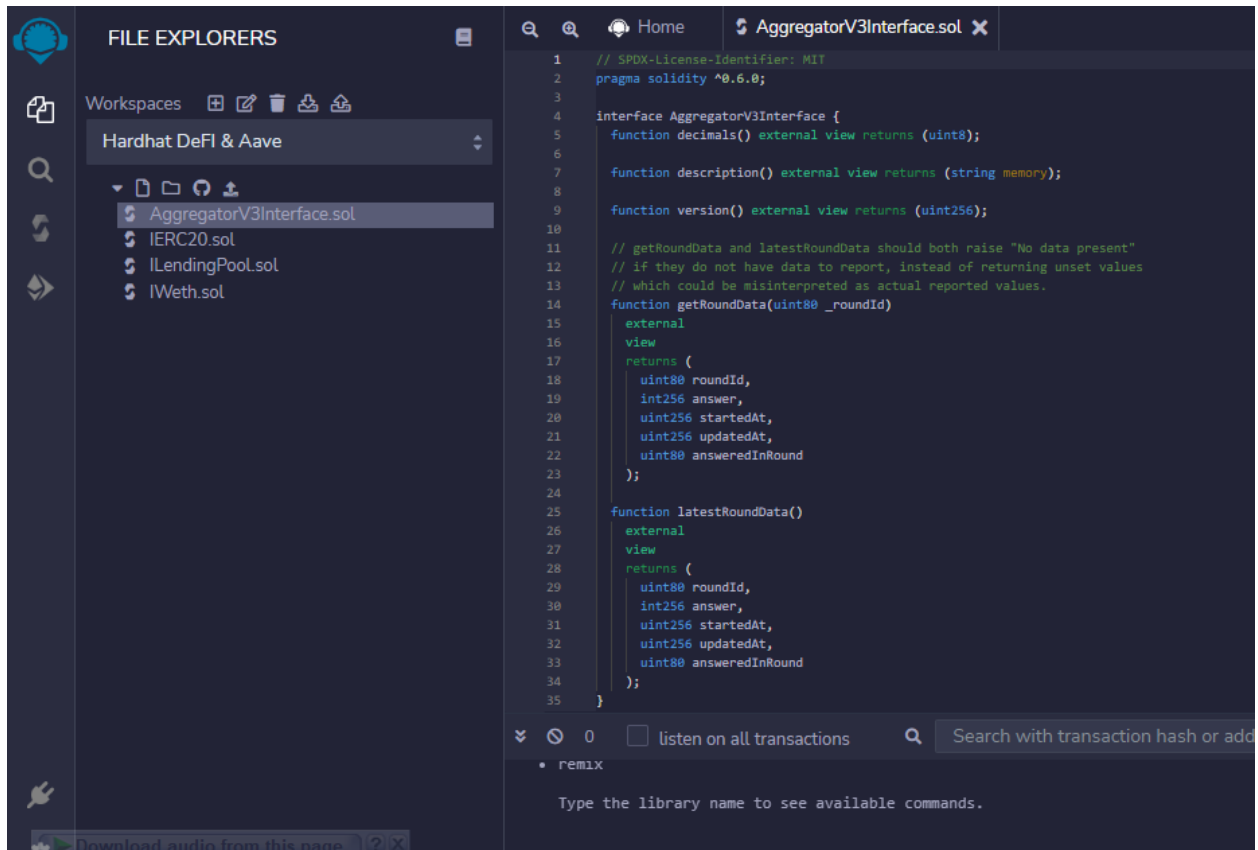
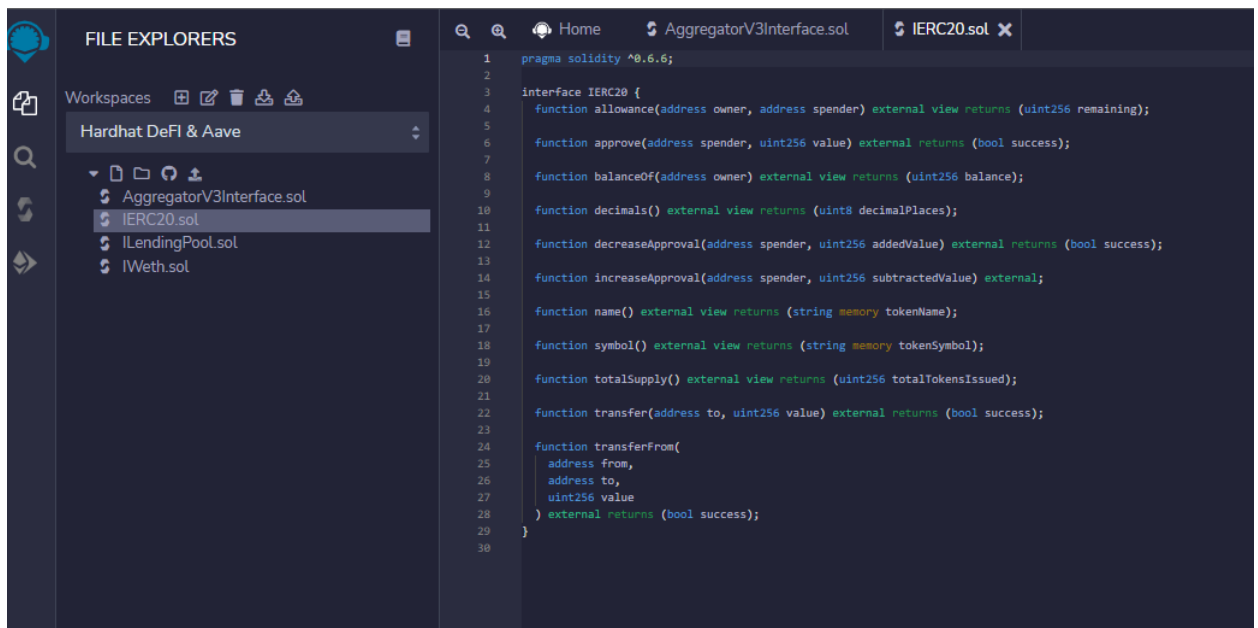


