RSK Truffle Plant Box¶

Truffle box is configured to create a complete dApp using Truffle framework onRSK Blockchain , including a user interface to interact with the smart contract.

RSK is an open source platform for Ethereum compatible smart contracts based on the Bitcoin network.

It was inspired byTruffle pet shop box . Thanks, Truffle team :)

Requirements¶

There are a few technical requirements before we start. To useTruffle boxes , you need to have installed in your computer:

- Git
- a POSIX compliant shell
- cURL
- Node.js and NPM
- a code editor

If you don't have any of them installed, go to the tutorialTruffle boxes prerequisites which have all the instructions to setup these requirements.

Truffle framework

Once you have those requirements installed, you only need one command to installTruffle . It is better to do it globally:

npm install -g truffle

Installing the Truffle box¶

1. Create a new folder. For example, create the folderrsk-plant
2. . Navigate to the folder in the terminal.

mkdir rsk-plantcd

rsk-plant 1. Run the unbox command. This also takes care of installing the necessary dependencies and it can take some time.

truffle unbox rsksmart/rsk-plant-box This is the result using Windows OS:

PlantShop.sol¶

Take a look at the smart contractPlantShop.sol . You can check it out in foldercontracts .

This smart contract has:

- A variablebuyers
- to store an array with 16 posisions to store addresses
- A functiongetBuyers
- to return the list of addresses stored at variablebuyers
- A functionbuy
- to update an address at variablebuyers
- , in the number of position sent as parameter

Development console¶

Truffle has an interactive console that also spawns a development blockchain. This is very useful for compiling, deploying and testing locally.

1. Run the development console. This command is successful if you see a list of 10 accounts, a mnemonic and the command prompt is nowtruffle(develop)>

truffle develop You will now be in the truffle develop console with seeded accounts and their associated private keys listed.

C:\RSK\rsk-plant>truffle develop

Truffle Develop started at http://127.0.0.1:8545/

Accounts: (0) 0x1056f747cf4bc7710e178b2aeed4eb8c8506c728 (1) 0x45a71c00382c2898b5d6fae69a6f7bfe6edab80c (2) 0x1596384706dc9ac4cca7f50279a4abe591d6c3fe (3) 0x9576d0a496b645baa64f22aceb2328e7468d4113 (4) 0xd431572eef7d77584d944c1809398a155e89f830 (5) 0x92c111839718fe0800fadccc67068b40b8524a0f (6) 0x6da22b5a027146619bfe6704957f7f36ff029c48 (7) 0x2c3a82d8c3993f8c80dcaf91025437bd057df867 (8) 0xc43ae7a44f7deb759177b7093f06512a0a9ff5d7 (9) 0xe61bf00cd7dce248449cfe58f23a4ef7d542bc0b

Private Keys: (0) f32f32839fe27ad906b63eafb326f26fed95c231e3c5e33c7cdd08f62db63167 (1) ebef990088f27f6ef13b5e52a77d5dcc5a76862a701908c586d01b6fe93562b3 (2) 598ccae5e4436fedeb0e798c0d254789c55a63401ebfc3ae8ddde29634ddfcde (3) 09934b80f391e0024b8cb00cd73790fdf64c4d0509e144766414fee317cd3f4e (4) ac745b84b6574b5738d364b43e0d471c9d5107504acc709c90f6f091b78c751b (5) 449654cde095f2349113ef12a93e139b4302bc95adb3619d08adf53dde9b8847 (6) c217f12a89c352fc70b5f1bd5742314b4fb1bb1e35cb779fdb3c2390106355db (7) 1d4c74dfa4e99e161130c18cc63938bb120a128cefbf1b9188efc678bf5722cb (8) 0f44e0becf2e090db498a1b747d2a758fcc81fb0241f350d61117a9c6b1fa82e (9) 85218c5eec657470dafeb09e6f7101f91d21bfe822fbeeecfc9275f798662a63

Mnemonic: virtual valve razor retreat either turn possible student grief engage attract fiber

⚠ Important ⚠ : This mnemonic was created for you by Truffle. It is not secure. Ensure you do not use it on production blockchains, or else you risk losing funds.

truffle(develop)> Inside the development console we don't preface commands withtruffle . 1. Compile the smart contract.

