



## DEPLOY & RUN TRANSACTIONS

Transactions recorded **3**

Deployed Contracts **3**

> SAFEMATH AT 0XAE0...96B8B (

> BASICTOKEN AT 0X9D8...A569 (

▼ STANDARDTOKEN AT 0XD4F..

Balance: 0 ETH

approve address\_spender, uint2!

decreaseAppr... address\_spender, uint2!

increaseAppr... address\_spender, uint2!

transfer address\_to, uint256\_va

### TRANSFERFROM

\_from: address

\_to: address

\_value: uint256

Calldata

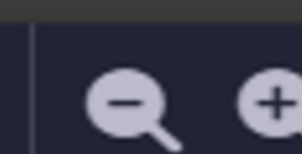
Parameters

transact

allowance address\_owner, address

balanceOf address\_owner

totalSupply



Home

FinToken.sol

1 X

```
125 * @dev Implementation of the basic standard token.
126 * @dev https://github.com/ethereum/EIPs/issues/20
127 * @dev Based on code by FirstBlood: https://github.com/Firstbloodio/token/blob/master/smart_contract/FirstBloodToken.sol
128 */
129 contract StandardToken is ERC20, BasicToken {
130
131     mapping (address => mapping (address => uint256)) internal allowed;
132
133     /**
134     * @dev Transfer tokens from one address to another
135     * @param _from address The address which you want to send tokens from
136     * @param _to address The address which you want to transfer to
137     * @param _value uint256 the amount of tokens to be transferred
138     */
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
```

infinite gas

Target Function

Execute Transaction

[vm] from: 0x5B3...eddC4 to: StandardToken.(constructor) value: 0 wei data: 0x606...70029 logs: 0 hash: 0x6d7...4dc7a