

**Code:**

```
// SPDX-License-Identifier: MIT

pragma solidity ^0.8.0;

contract EmbeddedWallet {

    event Deposit(address indexed user, uint amount);

    event Withdrawal(address indexed user, uint amount);

    event Transfer(address indexed from, address indexed to, uint amount);

    mapping(address => uint) public balances;

    modifier hasEnoughFunds(uint amount) {

        require(balances[msg.sender] >= amount, "Insufficient funds");

        _;

    }

    function deposit() public payable {

        require(msg.value > 0, "Deposit must be greater than 0");

        balances[msg.sender] += msg.value;

        emit Deposit(msg.sender, msg.value);

    }

    function withdraw(uint amount) public hasEnoughFunds(amount) {

        balances[msg.sender] -= amount;

        payable(msg.sender).transfer(amount);

        emit Withdrawal(msg.sender, amount);

    }

    function transfer(address to, uint amount) public hasEnoughFunds(amount) {

        require(to != address(0), "Invalid recipient address");
```

```

require(to != msg.sender, "Cannot transfer to yourself");

balances[msg.sender] -= amount;

balances[to] += amount;

emit Transfer(msg.sender, to, amount);

}

function getBalance(address _addr) public view returns (uint) {

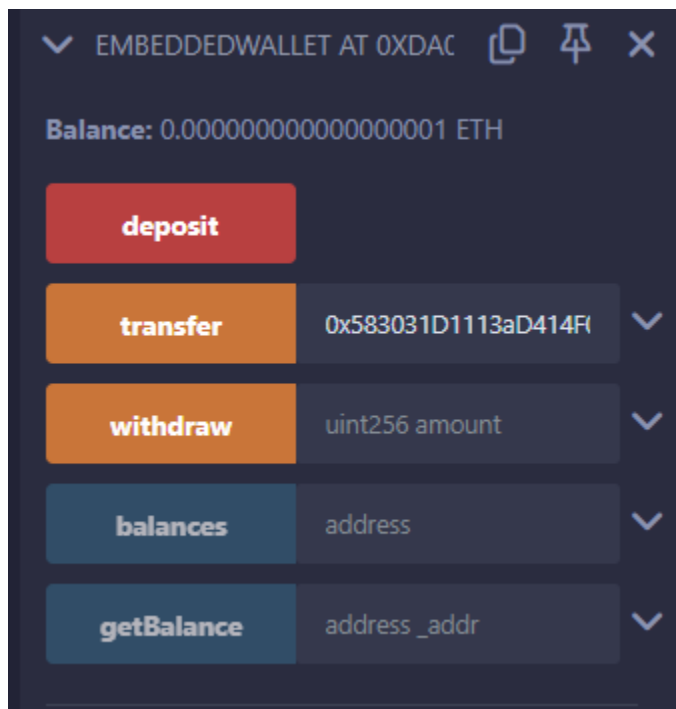
return balances[_addr];

}

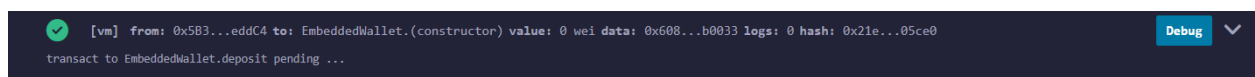
}

```

### Output:



### Deposit function -



## Transfer function -

```
✓ [vm] from: 0x583...eddC4 to: EmbeddedWallet.deposit() 0xDA0...42853 value: 1 wei data: 0xd0e...30db0 logs: 1 hash: 0xb38...601ee Debug ✓
transact to EmbeddedWallet.transfer errored: Error encoding arguments: Error: invalid BigNumber string (argument="value", value="", code=INVALID_ARGUMENT, version=bignumber/5.7.0)
transact to EmbeddedWallet.transfer errored: Error encoding arguments: Error: invalid BigNumber string (argument="value", value="", code=INVALID_ARGUMENT, version=bignumber/5.7.0)
transact to EmbeddedWallet.transfer pending ...
```