

## Guide ZHCASH Beginner 's Guide

The plan was admittedly excellent:  
simple and clear, it's the best thing to do.

He had only one drawback:  
it was completely unknown,  
how to enforce it.

Alice in Wonderland

If I had this guide,  
then I would have finished the smarts months earlier

Author

### DISCLAIMER

This version of the guide is unofficial and made in the art house style  
Revision 1.3 as of 19.01.2023

This guide describes the nuances of writing smarts under zh, possible mistakes of novice developers, how to work with a node via the terminal and python, launching your local test blockchain, nuances of a web wallet, possible RPC errors and installing a node via ssh on the ubuntu server

Definitions:

- 1) zh – hybrid blockchain ZHCASH. The result of crossing (love) of bitcoin and ether, which has undergone gene modifications <https://zh.cash://zh.cash/> .  
Pronounced "zh". It is the opposite of "xs".
- 2) zx, shekel — the main ZHC coin of the ZHCASH blockchain ZHCASH.
- 3) Zeroshka-an analog of Satoshi in bitcoin. Equal to  $10 \cdot 8^{-8}$  zx
- 4) lift - token of the LIFT project
- 5) ZRC20-analog of ERC20 and QRC20 (QTUM)standards
- 7) QTUM-parent of zh <https://qtum.org/en://qtum.org/en>. Older by 2 years. The most promising blockchain in the world (from the Chinese) was taken at the time of 2019 and significantly improved to the world's best zh. There is good support in Telegram, where you will get a quick answer to any question.
- 8) Console – a command line in the zh wallet terminal or a program with a zerohour-cli command-line interface. This is not a website <https://zhcash.org://zhcash.org/>. It is

used for entering commands and interacting with the blockchain. The API is available on the site <https://zh.cash/docs/en/ZHCash-RPC-API/>

8\*) Web Console-site <https://zncash.org://zncash.org/>.

9) hex — real wallet address in HEX format. It is used for calculating tokens and shekels when interacting through a smart contract. It turns out when you enter the command

```
gethexaddress ZEFnGiHuwDStHnBA3cvAgPPFhhAKKqXQna
```

go to the console. It turns out

```
184eb41e30b0d5974df3d1b2429fbdf728222a4c
```

This is almost an Ethereum address, except that it is not preceded by 0x. You can't use smart in the code (it doesn't compile). If you add 0x at the beginning, it's not considered a wallet in zh . Use only this type of wallet (without the 0x at the beginning), but do not use it directly in the smart code itself.

10) Remix-an ethereum environment for smart development

<https://remix.ethereum.org://remix.ethereum.org/>

12) Broken-smart (or тразакцияtransaction), which did not have enough gas and did not integrate into the blockchain. Displayed in black in explorer (zeroscan). For one batsman, two unbroken ones are given, or vice versa.

12) Explorer (zeroscan) — <https://zeroscan.io://zeroscan.io> its api is described at the end

13)Solidity smart contract на solidity

14) Nachilit (nachil)- get native coins from the testnet, token interest , or block mining rewards. The main condition is that this happens at the raceclab

15) Drop-centralized distribution of tokens or shekels at the beginning of the smart launch. It is short for air drop.

16) UTXO (Unspent Transaction Output) - output of unspent transactions

<https://2://2bitcoins.ru/chto-takoe-utxo-i-zachem-on-nuzhen.ru/chto-takoe-utxo-i-zachem-on-nuzhen/>

