

# REMIX IDE:

The screenshot displays the Remix IDE interface with a Solidity contract named `FoodTracking.sol` being deployed. The contract code is as follows:

```
1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.0;
3 contract FoodTracking {
4     address public owner;
5     struct FoodItem {
6         uint itemId;
7         string itemName;
8         address producer;
9         uint productionDate;
10        uint expirationDate;
11        bool isConsumed;
12    }
13
14    mapping(uint => FoodItem) public foodItems;
15    uint public itemCount;
16
17    event FoodItemAdded(uint itemId, string itemName,
18    event FoodItemConsumed(uint itemId);
19
20    constructor() {
21        owner = msg.sender;
22    }
23
24    modifier onlyOwner() {
```

The left sidebar shows the **DEPLOY & RUN TRANSACTIONS** panel. The **ENVIRONMENT** is set to **Injected Provider - MetaMask**. The **ACCOUNT** is `0x95d...44826 (0 ether)`. The **GAS LIMIT** is `3000000`. The **VALUE** is `0 Wei`. The **CONTRACT** is `FoodTracking - FoodTracking.sol`. The **evm version** is `paris`. The **Deploy** button is highlighted.

The right sidebar shows the **CONTRACT DEPLOYMENT** panel. It displays the **DETAILS** tab with the following information:

- Gas (estimated):** \$29.62 **0.01847916 ETH**
- Very likely in < 15 seconds**
- Max fee:** 0.01847916 ETH
- Total:** \$29.62 **0.01847916 ETH**
- Amount + gas fee:** Max amount: 0.01847916 ETH

A warning message is displayed: "You do not have enough ETH in your account to pay for transaction fees on Ethereum Mainnet network. Buy ETH or deposit from another account."

The bottom status bar shows the system time as 14:04 on 20-10-2023, with a temperature of 33°C and a sunny weather forecast.

FOOD-TRACKING-SYSTEM/IMP: x Remix - Ethereum IDE x +

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96js

Sabarimala Online (Gmail YouTube Maps)

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

Injected Provider - MetaMask

Main (1) network

ACCOUNT

0x95d...44826 (0 ether)

GAS LIMIT

3000000

VALUE

0 Wei

CONTRACT

FoodTracking - Food Tracking.sol

evm version: paris

Deploy

Publish to IPFS

At Address Load contract from Address

Transactions recorded 1 1

Deployed Contracts

```
24 modifier onlyOwner() {
25     require(msg.sender == owner, "Only the owner can perform this action");
26 }
27
28
29 function addFoodItem(string memory _itemName, uint _expirationDate) public { @infinite gas
30     itemCount++;
31     foodItems[itemCount] = FoodItem({
32         itemId: itemCount,
33         itemName: _itemName,
34         producer: msg.sender,
35         productionDate: block.timestamp,
36         expirationDate: _expirationDate,
37         isConsumed: false
38     });
39     emit FoodItemAdded(itemCount, _itemName, msg.sender);
40 }
41
42 function consumeFoodItem(uint _itemId) public { @infinite gas
43     require(foodItems[_itemId].producer == msg.sender, "Only the producer can consume the item");
44     require(!foodItems[_itemId].isConsumed, "Item has already been consumed");
45     require(block.timestamp < foodItems[_itemId].expirationDate, "Item has expired");
46
47     foodItems[_itemId].isConsumed = true;
```

0 Listen on all transactions Search with transaction hash or address

creation of FoodTracking pending...

Type here to search

Games ENG 14:04 20-10-2023

FOOD-TRACKING-SYSTEM/IMP: x Remix - Ethereum IDE x +

remix.ethereum.org/#lang=en&optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.18+commit.87f61d96js

Sabarimala Online (Gmail YouTube Maps)

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT

Injected Provider - MetaMask

Main (1) network

ACCOUNT

0x95d...44826 (0 ether)

GAS LIMIT

3000000

VALUE

0 Wei

CONTRACT

FoodTracking - Food Tracking.sol

evm version: paris

Deploy

Publish to IPFS

At Address Load contract from Address

Transactions recorded 1 1

Deployed Contracts

```
34     producer: msg.sender,
35     productionDate: block.timestamp,
36     expirationDate: _expirationDate,
37     isConsumed: false
38 });
39 emit FoodItemAdded(itemCount, _itemName, msg.sender);
40 }
41
42 function consumeFoodItem(uint _itemId) public { @infinite gas
43     require(foodItems[_itemId].producer == msg.sender, "Only the producer can consume the item");
44     require(!foodItems[_itemId].isConsumed, "Item has already been consumed");
45     require(block.timestamp < foodItems[_itemId].expirationDate, "Item has expired");
46
47     foodItems[_itemId].isConsumed = true;
48     emit FoodItemConsumed(_itemId);
49 }
50 }
```

0 Listen on all transactions Search with transaction hash or address

creation of FoodTracking pending...

Type here to search

Games ENG 14:05 20-10-2023