Hardhat DeFI & Aave



The screenshot shows the Hardhat IDE interface. On the left, the 'FILE EXPLORERS' panel displays a workspace named 'Hardhat DeFI & Aave' containing four files: 'AggregatorV3Interface.sol', 'IERC20.sol', 'ILendingPool.sol', and 'IWeth.sol'. The 'AggregatorV3Interface.sol' file is selected and its content is displayed in the main editor. The code defines an interface for an aggregator, including functions for decimals, description, version, and retrieving round data. The bottom panel shows the 'remix' command prompt with the text 'Type the library name to see available commands.'






```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.6.0;
3
4 interface AggregatorV3Interface {
5     function decimals() external view returns (uint8);
6
7     function description() external view returns (string memory);
8
9     function version() external view returns (uint256);
10
11     // getRoundData and latestRoundData should both raise "No data present"
12     // if they do not have data to report, instead of returning unset values
13     // which could be misinterpreted as actual reported values.
14     function getRoundData(uint80 _roundId)
15         external
16         view
17         returns (
18             uint80 roundId,
19             int256 answer,
20             uint256 startedAt,
21             uint256 updatedAt,
22             uint80 answeredInRound
23         );
24
25     function latestRoundData()
26         external
27         view
28         returns (
29             uint80 roundId,
30             int256 answer,
31             uint256 startedAt,
32             uint256 updatedAt,
33             uint80 answeredInRound
34         );
35 }
```



The screenshot shows the Hardhat IDE interface with the 'IERC20.sol' file selected in the 'FILE EXPLORERS' panel. The main editor displays the code for the IERC20 interface, which defines standard ERC20 token functions. The bottom panel shows the 'remix' command prompt.

```
1 pragma solidity ^0.6.6;
2
3 interface IERC20 {
4     function allowance(address owner, address spender) external view returns (uint256 remaining);
5
6     function approve(address spender, uint256 value) external returns (bool success);
7
8     function balanceOf(address owner) external view returns (uint256 balance);
9
10    function decimals() external view returns (uint8 decimalPlaces);
11
12    function decreaseApproval(address spender, uint256 addedValue) external returns (bool success);
13
14    function increaseApproval(address spender, uint256 subtractedValue) external;
15
16    function name() external view returns (string memory tokenName);
17
18    function symbol() external view returns (string memory tokenSymbol);
19
20    function totalSupply() external view returns (uint256 totalTokensIssued);
21
22    function transfer(address to, uint256 value) external returns (bool success);
23
24    function transferFrom(
25        address from,
26        address to,
27        uint256 value
28    ) external returns (bool success);
29 }
30
```

FILE EXPLORERS

Workspaces     

Hardhat DeFi & Aave

- AggregatorV3Interface.sol
- IERC20.sol
- ILendingPool.sol
- IWeth.sol

Home AggregatorV3Interface.sol IERC20.sol ILendingPool.sol

```
1 // SPDX-License-Identifier: agpl-3.0
2 pragma solidity 0.6.12;
3 pragma experimental ABIEncoderV2;
4
5 import {ILendingPoolAddressesProvider} from "@aave/protocol-v2/contracts/interfaces/ILendingPoolAddressesProvider.sol";
6 import {DataTypes} from "@aave/protocol-v2/contracts/protocol/libraries/types/DataTypes.sol";
7
8 interface ILendingPool {
9     /**
10      * @dev Emitted on deposit()
11      * @param reserve The address of the underlying asset of the reserve
12      * @param user The address initiating the deposit
13      * @param onBehalfOf The beneficiary of the deposit, receiving the aTokens
14      * @param amount The amount deposited
15      * @param referral The referral code used
16      */
17     event Deposit(
18         address indexed reserve,
19         address user,
20         address indexed onBehalfOf,
21         uint256 amount,
22         uint16 indexed referral
23     );
24
25     /**
26      * @dev Emitted on withdraw()
27      * @param reserve The address of the underlying asset being withdrawn
28      * @param user The address initiating the withdrawal, owner of aTokens
29      * @param to Address that will receive the underlying
30      * @param amount The amount to be withdrawn
31      */
32     event Withdraw(address indexed reserve, address indexed user, address indexed to, uint256 amount);
33
34     /**
35      * @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.

Activa

FILE EXPLORERS

Workspaces     

Hardhat DeFi & Aave

- AggregatorV3Interface.sol
- IERC20.sol
- ILendingPool.sol
- IWeth.sol

Home AggregatorV3Interface.sol IERC20.sol ILendingPool.sol IWeth.sol

```
1 pragma solidity ^0.4.19;
2
3 interface IMeth {
4     function allowance(address owner, address spender) external view returns (uint256 remaining);
5
6     function approve(address spender, uint256 value) external returns (bool success);
7
8     function balanceOf(address owner) external view returns (uint256 balance);
9
10    function decimals() external view returns (uint8 decimalPlaces);
11
12    function name() external view returns (string memory tokenName);
13
14    function symbol() external view returns (string memory tokenSymbol);
15
16    function totalSupply() external view returns (uint256 totalTokensIssued);
17
18    function transfer(address to, uint256 value) external returns (bool success);
19
20    function transferFrom(
21        address from,
22        address to,
23        uint256 value
24    ) external returns (bool success);
25
26    function deposit() external payable;
27
28    function withdraw(uint256 wad) external;
29 }
30
```