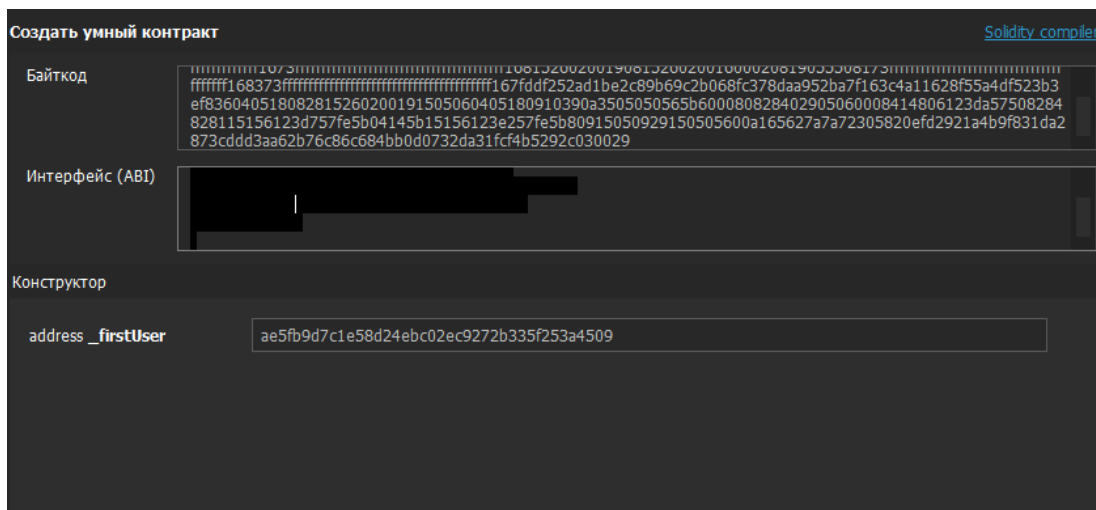
When sending 1 zx from the balance of 10 zx, the entire amount is sent to the address, and the change of 9 zx is returned in the next block. It does not allow you to make many transactions in the blockchain, but also increases the security of the blockchain. Used in Bitcoin. On-air uses an Account System. A comparison of these systems is described here <https://russianblogs.com/article/24511021659/>

## Smart development for zh

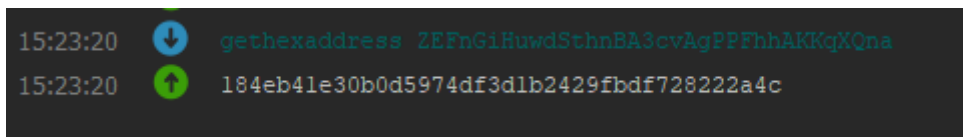
It is assumed that the reader already has an initial level of knowledge on solidity, which they can learn for example here <https://inaword.ru/smart-kontrakty/> or [https://www.tutorialspoint.com/solidity/solidity\\_variables.htm](https://www.tutorialspoint.com/solidity/solidity_variables.htm)

Система смартов The zh smart card system is identical to Ethereum, but there are some nuances. The address in zh is the same ether address, but without the 0x. However, this type of address cannot be specified directly in the remix code, so if we want to pass the value of the address (for example, the first user to be credited with a million tokens), we can do this when creating the contract. In this case, enter the address in hex format

You can get hex via the



gethexaddress console gethexaddress(take a closer look at the first line in the screenshot below)



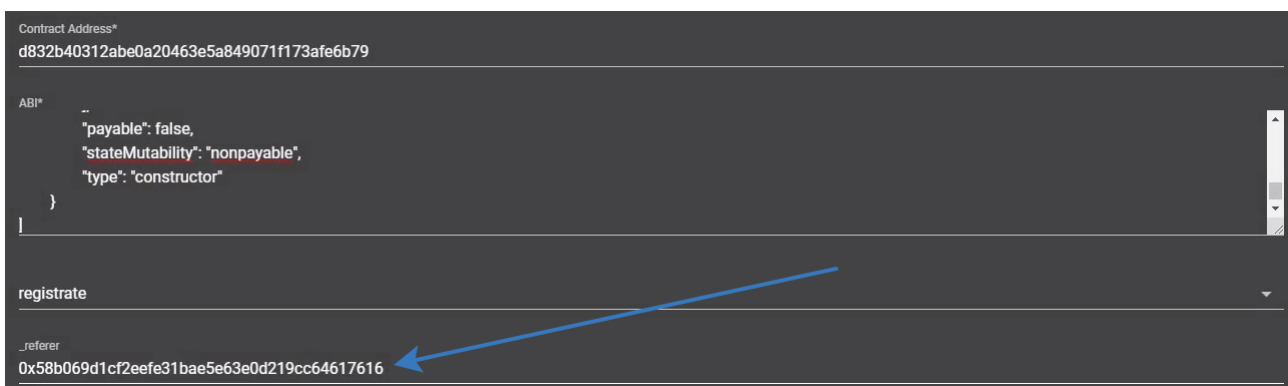
Using the my smart function to get your wallet address, note that it will also be in hex format

Contract Summary	
ContractAddress	0a2b496fa9ea27e3d2950e9d1b5056bb751ae3eb
Function	getMyAddress()
SenderAddress	
Result	
address	184eb41e30b0d5974df3d1b2429fbd728222a4c

