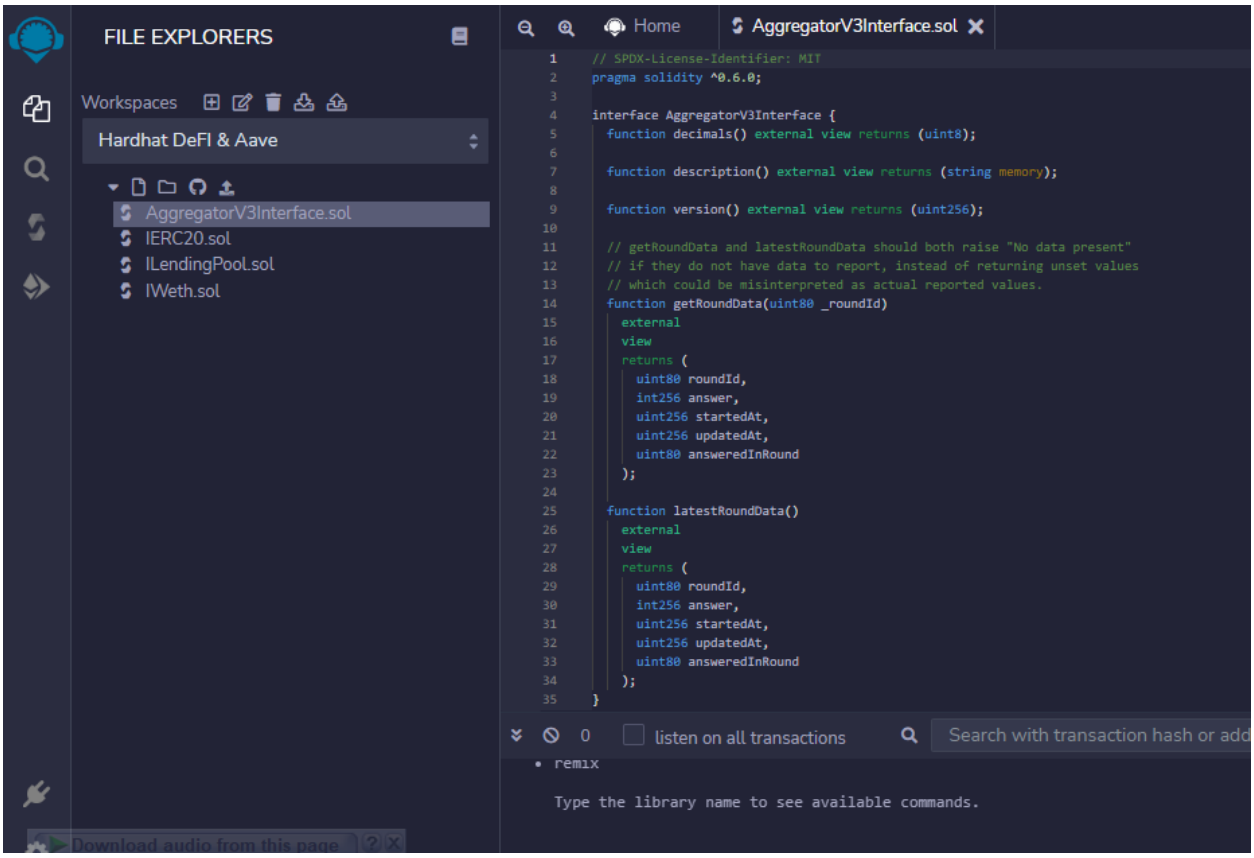
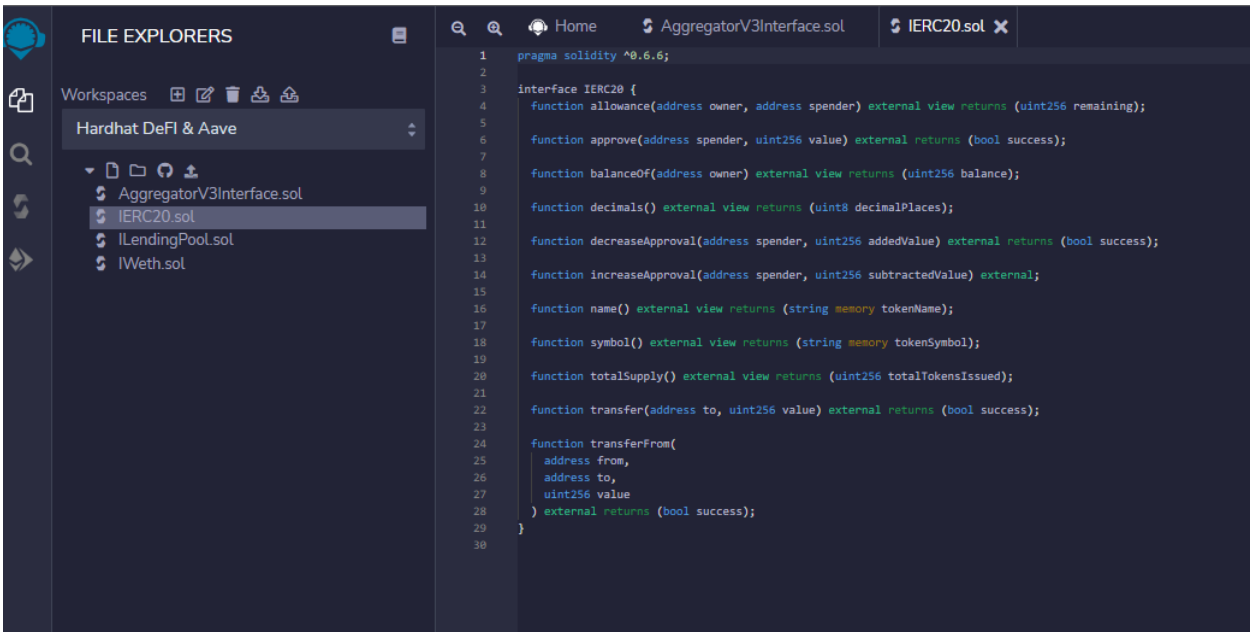
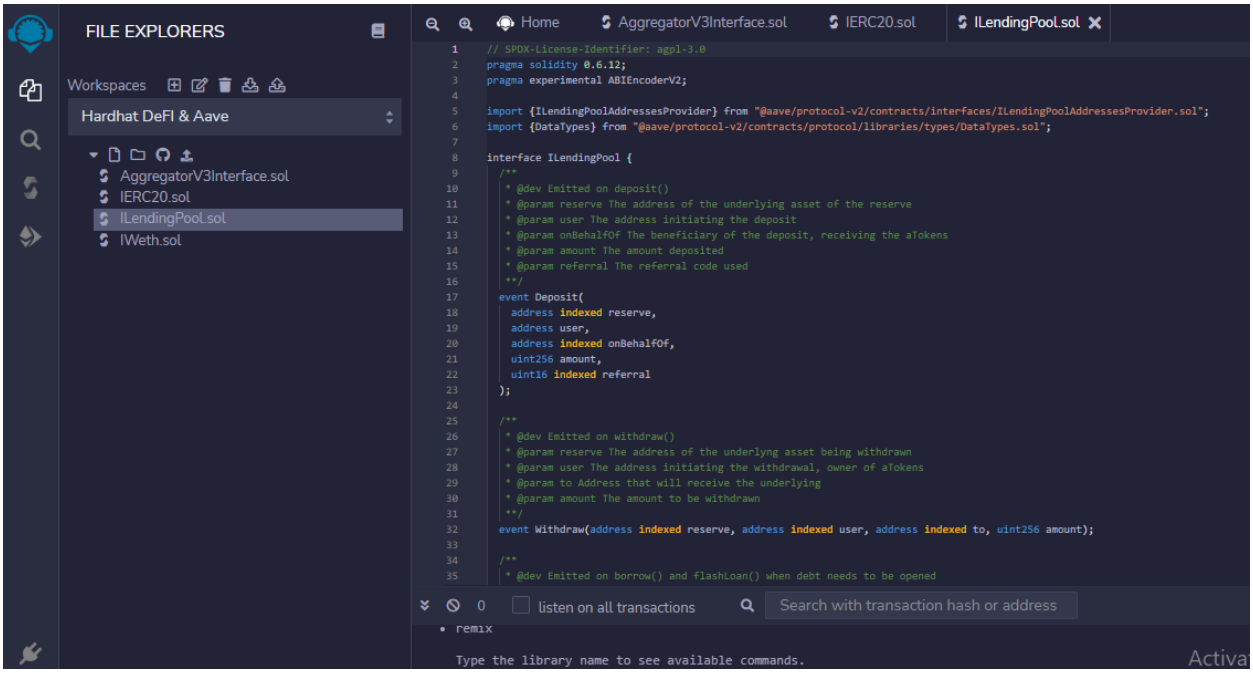
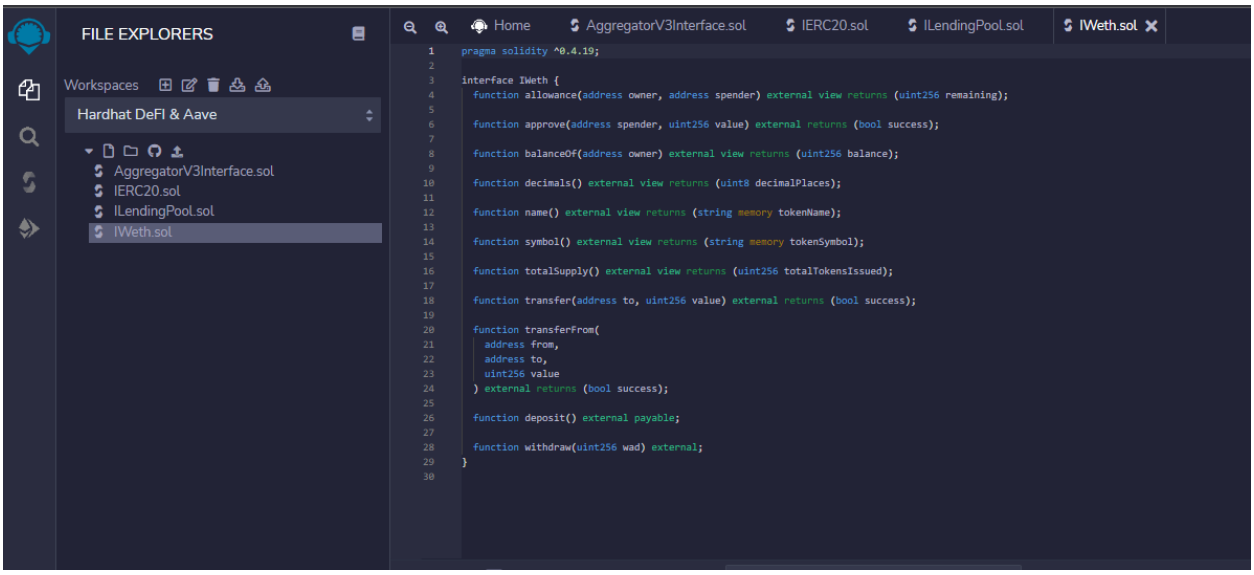
Hardhat DeFI & Aave






FILE EXPLORERS
Workspaces
Hardhat DeFI & Aave
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/ILendingPoolAddressesProvider.sol";
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
);
/**
* @dev Emitted on withdraw()
* @param reserve The address of the underlyng asset being withdrawn
* @param user The address initiating the withdrawal, owner of aTokens
* @param to Address that will receive the underlying
* @param amount The amount to be withdrawn
**/
event Withdraw(address indexed reserve, address indexed user, address indexed to, uint256 amount);
/**
* @dev Emitted on borrow() and flashLoan() when debt needs to be opened
0
listen on all transactions
Search with transaction hash or address
remix
Type the library name to see available commands.


FILE EXPLORERS
Workspaces
Hardhat DeFI & Aave
AggregatorV3Interface.sol
IERC20.sol
ILendingPool.sol
IWeth.sol
Home
AggregatorV3Interface.sol
IERC20.sol
ILendingPool.sol
IWeth.sol
pragma solidity ^0.4.19;
interface IWeth {
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 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(
address from,
address to,
uint256 value
) external returns (bool success);
function deposit() external payable;
function withdraw(uint256 wad) external;
}