FORESIGHT X

<OpenBuild Hack@HK/>

<Solidity Improvement Workshop />

Frank

Security Tech Lead @ Beosin

Foresight X
Gateway to
Web3 in Hongkong

OpenBuild
HACK.
SERIES

BLOCKCHAIN ACADEMY × BEOSIN Blockchain Security

代币和流动性挖矿

"SECURITY IN WEB3" WORKSHOP

Contents


BLOCKCHAIN ACADEMY
BEOSIN
Blockchain Security
BEOSIN
Blockchain Security
代币发行

代币发行

ICO：Initial Coin Offering，首次代币发行

IEO：即Initial Exchange Offering，也称为LaunchPad，首次交易所发行

IEO的 E (Exchange)是指中心化的交易所如：币安、FTX、Coinbase

IDO：即 Initial Dex Offering，首次去中心化交易所发行

IDO的 D (DEX)是指去中心化的交易所如：Uniswap、PancakeSwap

BNB ICO

Allocation

%	Amount (BNB)	Participant
50%	100,000,000	ICO
40%	80,000,000	Founding Team
10%	20,000,000	Angel investors

ICO Schedule

All times below are China Standard Time (CST), UTC+8 hours.

Date	Task
2017/06/14	Confirmed start of the Binance project
2017/06/16	Initial draft white paper completed, circulated to potential angel investors
2017/06/22	Announce Binance ICO plan, and release whitepaper to general public
2017/07/01	ICO starts (platforms will be announced soon)
2017/07/15	Binance.com release v0.1 go live, active trading begins
2017/07/21	ICO finishes, or whenever the coins are sold out

ICO will start from 3PM July 1st, investors can purchase BNB tokens in 3 phases on a first-come, first-served basis until 100,000,000 tokens are sold. As each new phase starts, the price will increase.

Investors will receive BNB tokens within 5 working days after the ICO finishes.

ICO 和 NFT

1. NFT的投资是对已实现的、目前存在的资产进行投资

2. NFT是社区的名片，并且不会夸大其功能实用性

3. NFT更多是针对实物资产的延伸

BEOSIN
Blockchain Security

代币标准

代币标准

ERC-20: 同质化代币标准，如：虚拟货币、投票代币、质押代币等

ERC-721: 非同质化代币标准，如：数字艺术、游戏道具、虚拟房地产等场景

ERC-1155: 允许一种代币合约同时管理多种同质化和非同质化代币，使得其适用于游戏中的道具和物品

ERC-20（EIP）

```
function name() public view returns (string)

function symbol() public view returns (string)

function decimals() public view returns (uint8)

function totalSupply() public view returns (uint256)

function balanceOf(address _owner) public view returns (uint256 balance)

function transfer(address _to, uint256 _value) public returns (bool success)

function transferFrom(address _from, address _to, uint256 _value) public returns (bool success)

function approve(address _spender, uint256 _value) public returns (bool success)

function allowance(address _owner, address _spender) public view returns (uint256 remaining)

event Transfer(address indexed _from, address indexed _to, uint256 _value)

event Approval(address indexed _owner, address indexed _spender, uint256 _value)
```

ERC-20

```
 * - `to` cannot be the zero address.
 * - the caller must have a balance of at least `amount`.
 */
function transfer(address to, uint256 amount) public virtual override returns (bool) {
    address owner = _msgSender();
    _transfer(owner, to, amount);
    return true;
}
```

```
 * - `from` cannot be the zero address.
 * - `to` cannot be the zero address.
 * - `from` must have a balance of at least `amount`.
 */
function _transfer(address from, address to, uint256 amount) internal virtual {
    require(from != address(0), "ERC20: transfer from the zero address");
    require(to != address(0), "ERC20: transfer to the zero address");

    _beforeTokenTransfer(from, to, amount);

    uint256 fromBalance = _balances[from];
    require(fromBalance >= amount, "ERC20: transfer amount exceeds balance");
    unchecked {
        _balances[from] = fromBalance - amount;
        // Overflow not possible: the sum of all balances is capped by totalSupply, and the sum is preserved by
        // decrementing then incrementing.
        _balances[to] += amount;
    }

    emit Transfer(from, to, amount);

    _afterTokenTransfer(from, to, amount);
}
```

ERC-20

```
 *
 * - `from` and `to` cannot be the zero address.
 * - `from` must have a balance of at least `amount`.
 * - the caller must have allowance for ``from``'s tokens of at least
 * `amount`.
 */
function transferFrom(address from, address to, uint256 amount) public virtual override returns (bool) {
    address spender = _msgSender();
    _spendAllowance(from, spender, amount);
    _transfer(from, to, amount);
    return true;
}
```

```
/**
 * @dev Updates `owner` s allowance for `spender` based on spent `amount`.
 *
 * Does not update the allowance amount in case of infinite allowance.
 * Revert if not enough allowance is available.
 *
 * Might emit an {Approval} event.
 */
function _spendAllowance(address owner, address spender, uint256 amount) internal virtual {
    uint256 currentAllowance = allowance(owner, spender);
    if (currentAllowance != type(uint256).max) {
        require(currentAllowance >= amount, "ERC20: insufficient allowance");
        unchecked {
            _approve(owner, spender, currentAllowance - amount);
        }
    }
}
```

ERC-20

```
 * Requirements:
 *
 * - `spender` cannot be the zero address.
 */
function approve(address spender, uint256 amount) public virtual override returns (bool) {
    address owner = _msgSender();
    _approve(owner, spender, amount);
    return true;
}
```

```
 * Requirements:
 *
 * - `owner` cannot be the zero address.
 * - `spender` cannot be the zero address.
 */
function _approve(address owner, address spender, uint256 amount) internal virtual {
    require(owner != address(0), "ERC20: approve from the zero address");
    require(spender != address(0), "ERC20: approve to the zero address");

    _allowances[owner][spender] = amount;
    emit Approval(owner, spender, amount);
}
```

ERC-721（eip）


```
function balanceOf(address _owner) external view returns (uint256);

function ownerOf(uint256 _tokenId) external view returns (address);

function safeTransferFrom(address _from, address _to, uint256 _tokenId, bytes data) external payable;

function safeTransferFrom(address _from, address _to, uint256 _tokenId) external payable;

function transferFrom(address _from, address _to, uint256 _tokenId) external payable;

function approve(address _approved, uint256 _tokenId) external payable;

function setApprovalForAll(address _operator, bool _approved) external;

function getApproved(uint256 _tokenId) external view returns (address);

function isApprovedForAll(address _owner, address _operator) external view returns (bool);
```

```
event Transfer(address indexed _from, address indexed _to, uint256 indexed _tokenId);

event Approval(address indexed _owner, address indexed _approved, uint256 indexed _tokenId);

event ApprovalForAll(address indexed _owner, address indexed _operator, bool _approved);
```

ERC-721

```
function safeTransferFrom(address from, address to, uint256 tokenId) public virtual override {
    safeTransferFrom(from, to, tokenId, "");
}

/**
 * @dev See {IERC721-safeTransferFrom}.
 */
function safeTransferFrom(address from, address to, uint256 tokenId, bytes memory data) public virtual override {
    require(_isApprovedOrOwner(_msgSender(), tokenId), "ERC721: caller is not token owner or approved");
    _safeTransfer(from, to, tokenId, data);
}
```

```
function _safeTransfer(address from, address to, uint256 tokenId, bytes memory data) internal virtual {
    _transfer(from, to, tokenId);
    require(_checkOnERC721Received(from, to, tokenId, data), "ERC721: transfer to non ERC721Receiver implementer");
}
```

```
function _transfer(address from, address to, uint256 tokenId) internal virtual {
    require(ERC721.ownerOf(tokenId) == from, "ERC721: transfer from incorrect owner");
    require(to != address(0), "ERC721: transfer to the zero address");

    _beforeTokenTransfer(from, to, tokenId, 1);

    // Check that tokenId was not transferred by `_beforeTokenTransfer` hook
    require(ERC721.ownerOf(tokenId) == from, "ERC721: transfer from incorrect owner");

    // Clear approvals from the previous owner
    delete _tokenApprovals[tokenId];

    unchecked {
        // `_balances[from]` cannot overflow for the same reason as described in `_burn`:
        // `from`'s balance is the number of token held, which is at least one before the current
        // transfer.
        // `_balances[to]` could overflow in the conditions described in `_mint`. That would require
        // all 2**256 token ids to be minted, which in practice is impossible.
        _balances[from] -= 1;
        _balances[to] += 1;
    }
    _owners[tokenId] = to;

    emit Transfer(from, to, tokenId);

    _afterTokenTransfer(from, to, tokenId, 1);
}
```

