Manual & Usage Documentation: Truffle

Contents

Truffle Initialisation:

(Must be in folder /contract)

Compile the SecuToken Contract & Migrations Contract

```
truffle compile
```

Deploy onto the blockchain

```
truffle migrate --network development
```

Enter development console to execute smart contract commands

```
truffle console --network development
```

SecuToken Contract Setup:

Get the Contract Instance:

```
let SecuToken = await SecuToken.deployed()
```

Get total supply from smart contract:

```
let totalSupply = await SecuToken.totalSupply()
```

[OPTIONAL]

Check total supply of SKTN (in ether value)

```
console.log("Total Supply:",
web3.utils.fromWei(totalSupply.toString(), "ether"), "SKTN")
```

Pre-Defined Admin & User in Blockchain:

Set user address in variable for use later

```
let userAddress =
"0x1f813224C89630dBF0059e7E2Df7879661470cA6"
```

Set admin address in variable for use later

```
let adminAddress =
"0x3B663aAb4fdd872C5B79882eef7159B9C11e0b52"
```

[OPTIONAL]

Get Admin Account address balance

```
let adminBalance = await SecuToken.balanceOf(adminAddress)
```

Check balance of Admin (in ether value)

```
console.log("Admin Balance:",
web3.utils.fromWei(adminBalance.toString(), "ether"),
"SKTN")
```

Get User Account address balance

```
let userBalance = await SecuToken.balanceOf(userAddress)
```

Check balance of User (in ether value)

```
console.log("User Balance:",
web3.utils.fromWei(userBalance.toString(), "ether"), "SKTN")
```

Get the deployed SecuToken smart contract Balance

```
let contractBalance = await
SecuToken.balanceOf(SecuToken.address)
```

Check SecuToken smart contract balance (in ether value)

```
console.log("Contract Balance:",
web3.utils.fromWei(contractBalance.toString(), "ether"),
"SKTN")
```

Redeclare Admin, User & Contract balances to view updated balances

```
adminBalance = await SecuToken.balanceOf(adminAddress)
userBalance = await SecuToken.balanceOf(userAddress)
contractBalance = await
SecuToken.balanceOf(SecuToken.address)
```

Payout Features:

[ADMIN ONLY]

Set 100 Tokens to deposit into the Contract (in ether value)

```
let depositAmount = web3.utils.toWei("100", "ether")
```

Deposit 100 Tokens into the Contract

```
await SecuToken.depositToContract(depositAmount, { from:
adminAddress })
```

Severity Rewards

```
Severity Levels:
0 = Low
1 = Medium
2 = High
3 = Critical
```

Set High Severity Level

```
let severity = 2
```

Set Reward amount for High to 10 SKTN (in ether value)

```
let rewardHigh = web3.utils.toWei("10", "ether")
```

Set mapping of High severity level to 10 SKTN and update contract storage on blockchain

```
await SecuToken.setSeverityReward(severity, rewardHigh, {
from: adminAddress })
```

Reward User Based on High Severity Level

```
await SecuToken.rewardUserBySeverity(userAddress, severity,
{ from: adminAddress })
```

Contract Powers:

[ADMIN ONLY]

Pause the Contract to halt transactions

```
await SecuToken.pause({ from: adminAddress })
```

Unpause the Contract to allow transactions

```
await SecuToken.unpause({ from: adminAddress })
```

Custom:

Replace with the recipient address on the blockchain

```
let recipient = "0xRecipientAddress"
```

Get recipient Account address balance

```
let recipientBalance = await SecuToken.balanceOf(recipient)
```

Check balance of recipient (in ether value)

```
console.log("Recipient Balance:",
web3.utils.fromWei(recipientBalance.toString(), "ether"),
"SKTN")
```

[ADMIN ONLY]

Set Severities for each level

```
let lowSeverity = 0
let mediumSeverity = 1
let highSeverity = 2
let criticalSeverity = 3
```

Set Reward Amounts for each level (in ether value) eg. 1, 5, 10, 20 SKTN

```
let rewardLow = web3.utils.toWei("1", "ether")
let rewardMedium = web3.utils.toWei("5", "ether")
let rewardHigh = web3.utils.toWei("10", "ether")
let rewardCritical = web3.utils.toWei("20", "ether")
```

Set mapping of X severity level to X SKTN reward and update contract storage on blockchain

```
await SecuToken.setSeverityReward(INSERT_SEVERITY,
INSERT_REWARD, { from: adminAddress })
```

Reward Recipient based on X severity level

```
await SecuToken.rewardUserBySeverity(recipient,
INSERT_SEVERITY, { from: adminAddress })
```

Reward Recipient for Multiple Severities eg. Low and Medium

```
await SecuToken.rewardUserBulk(recipient,
[lowSeverity,mediumSeverity], { from: adminAddress });
```

[ADMIN ONLY]

Set Custom Amount of Tokens from the Contract (in ether value) eg. 50 SKTN

```
let amount = web3.utils.toWei("50", "ether")
```

Reward Custom Amount of Tokens from the Contract

```
await SecuToken.sendSKTN(recipient, amount, { from:
adminAddress })
```