

Hardhat ERC20s

The image displays three sequential screenshots of a code editor, likely VS Code, showing the development of a Hardhat ERC20 token contract. Each screenshot includes a sidebar with a 'FILE EXPLORERS' panel and a main editor window.

Top Screenshot: The editor shows the initial setup of the contract. It includes a license header, a pragma statement for Solidity 0.8.7, and an interface for a token recipient. The contract itself is named `TokenERC20` and includes public variables for `name`, `symbol`, `decimals` (set to 18), and `totalSupply`. It also features mappings for `balanceOf` and `allowance`.

Middle Screenshot: This view shows the addition of public events. The `Transfer` event is defined with `address indexed from`, `address indexed to`, and `uint256 value`. The `Approval` event is defined with `address indexed _owner`, `address indexed _spender`, and `uint256 _value`. A `Burn` event is also added with `address indexed from` and `uint256 value`.

Bottom Screenshot: This view shows the implementation of the `constructor` and the `_transfer` function. The `constructor` takes `uint256 initialSupply`, `string memory tokenName`, and `string memory tokenSymbol` as arguments. It calculates the total supply based on the decimal amount and initializes the `balanceOf` for the creator. The `_transfer` function is marked as internal and includes checks to prevent transfers to the zero address, ensure the sender has enough balance, and prevent overflows.