You can use the `fromhexaddress` command to convert hex to the classic view (take a closer look at the first line in the screenshot below)

```
21:00:24 [down arrow] fromhexaddress 184eb41e30b0d5974df3d1b2429fbd728222a4c
21:00:24 [up arrow] ZEFnGiHuwDStHnBA3cvAgPPFhhAKKqXQna
```

In the web wallet, by the way, when sending any wallet to smart in the form of data in addition to hex format, you also need to add `0x` as in the good old classic



With `aidrop`, you should not put more than 50 transactions in one block. Otherwise, the entire amount on the balance sheet is spent. 20 transactions are safe.

If the contract turns out to be broken (after it is unloaded, there is no "extracted" icon in transactions and `зеросканейт` will be displayed in black on the `zeroscan` in the smart block), then you should increase the gas. It is recommended to bet 20% more than shown in the remix

```

[vm] from: 0x5B3...eddC4 to: HotHeads.(constructor) value: 0 wei data: 0x608...c02db logs: 0 hash: 0xb75...a694b

status      true Transaction mined and execution succeed
transaction hash  0xb75738edf8edff56ceca55ef3b1b136084346c31cc1b965430632028e5ea694b
from        0x5B38Da6a701c568545dCfcB03FcB875f56beddC4
to          HotHeads.(constructor)
gas         888347 gas
transaction cost  728997 gas
execution cost  728997 gas

```

Also specify which wallet the smart card will be created from

Опции

Количество	<input type="text"/>	ZHC
Лимит Gas	<input type="text" value="1000000"/>	Цена Gas <input type="text" value="0.00000040"/>
Адрес отправителя	<input type="text" value="ZEFnGiHuwDSthnBA3cvAgPPFhhAKKqXQna"/>	

ОЧИСТИТЬ ВСЕ    ОТПРАВИТЬ КОНТРАКТ

Normal situations are shown below.

04.06.2022 22:16	Добыто	(ZbFkrDeGDr6EW1UyXBCGV7vSojEgpLF8fm)	[0.26905840]
04.06.2022 22:13	Отправка контракта	(fea82863035a4a5edf5fe8434a39b17e5ab22a05)	-1.44250400

469e8e0ff1aa6888d3de869572cbaca3d2727405e5883877f74cddbfb4453c1	3 confirmations	2022-06-04 22:17:04
ZbFkrDeGDr6EW1UyXBCGV7vSojEgpLF8fm	495.45034960 ZHC	Contract Create 494.00784560 ZHC
Gas Back		0.26905840 ZHC
Mint Tokens		999000000 TESTF9
Mint Tokens		1000000 TESTF9
		Fee 1.1734456 ZHC

You can find out the required gas in the remix by expanding the transaction data. But when you try to send the transaction shown in the picture below, with gas 50000 it failed, but with gas 100000 it passed. Recommended gas for any transactions in zh 250000, in qtum 100000.

```

[vm] from: 0xAb8...35cb2 to: ZRC20Token.addNewUser(address,uint256) 0xd91...39139 value: 0 wei data: 0x10f...0000a logs: 1 hash: 0xa8f...9137c
status true Transaction mined and execution succeed
transaction hash 0xa8f451b9a13bca776d52ce8eb49f28fdad66d19067d6f92ab220de917e
from 0xab8433f649c641c2f9b9493a477d0315935cb2
to ZRC20Token.addNewUser(address,uint256) 0xd91450CE52D386E254917e491e84e9943f9139
gas 42005 gas ← МИНИМАЛЬНО НЕОБХОДИМЫЙ ГАЗ
transaction cost 36026 gas
execution cost 26526 gas
input 0x10f...0000a ← КОМАНДА ДЛЯ КОМАНДНОЙ СТРОКИ
decoded input {
  "address": "0x4B20995B461177e07E8F571c0cE3A9a22C02db",
  "uint256": "10"
}

```

You can also find out exactly what command (command code) is being executed there. This is necessary for interacting with the blockchain via the console.

