## On-line auctions implementation

Author: Alex B

Telegram: @alexbsoft

Forum: <a href="https://forum.freeton.org/u/alex/">https://forum.freeton.org/u/alex/</a>

Ton.surf:

0:7dab3d2563d1d225e646cb6eb0b17752a6dde9db6a91cfcf6c06ba2754af7350

### **Source code & Readme:**

https://github.com/AlexBSoft/TonSimpleAuction

### **Sample Root in Devnet**

0:0eb6fef10aa7f2496c8f43c5b120ae01a2f0661508184b98b9b0c8f44052b709

## **Sample DeBot in Devnet**

0:ed64f712d1c8e263cbaf338944f1a78e8beaa3acc5534c57e759471a342a4868

## **Ton Simple Auction**

This project implements simplified English & Dutch Auctions if Free TON smart contracts.

The system consists of 3 contracts:

- SimpleRoot (aka Auction deployer)
- SimpleAuction (Auction itself, English or Dutch variant)
- DeBot (Auction interface)

## Key idea

Why these Auctions are simple? Because bid is a simple transfer to contract. No need to deploy new contracts for every bid (waste of gas), bid is a just transaction.

It makes easy to place bids <u>using every wallet</u> and client. No need to know Auction contract ABI, just know its address and you are ready.

Also, it makes these contracts more flexible, you can easily integrate it into every project.

#### **Features**

- Flexible
- Scalable
- Easy to use

- Just 200 lines of code
- Transparent
- Events
- Transfers as bids

# **Tests**

Tests made using mocha and.

Run npm run test:common for full test on local node.

# **Deploy**

Run npm run test:deploy for deploy contracts and DeBot to Devnet.

## **Documentation**

For full documentation read Readme.md in project repository on GitHub

### **DeBot**

DeBot allows user to interact with smart contracts. Not finished yet.

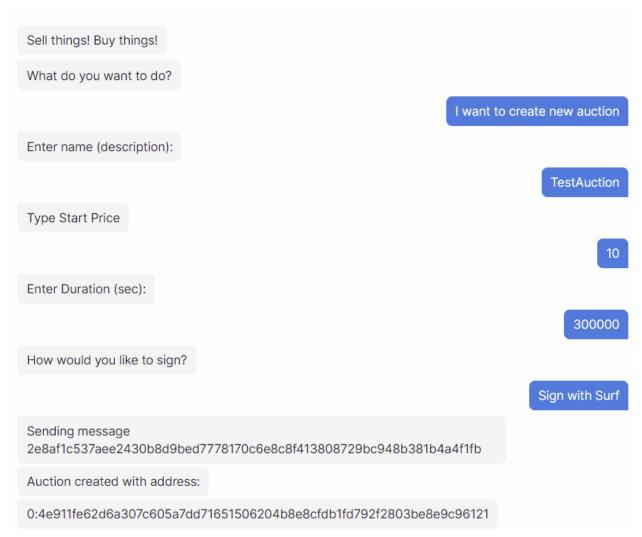


Figure 1 - Auction creation in ton.surf

Figure 2 - DeBot in tonos-cli