To make sure you're in the development console, the command prompt must betruffle(develop)> compile Thecompile output should be similar to:

1. Deploy (migrate) the smart contract.

migrate And themigrate output should be similar to:

1. Testing the smart contract.

This Truffle box also comes with the fileTestPlantShop.js which include some examples for testing the smart contract. You can check it out in thetest folder.

Run this command in the development console:

test Thetest output should be similar to:

Exit Truffle console¶

In the Truffle console, enter this command to exit the terminal:

.exit

Using RSK networks¶

This Truffle box is already configured to connect to three RSK networks:

1. regtest (local node)
2. testnet
3. mainnet

Testnet will be used here.

We need to do some tasks:

- Setup an account and get R-BTC
- Update RSK network gas price
- Connect to an RSK network
- Deploy in the network of your choose

Setup an account & get R-BTC¶

1. Create a wallet

The easy way to setup an account is using a web3 wallet injected in the browser. Some options are:Metamask -Nifty

Select the RSK Network in the web wallet. - Nifty: select in the dropdown list - Metamask: go toRSK Testnet to configure it inCustom RPC

You can learn more aboutaccount based RSK addresses .

Take a looktruffle-config.js file to realize that we are usingHDWalletProvider with RSK Networks derivations path: - RSK Testnet dpath:m/44’/37310’/0’/0 - RSK Mainnet dpath:m/44’/137’/0’/0

For more information checkRSKIP57 .

1. Update.secret
2. file

After create your wallet, update your mnemonic in the file.secret , located in the folder project, and save it.

1. Get some R-BTCs:
2. For the RSK Testnet, get tR-BTC fromour faucet
3. .

Setup the gas price¶

Gas is the internal pricing for running a transaction or contract. When you send tokens, interact with a contract, send R-BTC, or do anything else on the blockchain, you must pay for that computation. That payment is calculated as gas. In RSK, this is paid inR-BTC . TheminimumGasPrice is written in the block header by miners and establishes the minimum gas price that a transaction should have in order to be included in that block.

To update the TestnetminimumGasPrice in our project run this query using cURL:

curl https://public-node.testnet.rsk.co/ -X POST -H "Content-Type: application/json"

--data '{"jsonrpc":"2.0","method":"eth_getBlockByNumber","params":["latest",false],"id":1}'

> .minimum-gas-price-testnet.json This query saved the details of latest block to file .minimum-gas-price-testnet.json

In thetruffle-config.js , we are reading the parameterminimumGasPrice from this json file.

For more information about theGas andminimumGasPrice please go to thegas page .

Connect to RSK Testnet¶

Run the development console for any RSK network.

truffle console --network testnet

Test the connection to RSK network¶

Run this commands in the Truffle console:

Block number

Shows the last block number.

(await

web3 . eth . getBlockNumber ()). toString () Network ID

To get the network ID, run this command:

(await

web3 . eth . net . getId ()). toString () List of network IDs: - mainnet: 30 - testnet: 31 - regtest (local node): 33

Check it out the last steps in this image:

You can verify that I get the last block twice, and the block number inscreased, so we conclude that the connection is ok.

Exit the Truffle console:

.exit

Migrate the smart contract¶

We will do it running the below commands directly in the terminal, without using the Truffle console, to show you this

alternative.

On any of the networks, run this commands in a terminal (not in Truffle console). To use Testnet or Mainnet, you need to specify this using the parameter-- network :

truffle migrate --network testnet The migrate process in a real blockchain takes more time, because Truffle creates some transactions which need to be mined on the blockchain.

The dApp¶

Included with the plant-shop Truffle Box was the code for the app's front-end. That code exists within thesrc directory.

Make sure you have selected the RSK testnet in the wallet.

Running the dev server¶

Now we're ready to use our dapp!

Start the local web server:

npm run dev The dev server will launch and automatically open a new browser tab containing your dapp.

It is running athttp://localhost:3000

Buying plants¶

In our garden store, don't worry about the prices, the plants are free!

Click theGet button on the plant of your choice.

You'll be automatically prompted to approve the transaction by the web wallet. Click submit / confirm to approve the transaction.

After the transaction is confirmed, you'll see the button next to the choosed plant change to show the first characters of the wallet that got the plant and become disabled, just as we specified, because the plant has now been acquired.

Congratulations ! You built and ran a complete dApp on RSK network!

Do you have questions?¶

Ask in theRSK chat .