Zh supports the latest version of solidity (it may not work correctly), but we recommend using version 0.4.18 (since the ABI in later versions does not support accessing smart via callcall). In the first contracts, I used version 0.4.18, as in the standard qtum QRC20 Token example

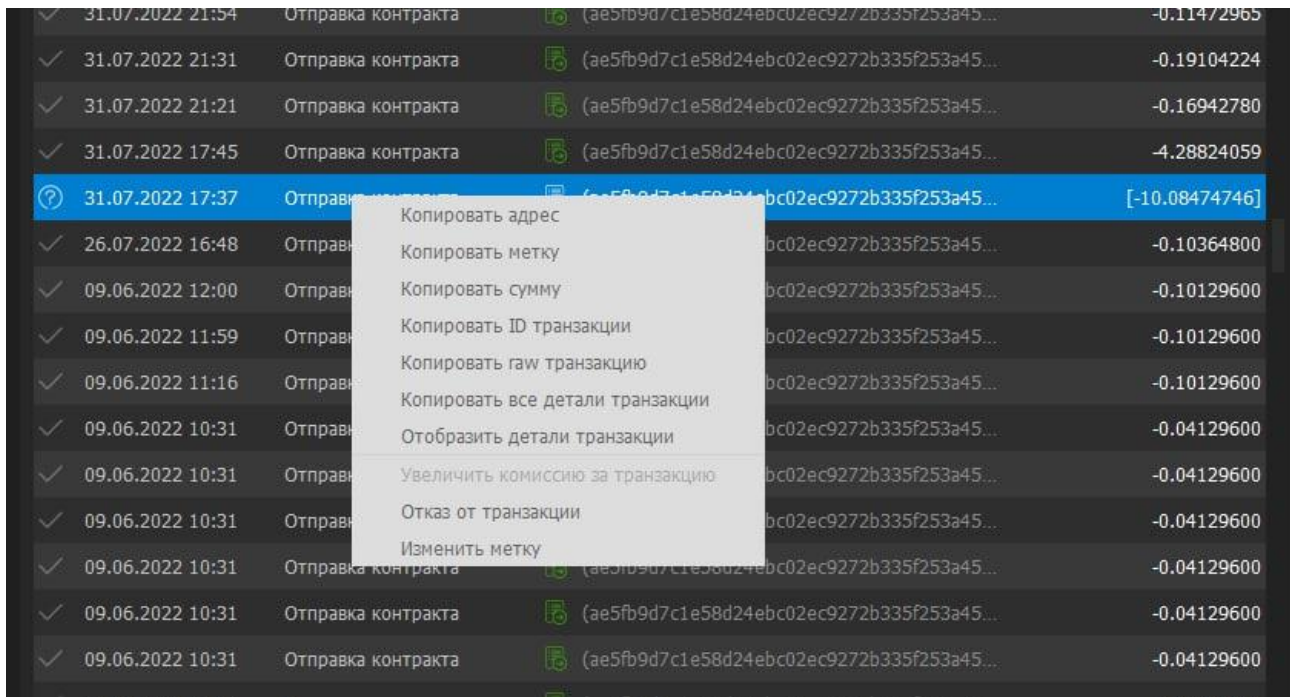
<https://docs.qtum.site/en/QRC20-Token-Introduce.html> , because I didn't change the standard gas value.

And when compiling on versions 0.7, even the standard example of smart beats if you leave the gas 250000.

It is recommended to upload smart cards and send data (sendtocontract) to smart cards with no more than 20 shipments per block, if you have set the default values in the wallet settings, otherwise some transactions may freeze, and the entire balance is included in them. Next, we will show you how to remove one check mark in the wallet settings and this immediately fixes this problem by 99%. The author has repeatedly encountered this. This rule does not apply to calling contract functions (callcontract) and sending individual tokens to anyone. It was noticed that when you try to cram more than 30 transactions into one block from one wallet, a strong deduction of shekels from the balance (from 1 thousand to 400 thousand) begins. So дрон you can merge the entire node balance (1 million) to a drop, sending tokens to 60 users in one block. As a result, the author made a drop message to 20 users in one block (with a delay of 10 minutes). 200 users are sent per hour, which is acceptable. For 4 hours drop nachililsya all. Merged zx can be returned by canceling the suspended transaction. Still, there are ways to circumvent this restriction and transfer tokens to thousands of addresses instantly (by splitting your wallet into many small ones for one shekel, or using the so-called "batch transactions" senddrawtransaction <https://zh.cash/docs/en/ZHCash-RPC-API/#senddrawtransaction> ), or write a smart that will receive 3xzx and make shipments to the specified addresses (smart can make a batch transaction), but we will leave this is an exercise for the reader.