ERC-721


```
function _checkOnERC721Received(
    address from,
    address to,
    uint256 tokenId,
    bytes memory data
) private returns (bool) {
    if (to.isContract()) {
        try IERC721Receiver(to).onERC721Received(_msgSender(), from, tokenId, data) returns (bytes4 retval) {
            return retval == IERC721Receiver.onERC721Received.selector;
        } catch (bytes memory reason) {
            if (reason.length == 0) {
                revert("ERC721: transfer to non ERC721Receiver implementer");
            } else {
                /// @solidity memory-safe-assembly
                assembly {
                    revert(add(32, reason), mload(reason))
                }
            }
        }
    } else {
        return true;
    }
}
```

ERC-721

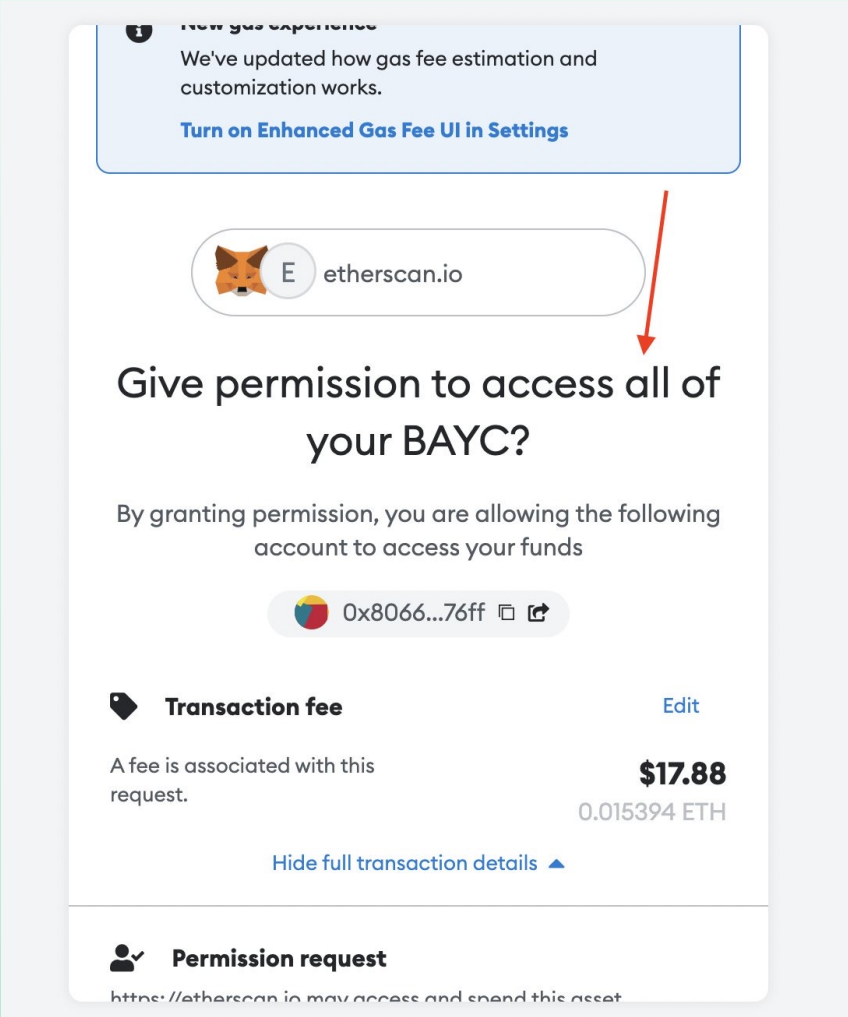
```
function transferFrom(address from, address to, uint256 tokenId) public virtual override {
    //solhint-disable-next-line max-line-length
    require(_isApprovedOrOwner(_msgSender(), tokenId), "ERC721: caller is not token owner or approved");

    _transfer(from, to, tokenId);
}
```

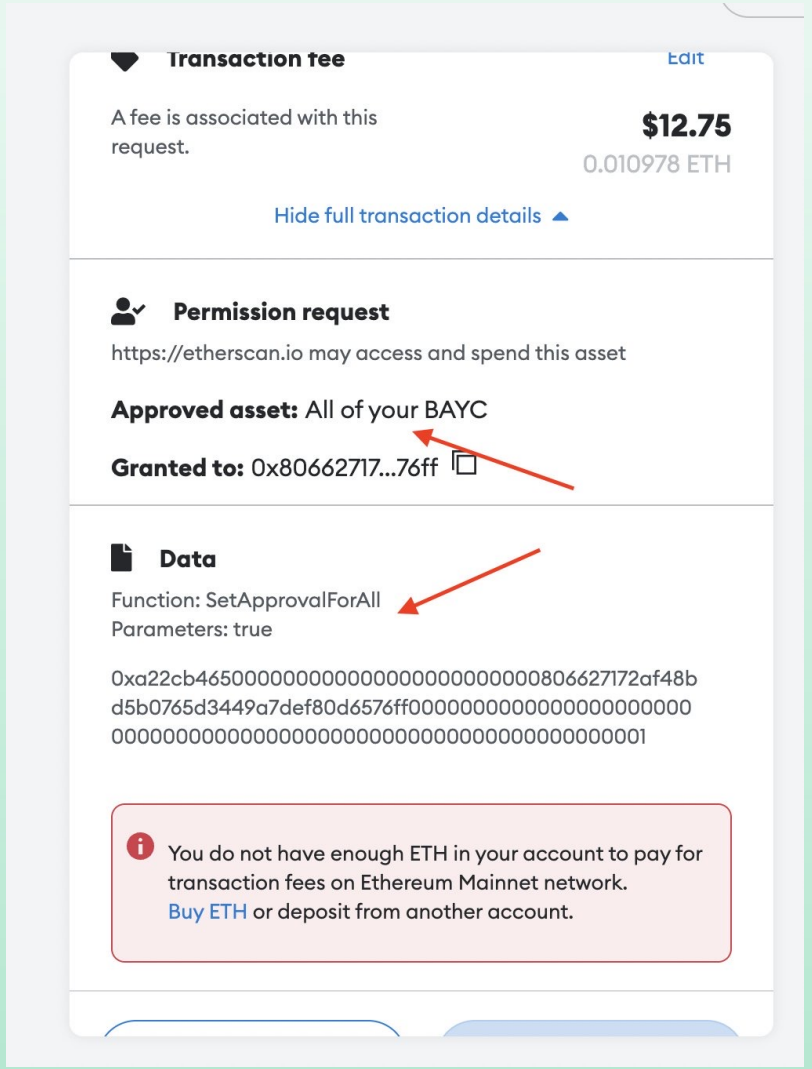
```
function _isApprovedOrOwner(address spender, uint256 tokenId) internal view virtual returns (bool) {
    address owner = ERC721.ownerOf(tokenId);
    return (spender == owner || isApprovedForAll(owner, spender) || getApproved(tokenId) == spender);
}
```

```
function isApprovedForAll(address owner, address operator) public view virtual override returns (bool) {
    return _operatorApprovals[owner][operator];
}
```

```
function getApproved(uint256 tokenId) public view virtual override returns (address) {
    _requireMinted(tokenId);

    return _tokenApprovals[tokenId];
}
```

ERC-721


We've updated how gas fee estimation and customization works.
Turn on Enhanced Gas Fee UI in Settings
etherscan.io
Give permission to access all of your BAYC?
By granting permission, you are allowing the following account to access your funds
0x8066...76ff
Transaction fee
Edit
A fee is associated with this request.
$17.88
0.015394 ETH
Hide full transaction details
Permission request


Transaction fee
Edit
A fee is associated with this request.
$12.75
0.010978 ETH
Hide full transaction details
Permission request
https://etherscan.io may access and spend this asset
Approved asset: All of your BAYC
Granted to: 0x80662717...76ff
Data
Function: SetApprovalForAll
Parameters: true
0xa22cb465000000000000000000000000806627172af48b
d5b0765d3449a7def80d6576ff00000000000000000000000
0000000000000000000000000000000000000000000001
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.

