Ruferium - RUFE Token Smart Contract

BSC Link:
https://bscscan.com/address/0xb93a79fc01d2d55722e459e2f36a73b9cb1f8e87#code

Technical Documentation for RUFE Token Contract

SafeMath Contract:

The SafeMath contract provides mathematical operations with safety checks to prevent common vulnerabilities such as overflow and division by zero. It includes the following functions:

safeMul: Safely multiplies two uint256 numbers and prevents multiplication overflow.
safeDiv: Safely divides two uint256 numbers and prevents division by zero and division overflow.
safeSub: Safely subtracts two uint256 numbers and prevents subtraction overflow.
safeAdd: Safely adds two uint256 numbers and prevents addition overflow.
RUFE Token Contract:

The RUFE token contract implements the ERC-20 standard and includes additional functionality for token transfers, approvals, and burning. It consists of the following components:

name: A public variable representing the name of the token.
symbol: A public variable representing the symbol of the token.
decimals: A public variable representing the number of decimal places for the token.
totalSupply: A public variable representing the total supply of RUFE tokens.
owner: An address variable representing the owner of the contract.
balanceOf: A mapping that tracks the balance of each address.
allowance: A mapping that tracks the approved allowance for each address.

Events:

Transfer: Triggered when tokens are transferred from one address to another.
Burn: Triggered when tokens are burned, reducing the total token supply.
Constructor:

The constructor function initializes the token contract with the specified initial supply, token name, decimal units, and token symbol. It sets the balance of the contract creator (msg.sender) to the initial supply and assigns the remaining variables.

Token Transfer:

The transfer function allows token holders to transfer tokens to another address. It includes various checks to ensure a valid transfer, such as verifying the recipient address is not zero, the transfer amount is greater than zero, and the sender has sufficient balance. It also prevents overflow by checking the resulting balance after the transfer.

Approval:

The approve function allows token holders to approve another address to spend a specified amount of tokens on their behalf. It updates the allowance mapping for the sender and the spender.

Token Transfer From:

The transferFrom function allows approved addresses to transfer tokens from a specified sender to another address. It checks for valid transfers, including verifying the recipient address is not zero, the transfer amount is greater than zero, the sender has sufficient balance, the allowance is not exceeded, and it prevents overflow.

Token Burning:

The burn function allows token holders to burn a specified amount of their tokens, reducing the total token supply. It checks for a sufficient balance and updates the balance and total supply accordingly.

Ruferium Mining Smart Contract

BSC Link:
https://bscscan.com/address/0xb639655f090b01d703e1172ce689883e1182442d#code

Technical Documentation for Ruferium Mining Contract

MiningContract Interface:

The MiningContract interface allows interaction with the Ruferium mining contract. It includes the following functions:

mine: Allows users to start the mining process by paying a fee of 0.1 BNB.
getTokensMined: Returns the total number of tokens mined by a specific miner based on the elapsed time since they started mining.
withdrawToken: Allows miners to withdraw the tokens they have mined.
withdraw: Allows the contract owner to withdraw the contract's balance.
Events:

Mined: Triggered when a miner successfully starts the mining process, indicating the number of tokens mined.
TokenWithdrawn: Triggered when a miner withdraws their mined tokens.
MiningContract Contract:

The MiningContract contract facilitates the mining of Ruferium tokens. It includes the following components:

owner: An address variable representing the owner of the contract.
token: An address variable representing the Ruferium token contract address.
DEFAULT_MINING_RATE: A uint256 variable representing the default mining rate, set to 1 token per second.
miningStartTimes: A mapping that tracks the start times of mining for each miner.
Constructor:

The constructor function initializes the mining contract with the address of the Ruferium token contract. Only the contract owner can deploy the mining contract.

Mining Process:

The mine function allows users to start the mining process by paying a fee of 0.1 BNB. It checks that the fee is correct and that the miner has not already started mining.
The getTokensMined function calculates the total number of tokens mined by a specific miner based on the elapsed time since they started mining. It checks if the miner has started mining.
The withdrawToken function allows miners to withdraw the tokens they have mined. It calculates the number of tokens based on the elapsed time, transfers the tokens to the

miner, and updates the mining start time. It checks if the miner has started mining and if the token transfer is successful.

The withdraw function allows the contract owner to withdraw the contract's balance. Only the contract owner can call this function.