This is due to the fact that the delivery according to the UTXO algorithm should be returned in the next block and the entire balance may be used up. Due to UTXO (or my programs), it may no longer be displayed in the total wallet balance (the transaction may freeze) So 400 thousand shekels were lost in 5 minutes. You can

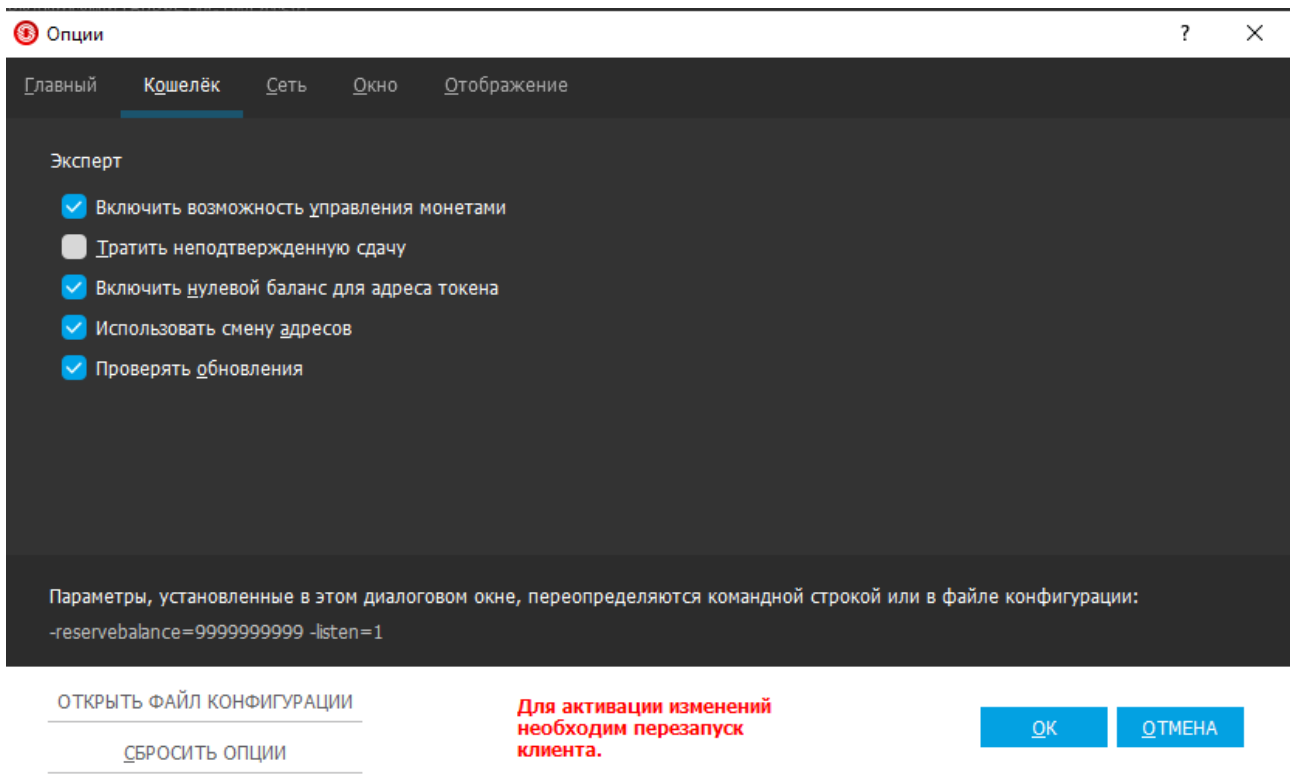
cancel a suspended transaction at any time (even after several months) by clicking on "Cancel transaction" and after cancellation, the lost amount is returned to the balance. This transaction is displayed under the question (instead of the check mark on the left).



The screenshot shows a list of transactions in a wallet interface. The transactions are listed with a checkmark on the left, a date and time, a description, a transaction ID, and a balance change. One transaction, dated 31.07.2022 17:37, is highlighted in blue and has a question mark icon instead of a checkmark. A context menu is open over this transaction, listing several actions: 'Копировать адрес', 'Копировать метку', 'Копировать сумму', 'Копировать ID транзакции', 'Копировать raw транзакцию', 'Копировать все детали транзакции', 'Отобразить детали транзакции', 'Увеличить комиссию за транзакцию', 'Отказ от транзакции', and 'Изменить метку'.