ERC-1155

- 批量传输：通过一次合约调用转移多种类型的资产
- 批量余额查询：在一次调用中获取多个资产的余额
- 批量approve：将指定地址的所有代币授权到一个地址
- Hook：接收代币的Hook函数
- 支持NFT：如果供应量为1，就将其作为NFT处理

ERC-1155

```
// ERC-20
function transferFrom(address from, address to, uint256 value) external returns
(bool);

// ERC-1155
function safeBatchTransferFrom(
    address _from,
    address _to,
    uint256[] calldata _ids,
    uint256[] calldata _values,
    bytes calldata _data
) external;
```

```
// ERC-20
function balanceOf(address owner) external view returns (uint256);

// ERC-1155
function balanceOfBatch(
    address[] calldata _owners,
    uint256[] calldata _ids
) external view returns (uint256[] memory);
```

ERC-1155

```
// ERC-1155
function setApprovalForAll(
    address _operator,
    bool _approved
) external;

function isApprovedForAll(
    address _owner,
    address _operator
) external view returns (bool);
```

```
function onERC1155BatchReceived(
    address _operator,
    address _from,
    uint256[] calldata _ids,
    uint256[] calldata _values,
    bytes calldata _data
) external returns(bytes4);
```

BLOCKCHAIN ACADEMY × BEOSIN Blockchain Security

BEOSIN
Blockchain Security

Remix发行ERC-20代币

ERC-20（实现）

```
contract ERC20 is Context, IERC20, IERC20Metadata {
    mapping(address => uint256) private _balances;

    mapping(address => mapping(address => uint256)) private _allowances;

    uint256 private _totalSupply;

    string private _name;
    string private _symbol;
```

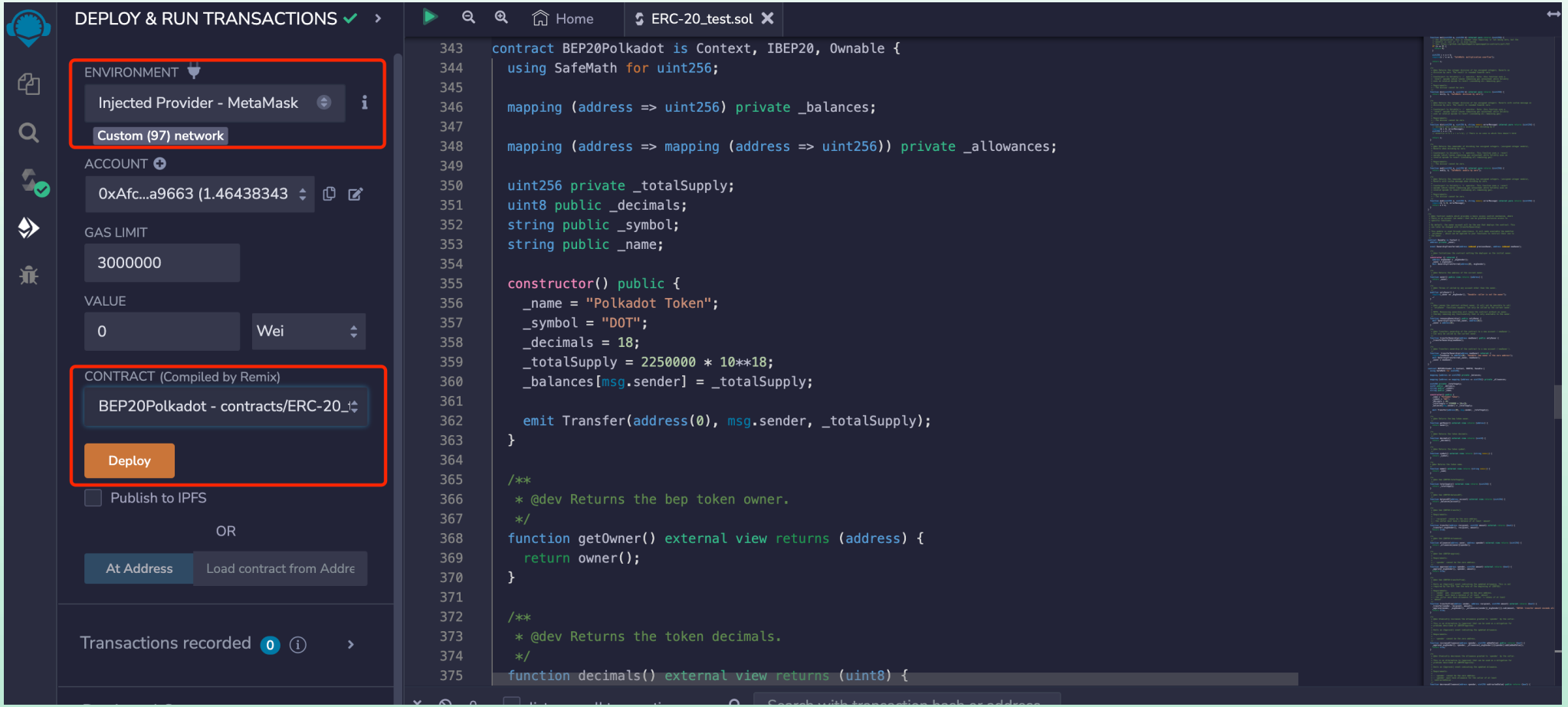
```
constructor(string memory name_, string memory symbol_) {
    _name = name_;
    _symbol = symbol_;
}
```

ERC-20

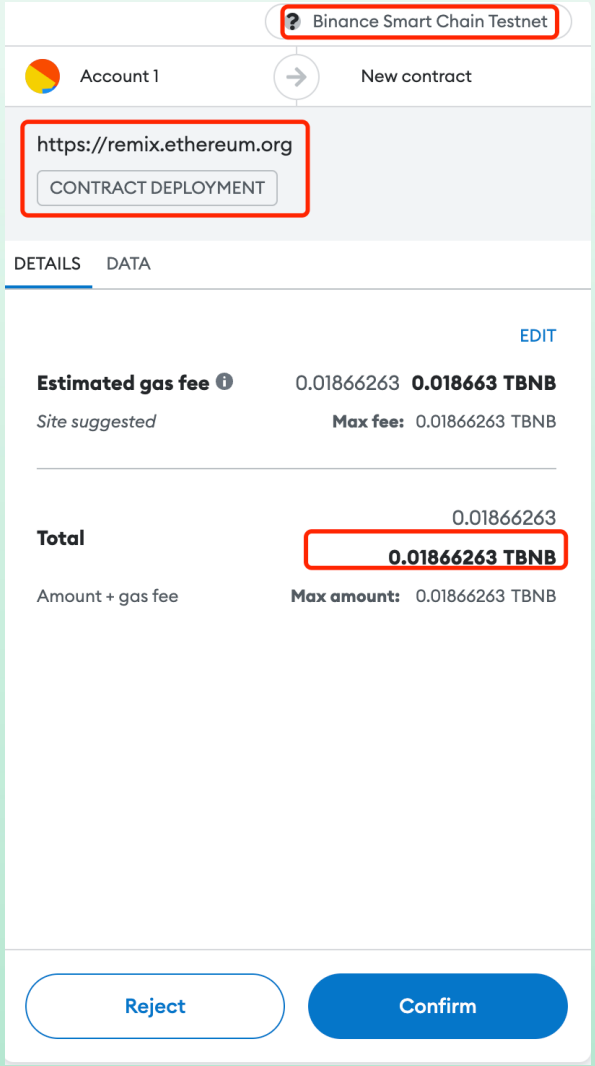
```
 * - `spender` cannot be the zero address.
 */
function increaseAllowance(address spender, uint256 addedValue) public virtual returns (bool) {
    address owner = _msgSender();
    _approve(owner, spender, allowance(owner, spender) + addedValue);
    return true;
}
```

