoundData should both raise "No data present"
// if they do not have data to report, instead of returning unset values
// which could be misinterpreted as actual reported values.
function getRoundData(uint80 _roundId)
external
view
returns (
uint80 roundId,
int256 answer,
uint256 startedAt,
uint256 updatedAt,
uint80 answeredInRound
);


FILE EXPLORERS
Workspaces
default_workspace
AggregatorV3Interface.sol
IERC20.sol
ILendingPool.sol
IWeth.sol
Home
AggregatorV3Interface.sol
IERC20.sol
pragma solidity ^0.6.6;
interface IERC20 {
function allowance(address owner, address spender) external view returns (uint256 remaining);
function approve(address spender, uint256 value) external returns (bool success);
function balanceOf(address owner) external view returns (uint256 balance);
function decimals() external view returns (uint8 decimalPlaces);
function decreaseApproval(address spender, uint256 addedValue) external returns (bool success);
function increaseApproval(address spender, uint256 subtractedValue) external;
function name() external view returns (string memory tokenName);
function symbol() external view returns (string memory tokenSymbol);
function totalSupply() external view returns (uint256 totalTokensIssued);
function transfer(address to, uint256 value) external returns (bool success);
function transferFrom(


FILE EXPLORERS
Workspaces
default_workspace
AggregatorV3Interface.sol
IERC20.sol
ILendingPool.sol
IWeth.sol
Home
AggregatorV3Interface.sol
IERC20.sol
ILendingPool.sol
// SPDX-License-Identifier: agpl-3.0
pragma solidity 0.6.12;
pragma experimental ABIEncoderV2;
import {ILendingPoolAddressesProvider} from "@aave/protocol-v2/contracts/interfaces/ILendingPoolAddres
import {DataTypes} from "@aave/protocol-v2/contracts/protocol/libraries/types/DataTypes.sol";
interface ILendingPool {
/**
* @dev Emitted on deposit()
* @param reserve The address of the underlying asset of the reserve
* @param user The address initiating the deposit
* @param onBehalfOf The beneficiary of the deposit, receiving the aTokens
* @param amount The amount deposited
* @param referral The referral code used
**/
event Deposit(
address indexed reserve,
address user,
address indexed onBehalfOf,
uint256 amount,
uint16 indexed referral
);