Icon	Date/Time	Description	Transaction ID	Balance Change
✓	31.07.2022 21:54	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.11472965
✓	31.07.2022 21:31	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.19104224
✓	31.07.2022 21:21	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.16942780
✓	31.07.2022 17:45	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-4.28824059
?	31.07.2022 17:37	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	[-10.08474746]
✓	26.07.2022 16:48	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.10364800
✓	09.06.2022 12:00	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.10129600
✓	09.06.2022 11:59	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.10129600
✓	09.06.2022 11:16	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.10129600
✓	09.06.2022 10:31	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.04129600
✓	09.06.2022 10:31	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.04129600
✓	09.06.2022 10:31	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.04129600
✓	09.06.2022 10:31	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.04129600
✓	09.06.2022 10:31	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.04129600
✓	09.06.2022 10:31	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.04129600
✓	09.06.2022 10:31	Отправка контракта	(ae5fb9d7c1e58d24ebc02ec9272b335f253a45...	-0.04129600

In order to spam the blockchain, it was enough to uncheck the box in the settings, the Wallet section, and the "Spend unconfirmed change" box.

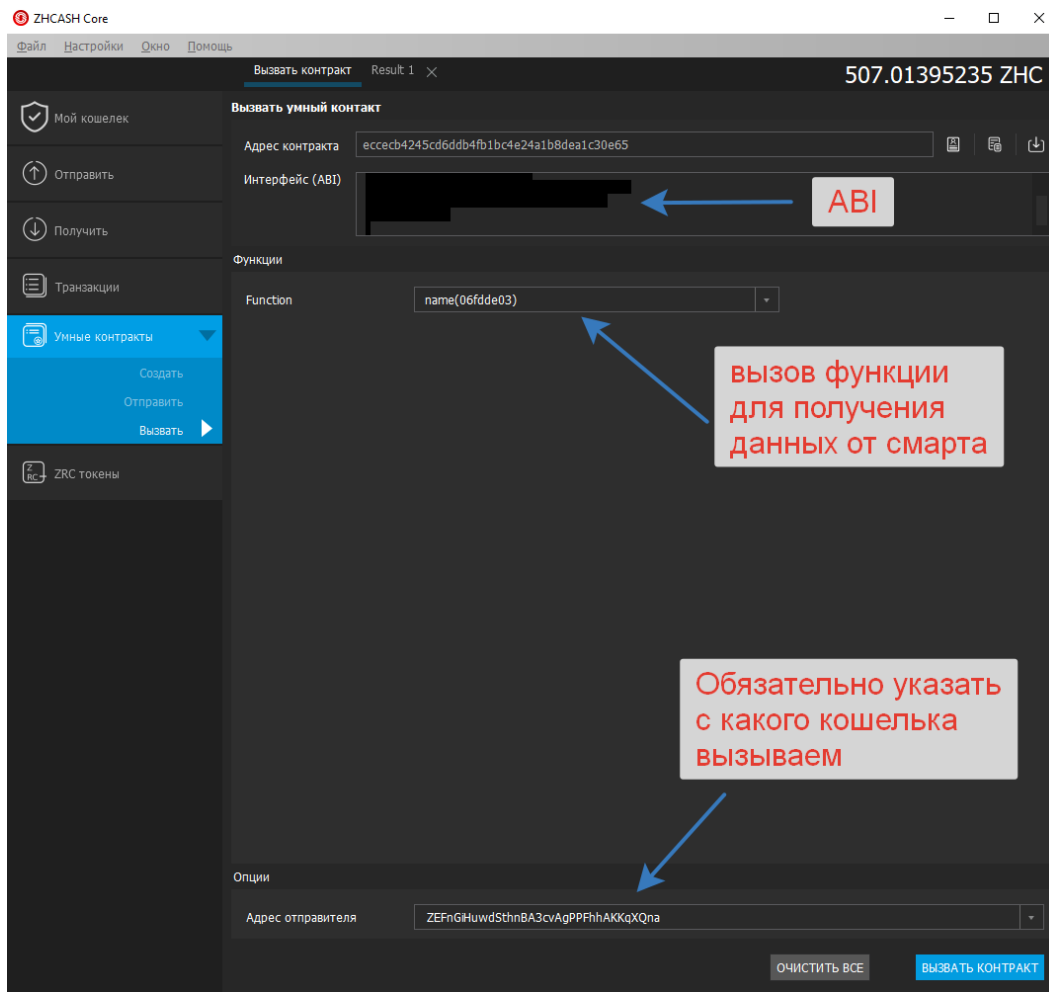


But still, with continuous spam, a situation may arise when some transactions hang with all funds. So far, a complete solution to this issue has not been found. Somehow it is necessary to cancel suspended transactions.