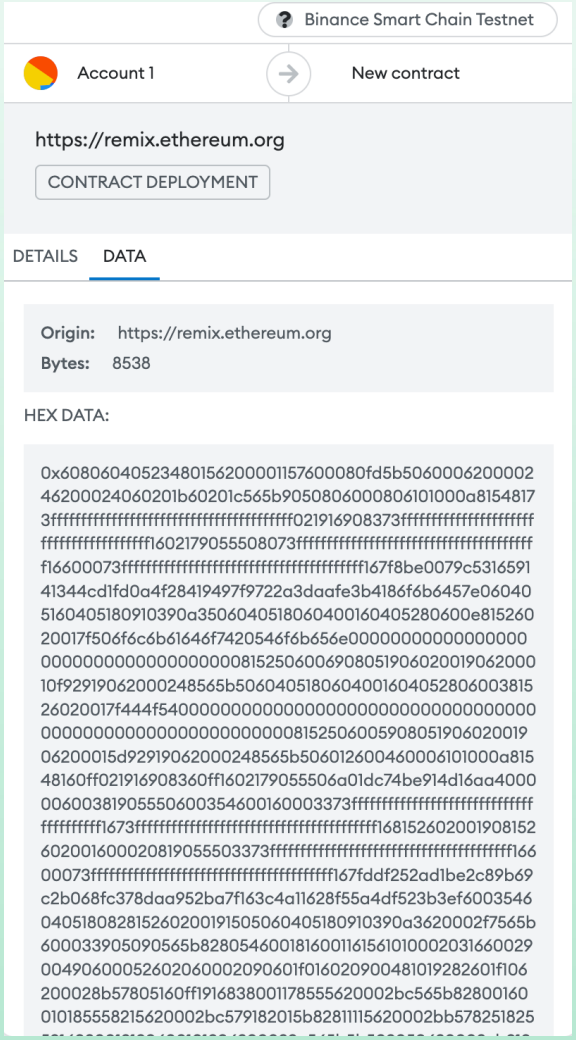
```
 * - `spender` cannot be the zero address.
 * - `spender` must have allowance for the caller of at least
 * `subtractedValue`.
 */
function decreaseAllowance(address spender, uint256 subtractedValue) public virtual returns (bool) {
    address owner = _msgSender();
    uint256 currentAllowance = allowance(owner, spender);
    require(currentAllowance >= subtractedValue, "ERC20: decreased allowance below zero");
    unchecked {
        _approve(owner, spender, currentAllowance - subtractedValue);
    }

    return true;
}
```

Token发布


DEPLOY & RUN TRANSACTIONS
ENVIRONMENT
Injected Provider - MetaMask
Custom (97) network
ACCOUNT
0xAfc...a9663 (1.46438343
GAS LIMIT
3000000
VALUE
0
Wei
CONTRACT (Compiled by Remix)
BEP20Polkadot - contracts/ERC-20_1
Deploy
Publish to IPFS
OR
At Address
Load contract from Addre
Transactions recorded 0
Home
ERC-20_test.sol
contract BEP20Polkadot is Context, IBEP20, Ownable {
using SafeMath for uint256;
mapping (address => uint256) private _balances;
mapping (address => mapping (address => uint256)) private _allowances;
uint256 private _totalSupply;
uint8 public _decimals;
string public _symbol;
string public _name;
constructor() public {
_name = "Polkadot Token";
_symbol = "DOT";
_decimals = 18;
_totalSupply = 2250000 * 10**18;
_balances[msg.sender] = _totalSupply;
emit Transfer(address(0), msg.sender, _totalSupply);
}
/**
* @dev Returns the bep token owner.
*/
function getOwner() external view returns (address) {
return owner();
}
/**
* @dev Returns the token decimals.
*/
function decimals() external view returns (uint8) {

Token发布


Binance Smart Chain Testnet
Account 1
New contract
https://remix.ethereum.org
CONTRACT DEPLOYMENT
DETAILS
DATA
EDIT
Estimated gas fee
0.01866263 0.018663 TBNB
Site suggested
Max fee: 0.01866263 TBNB
Total
0.01866263
0.01866263 TBNB
Amount + gas fee
Max amount: 0.01866263 TBNB
Reject
Confirm


Binance Smart Chain Testnet
Account 1
New contract
https://remix.ethereum.org
CONTRACT DEPLOYMENT
DETAILS
DATA
Origin: https://remix.ethereum.org
Bytes: 8538
HEX DATA:
0x6080604052348015620000115760008 0fd5b5060006200002
46200024060201b60201c565b905080600080610100 0a8154817
3ffffffffffffffffffffffffffffffffffffffff021916908373ffffffffffffffffffffffff
ffffffffffffffff1602179055508073ffffffffffffffffffffffffffffffffffffffff
f16600073ffffffffffffffffffffffffffffffffffffffff167f8be0079c5316591
41344cd1fd0a4f28419497f9722a3daafe3b4186f6b6457e06040
5160405180910390a3506040518060400160405280600e815260
20017f506f6c6b61646f7420546f6b656e00000000000000000
00000000000000000008152506006908051906020019062000
10f92919062000248565b5060405180604001604052806003815
26020017f444f5400000000000000000000000000000000000000
00000000000000000000000081525060059080519060200019
06200015d92919062000248565b506012600460006101000a815
48160ff021916908360ff1602179055506a01dc74be914d16aa4000
0060038190555060035460016000337 3ffffffffffffffffffffffffffffffff
ffffffff1673ffffffffffffffffffffffffffffffffffffffff16815260200160
602001600020819055503373ffffffffffffffffffffffffffffffffffffffff166
00073ffffffffffffffffffffffffffffffffffffffff167fddf252ad1be2c89b69
c2b068fc378daa952ba7f163c4a11628f55a4df523b3ef6003546
040518082815260200191505060405180910390a3620002f7565b
600033905090565b828054600181600116156101000203166002900
4906000526020600020906 01f016020900481019282601f106
200028b57805160ff1916838001178555620002bc565b82800160
010185558215620002bc579182015b828111156200002bb578251825

Token发布

[block:27573585 txIndex:4] from: 0xAfc...a9663 to: BEP20Polkadot.(constructor) value: 0 wei data: 0x608...00032 logs: 2 hash: 0x02b...c7ddb

Debug

status	true Transaction mined and execution succeed
transaction hash	0x679cc46e508d641ebb01207d132bbb0cf194c6eb20e9ade5ed0f1635802a0917
from	0xAfc[redacted]
to	BEP20Polkadot.(constructor)

Transaction Details

Overview | Logs (2)

[This is a Bsc **Testnet** transaction only]

Transaction Hash:	0x679cc46e508d641ebb01207d132bbb0cf194c6eb20e9ade5ed0f1635802a0917
Status:	Success
Block:	27573585 69 Block Confirmations
Timestamp:	3 mins ago (Feb-26-2023 10:01:25 AM +UTC)
From:	0xa[redacted]63
Interacted With (To):	[Contract 0xedebc6735b52b02b17e427735f8ca7cdcf2c9c2b Created]
Tokens Transferred:	From 0x0000000000000… To 0xafcd13ef950e8e… For 2,250,000 Polkadot Tok... (DOT)
Value:	0 BNB ($0.00)
Transaction Fee:	0.01866263 BNB ($5.66)

