

DEPLOY & RUN TRANSACTIONS

ENVIRONMENT Reset State

Remix VM (Cancun)

VM

ACCOUNT + 0x5B3...eddC4 (100 ether)

GAS LIMIT

Estimated Gas

Custom 3000000

Select Target Contract

VALUE 0 Wei

CONTRACT SafeMath - contracts/FinToken.sol

evm version: byzantium

Deploy

Publish to IPFS

At Address Load contract from Address

Transactions recorded 0 i >

Deployed Contracts 0

Home FinToken.sol 1 X

@dev Based on code by FIRSTUDIO. https://github.com/FIRSTUDIO/tokenydiode/master/simare_contract/FIRSTTOKEN.sol

```
127
128 */
129 contract StandardToken is ERC20, BasicToken {
130
131     mapping (address => mapping (address => uint256)) internal allowed;
132
133
134 /**
135 * @dev Transfer tokens from one address to another
136 * @param _from address The address which you want to send tokens from
137 * @param _to address The address which you want to transfer to
138 * @param _value uint256 the amount of tokens to be transferred
139 */
140 function transferFrom(address _from, address _to, uint256 _value) public returns (bool) {
141     require(_to != address(0));
142     require(_value <= balances[_from]);
143     require(_value <= allowed[_from][msg.sender]);
144
145     balances[_from] = balances[_from].sub(_value);
146     balances[_to] = balances[_to].add(_value);
147     allowed[_from][msg.sender] = allowed[_from][msg.sender].sub(_value);
148     emit Transfer(_from, _to, _value);
149     return true;
150 }
151
152 /**
153 * @dev Approve the passed address to spend the specified amount of tokens on behalf of msg.sender.
154 *
155 * Beware that changing an allowance with this method brings the risk that someone may use both the old
156 * and the new allowance by unfortunate transaction ordering. One possible solution to mitigate this
157 * race condition is to first reduce the spender's allowance to 0 and set the desired value afterwards:
158 * https://github.com/ethereum/EIPs/issues/20#issuecomment-263524729

```

Target Function