```solidity
@@ -0,0 +1,29 @@

1 +pragma solidity ^0.8.20;

2 +

3 +contract EtherSplitter {

4 +    event EtherDistributed (address indexed sender, uint totalAmount, uint recipients);

5 +    event TransferSuccess (address recipient, uint amount);

6 +    event TransferFailed (address recipient, uint amount);

7 +

8 +    // Payable function to split Ether among recipients

9 +    function splitEther (address[] calldata recipients) external payable {

10 +        uint totalRecipients =

recipients.length;

11 +        require(totalRecipients > 0, No recipients");

12 +        require(msg.value > 0, No Ether sent");
```

```solidity
13 +

14 +

uint amount Per Recipient = msg.

value / totalRecipients;

15 +

require (amount Per Recipient > 0

"Amount too small");

16 +

17 +

for (uint i = 0; i <

totalRecipients; i++) {

18 +

// Using call t
```

o send Ether with success/

fail check

19 +

(bool sent,) =

payable (recipients[i]).call {value: amount Per Recipient}("");

20 +

if (sent) {

21

+

emit Transfer Success

(recipients[i], amountPerRecipient);

22 +

} else {

23 +

emit Transfer Failed

(recipients[i], amountPerRecipient);

24 +

}

25 +

}

26 +

emit EtherDistributed(msg.

27 +

sender, msg.value, totalRecipients);

28 + }

29 +}