Token发布

Token Polkadot Token

Overview BEP-20

Total Supply:	2,250,000 **DOT**
Holders:	1 addresses
Transfers:	1

Profile Summary

Contract:	0xedebc6735b52b02b17e427735f8ca7cdcf2c9c2b
Decimals:	18

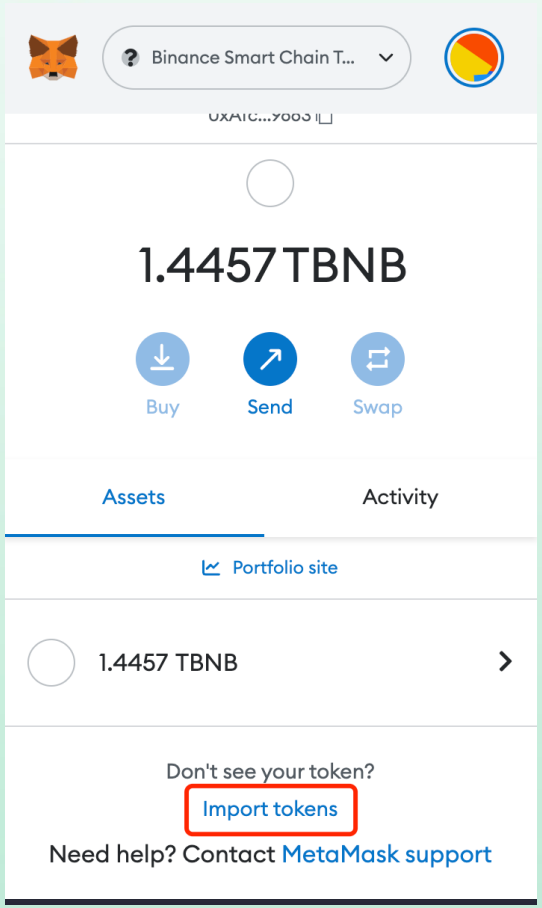
Transfers | Holders | Contract

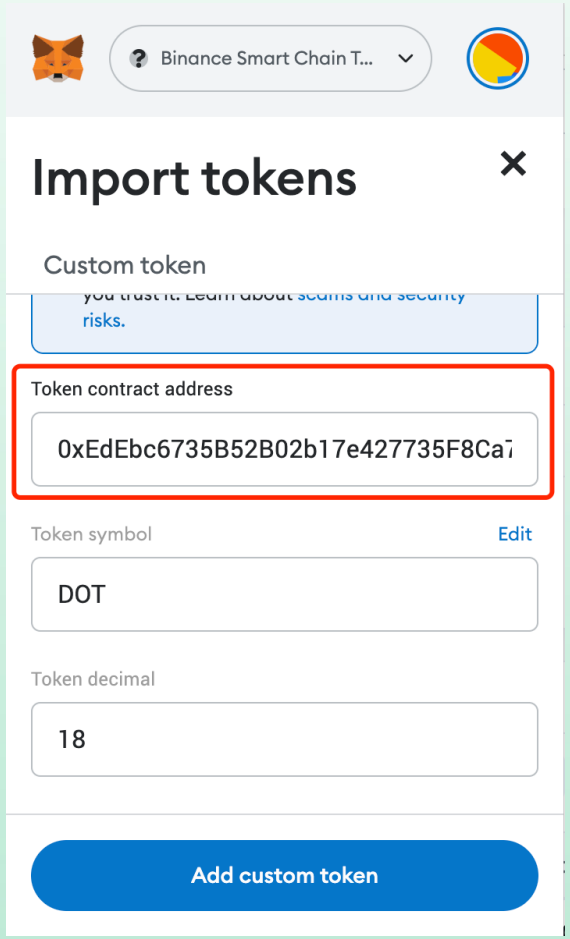
A total of 1 transaction found

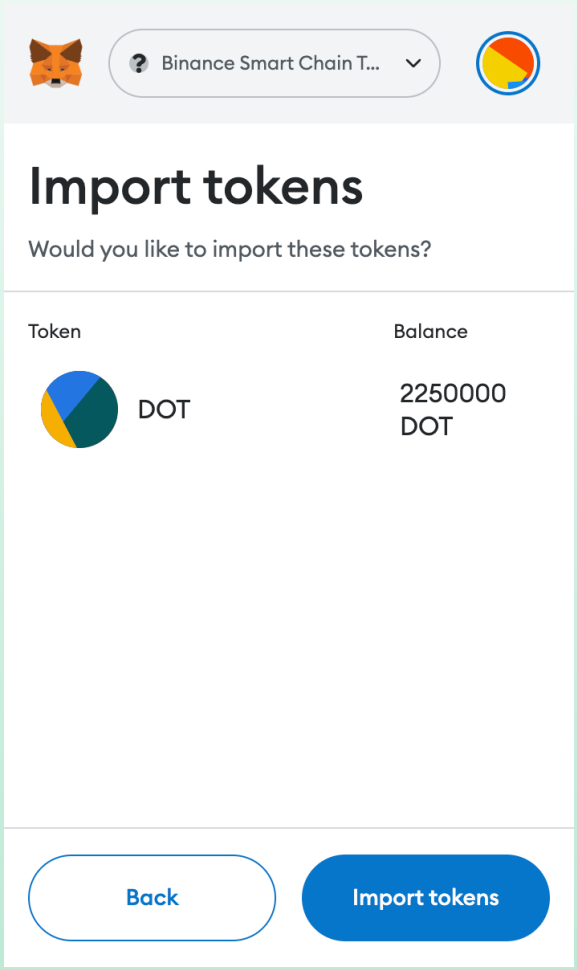
First | Page 1 of 1 | Last

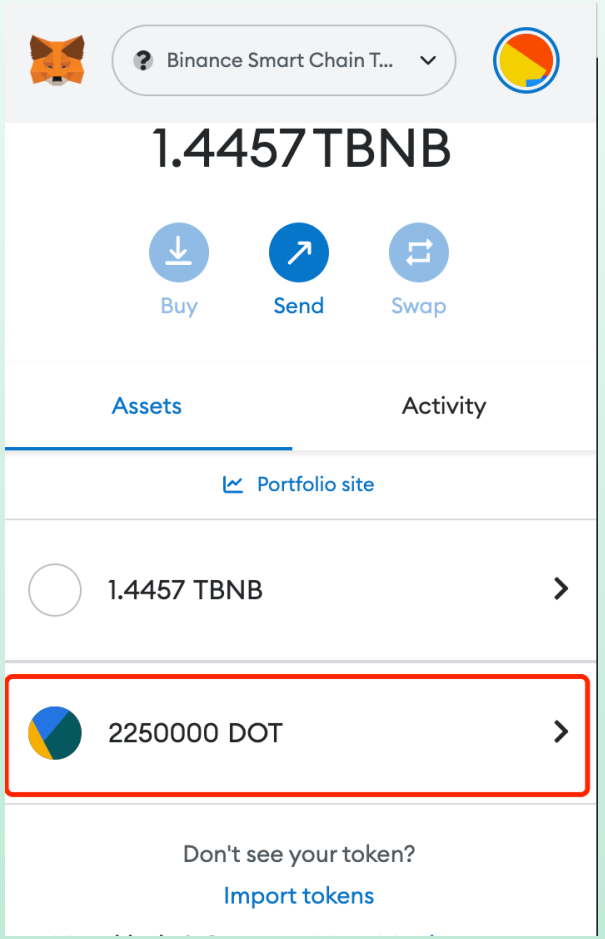
Txn Hash	Method	Age	From	To	Quantity
0x679cc46e508d641ebb…	0x60806040	6 mins ago	0x0000000000000000000…	0xafcd13ef950e8edd869…	2,250,000

Token导入


Binance Smart Chain T...
1.4457 TBNB
Buy
Send
Swap
Assets
Activity
Portfolio site
1.4457 TBNB
Don't see your token?
Import tokens
Need help? Contact MetaMask support

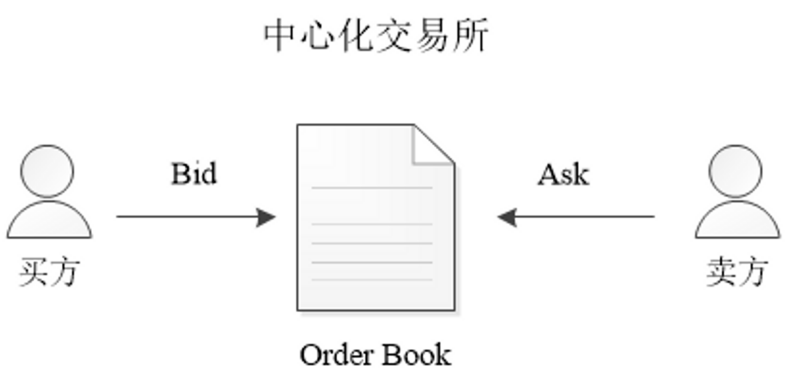

Binance Smart Chain T...
Import tokens
Custom token
risks.
Token contract address
0xEdEbc6735B52B02b17e427735F8Ca7
Token symbol
Edit
DOT
Token decimal
18
Add custom token

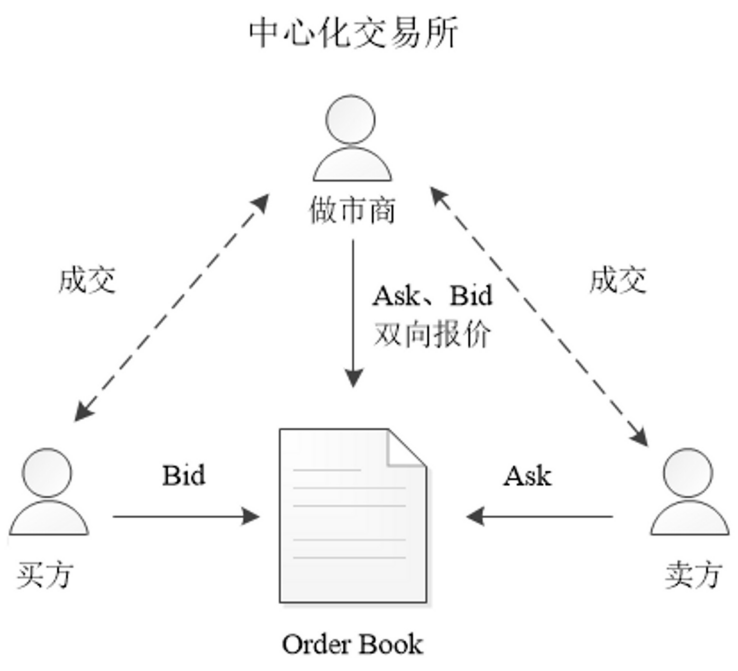

Binance Smart Chain T...
Import tokens
Would you like to import these tokens?
Token
Balance
DOT
2250000 DOT
Back
Import tokens

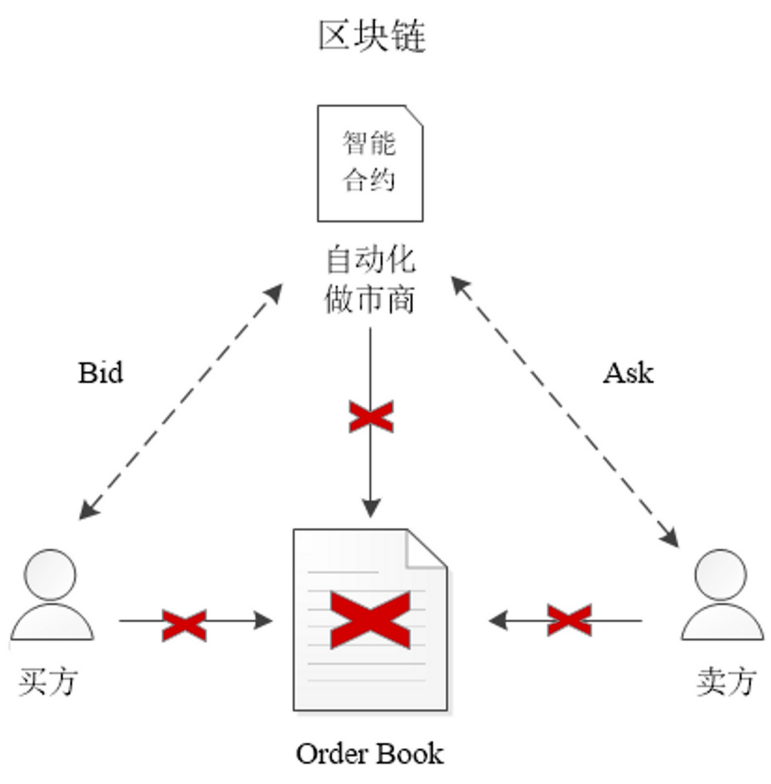

Binance Smart Chain T...
1.4457 TBNB
Buy
Send
Swap
Assets
Activity
Portfolio site
1.4457 TBNB
2250000 DOT
Don't see your token?
Import tokens


BLOCKCHAIN ACADEMY
BEOSIN
Blockchain Security
BEOSIN
Blockchain Security
Remix部署流动性
挖矿合约

交易所发展历程


中心化交易所
Bid
Ask
买方
卖方
Order Book


中心化交易所
做市商
成交
Ask、Bid
双向报价
成交
Bid
Ask
买方
卖方
Order Book


区块链
智能
合约
自动化
做市商
Bid
Ask
买方
卖方
Order Book

AMM基本概念

流动性池：两种不同的ERC-20代币组成一个流动性池，二者互相之间可以进行兑换

流动性：常见AMM中资产的兑换是两两进行的，其中可用于兑换的资产就是流动性

AMM中的三种用户角色：

交易者：希望在AMM中将某类资产交换为另一种资产

流动性提供者：提供资产来满足交易需求，赚取交易费，添加流动性后会给其发放流动性凭证lp token

套利者：通过套利行为将交易对中的资产维持在市场价格

流动性挖矿

流动性挖矿（liquidity mining）是指通过提供流动性来支持交易所的运行，获得一定数量的代币奖励。

它通常在去中心化交易所（DEX）中实现，例如 Uniswap、Balancer、SushiSwap 等。这个过程通常需要将代币锁定在流动性池中，并提供给交易对进行交易。这种方式可以激励用户为交易所提供更多的流动性，从而增加交易量和深度，进而吸引更多的用户和流动性。

MasterChef是一个去中心化金融平台（DeFi）的代币奖励机制，基于流动性挖矿。MasterChef为提供流动性的用户提供一种奖励代币，用户可以将他们的奖励代币放入农场中以获得更多的奖励，同时这些代币可以随时取出并交易。

MasterChef

```
struct UserInfo {
    uint256 amount; // How many LP tokens the user has provided.
    uint256 rewardDebt; // Reward debt. See explanation below.
}
```

```
struct PoolInfo {
    IERC20 lpToken; // Address of LP token contract.
    uint256 allocPoint; // How many allocation points assigned to this pool. SUSHIs to distribute per block.
    uint256 lastRewardBlock; // Last block number that SUSHIs distribution occurs.
    uint256 accSushiPerShare; // Accumulated SUSHIs per share, times 1e12. See below.
}
```

```
    // The SUSHI TOKEN!
    SushiToken public sushi;
    // Info of each pool.
    PoolInfo[] public poolInfo;
    // Info of each user that stakes LP tokens.
    mapping (uint256 => mapping (address => UserInfo)) public userInfo;
    // Total allocation poitns. Must be the sum of all allocation points in all pools.
    uint256 public totalAllocPoint = 0;
    // The block number when SUSHI mining starts.
    uint256 public startBlock = 0;
    event Deposit(address indexed user, uint256 indexed pid, uint256 amount);
    event Withdraw(address indexed user, uint256 indexed pid, uint256 amount);
```

```
constructor(
    SushiToken _sushi
) public {
    sushi = _sushi;
}
```

MasterChef

```
// 添加质押池
// 添加一个质押池，一个质押池100个分配点位
function add(IERC20 _lpToken) public {
    uint256 lastRewardBlock = block.number > startBlock ? block.number : startBlock;
    totalAllocPoint = totalAllocPoint.add(100);
    poolInfo.push(PoolInfo({
        lpToken: _lpToken,
        // 这个质押池的分配点位
        allocPoint: 100,
        // 上一个更新奖励的区块
        lastRewardBlock: lastRewardBlock,
        // 质押一个lp token的全局收益
        // 用户在质押LPToken的时候，会把当前accSushiPerShare记下来作为起始点位，当解除质押的时候，可以通过最新的accSushiPerShare减去起始点位，就可以得到用户实际的收益。
        accSushiPerShare: 0
    }));
}
```

```
// Deposit LP tokens to MasterChef for SUSHI allocation.
function deposit(uint256 _pid, uint256 _amount) public {
    PoolInfo storage pool = poolInfo[_pid];
    UserInfo storage user = userInfo[_pid][msg.sender];
    updatePool(_pid);
    // 追加质押，先结算之前的奖励
    if (user.amount > 0) {
        uint256 pending = user.amount.mul(pool.accSushiPerShare).div(1e12).sub(user.rewardDebt);
        sushi.transfer(msg.sender, pending);
    }
    // 把用户的lp token转移到 MasterChef 合约中
    pool.lpToken.safeTransferFrom(address(msg.sender), address(this), _amount);
    user.amount = user.amount.add(_amount);
    // 更新不可领取部分
    user.rewardDebt = user.amount.mul(pool.accSushiPerShare).div(1e12);
    emit Deposit(msg.sender, _pid, _amount);
}
```

MasterChef

```
// Withdraw LP tokens from MasterChef.
function withdraw(uint256 _pid, uint256 _amount) public {
    PoolInfo storage pool = poolInfo[_pid];
    UserInfo storage user = userInfo[_pid][msg.sender];
    require(user.amount >= _amount, "withdraw: not good");
    updatePool(_pid);
    uint256 pending = user.amount.mul(pool.accSushiPerShare).div(1e12).sub(user.rewardDebt);
    sushi.transfer(msg.sender,pending);
    user.amount = user.amount.sub(_amount);
    user.rewardDebt = user.amount.mul(pool.accSushiPerShare).div(1e12);
    pool.lpToken.safeTransfer(address(msg.sender), _amount);
    emit Withdraw(msg.sender, _pid, _amount);
}
```

MasterChef部署

Transaction Details

Overview | Logs (1)

[This is a Bsc **Testnet** transaction only]

Transaction Hash:	0x7459bea80[redacted]9a9743e3291dbd04ea
Status:	Success
Block:	27593992 56 Block Confirmations
Timestamp:	2 mins ago (Feb-27-2023 03:02:02 AM +UTC)
From:	0xafcd13e[redacted]82f0a9663
To:	[Contract 0x419fac9ac507e56409e12fb618d90925101ac3ab Created]
Value:	0 BNB ($0.00)
Transaction Fee:	0.03305037560273 BNB ($10.20)

Click to see More

MasterChef部署

1. 部署质押奖励的代币合约 SushiToken.sol

2. 部署流动性挖矿合约 MasterChef（构造函数中传入SushiToken 合约地址）

3. 调用SushiToken的transferOwnership()将owner权限设置为MasterChef合约地址

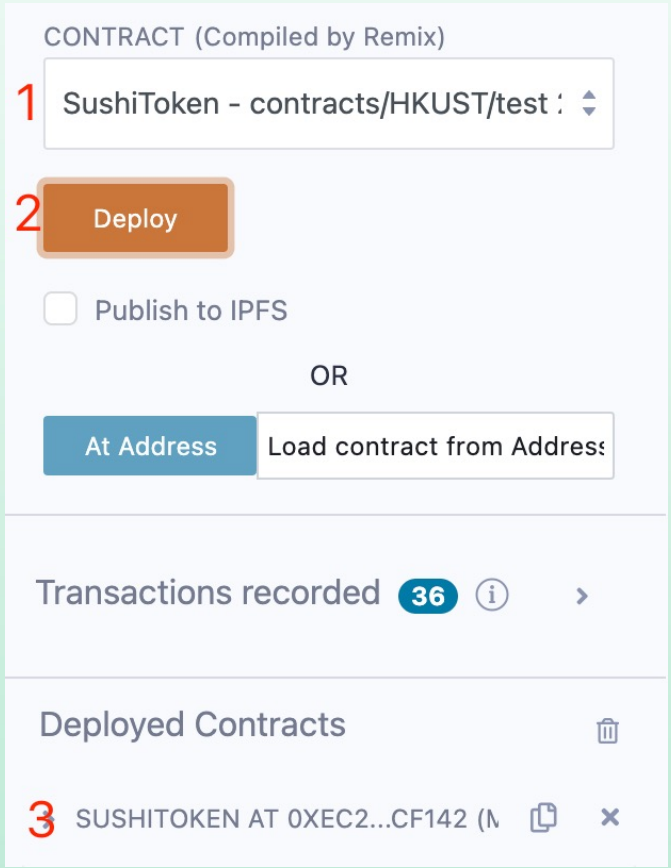
4. 部署lp token合约 BEP20.sol

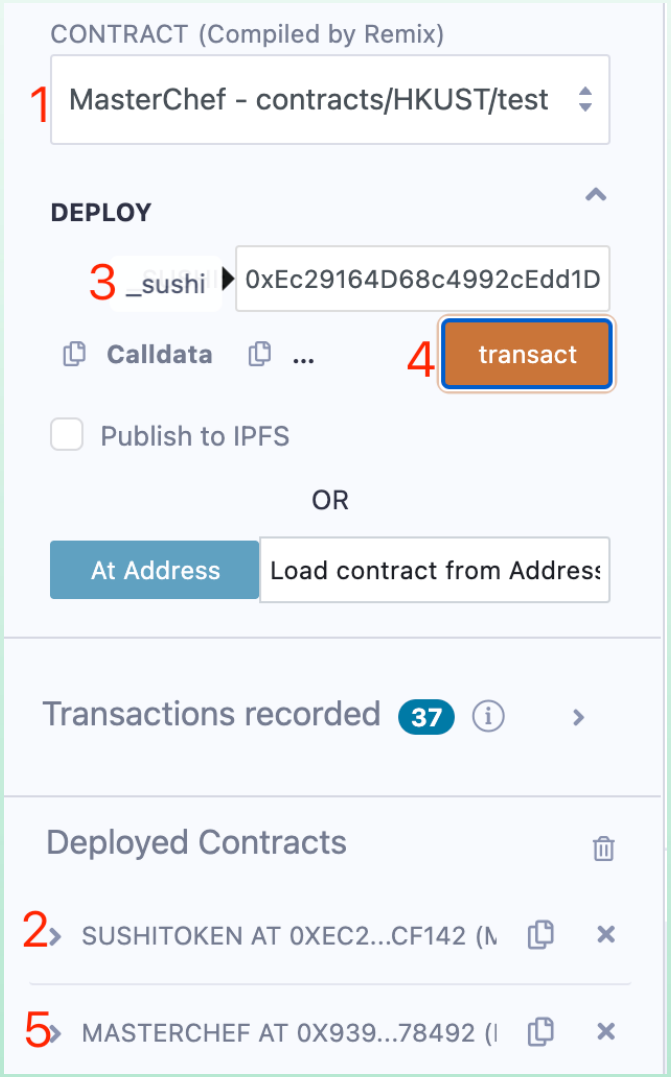
5. 调用BEP20的approve()授权MasterChef

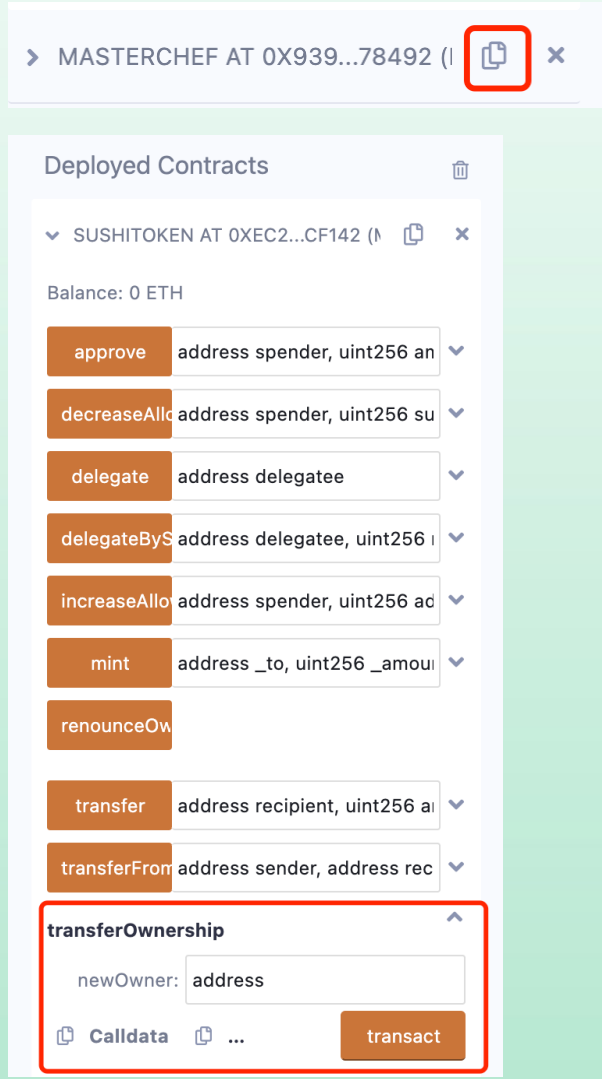
6. 调用MasterChef的add()添加BEP20 的质押池

7. 调用MasterChef的deposit和withdraw进行交互

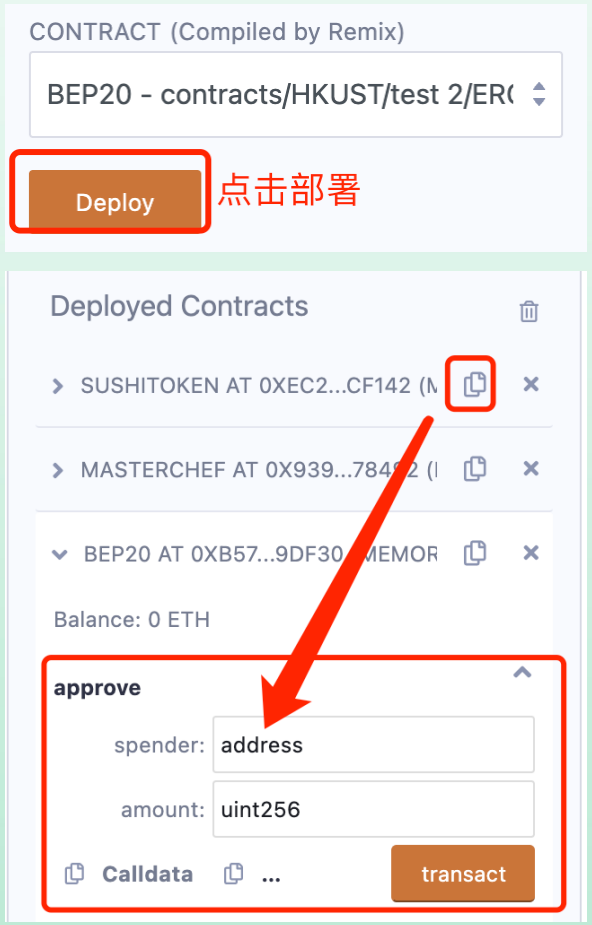
MasterChef部署

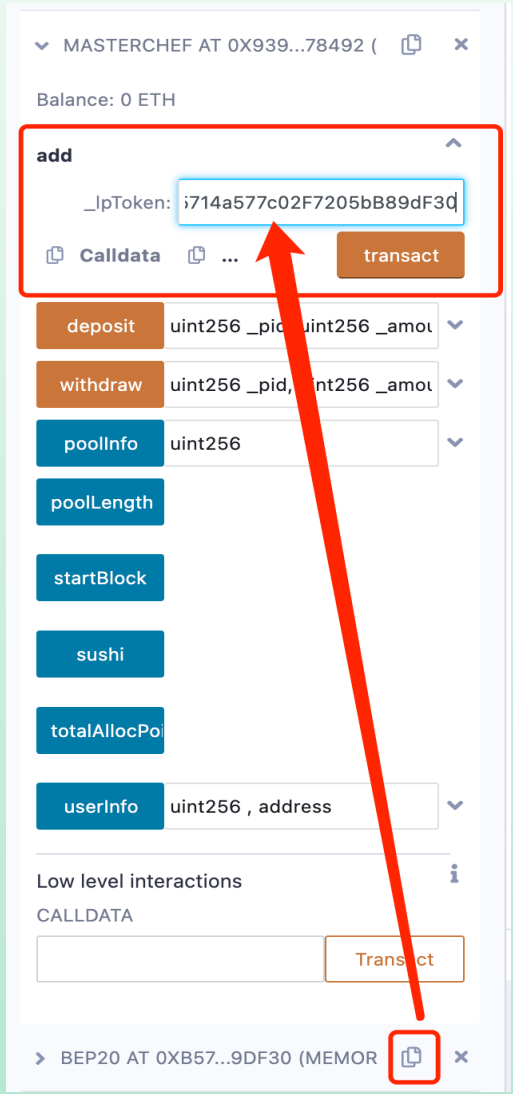


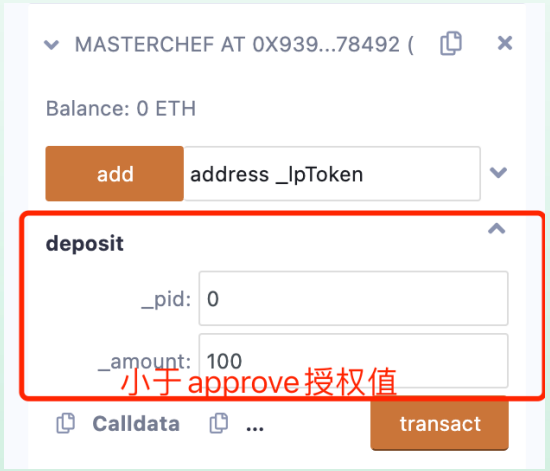




MasterChef部署







BEOSIN
Blockchain Security

课程练习

课程练习

- 完成一个ERC20代币部署到测试网
- 完成流动性挖矿的部署与操作

BLOCKCHAIN ACADEMY × BEOSIN Blockchain Security

About Beosin

Beosin is a leading global blockchain security company co-founded by several professors from world-renowned universities and there are 40+ PhDs in the team. It has offices in Singapore, UAE, Korea, Japan and other 10+ countries. With the mission of "Securing Blockchain Ecosystem", Beosin provides "All-in-One" blockchain security solution covering Smart Contract Audit, Risk Monitoring & Alert, KYT/AML, and Crypto Tracing. Beosin has already audited more than 3000 smart contracts including famous Web3 projects PancakeSwap, Uniswap, DAI, OKSwap and all of them are monitored by Beosin EagleEye. The KYT AML are serving 100+ institutions including Binance.

3,000+
Smart Contracts Audited

$500 Billion
Protected Assets

85,000+
Identified Code Vulnerabilities

1 Million+
Users

1 Billion+
Crypto Addresses Labeled

20+ Years
Cybersecurity Experience

150+ Members
40+ PhD Security Experts

Formal Verification
AI Data Analysis
Over 97% Accuracy

BLOCKCHAIN ACADEMY × BEOSIN Blockchain Security

CONTACT US

- Website: www.beosin.com
- Email: contact@beosin.com
- Official Twitter: twitter.com/Beosin_com
- Alert Twitter: twitter.com/BeosinAlert
- Telegram: t.me/beosin
- Discord: discord.com/invite/B4QJxhStV4
- Linkedin: linkedin.com/company/beosin