The record for spam can be found here <https://zeroscan.io/block/771654>

Let's look at how to exchange information with smart. Let's first look at how to interact with smart through the graphical interface of the wallet. Then we'll look at how to do this from the command line.



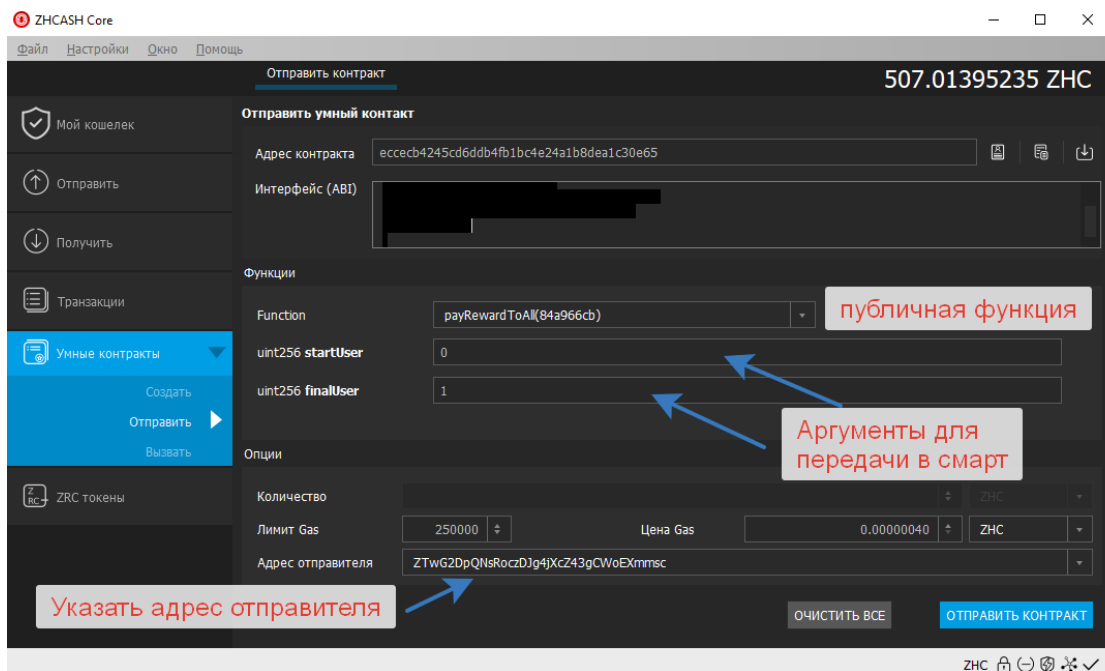


If we want to get data, we should use the "Call" tab. We get the following result.

Contract Summary	
ContractAddress	eccecb4245cd6ddb4fb1bc4e24a1b8dea1c30e65
Function	name()
SenderAddress	
Result	
string	TESTF9

So we can get any value of a public variable or execution of an external function (external view).

If we want to send data, we should use the "Send" tab



If we want to pay rewards to all users, we should also pay them in chunks of 20 transactions per block. And put a larger gas value. After such a request, we get the following.

Contract Summary	
Transaction ID	9972da26f233d7e963aa1ca93eef7bb341c2a89571c2762545d05c92b8f698f8
SenderAddress	ZTWG2DpQNsRoczDjg4jXcZ43gCWoEXmmsc
Hash160	ae5fb9d7c1e58d24ebc02ec9272b335f253a4509

After waiting for a new block and going to "Transactions", we will notice that the "Extracted" icon will appear

05.06.2022 16:03	Добыто	(ZL25qnzUA7uz7kQBYC3vmZAt8jgS19UXwL)	[0.07348320]
05.06.2022 16:02	Отправка контракта	(57884d4449512ede1e192415c8cf4029969fd5f0)	-0.10188400

This means that the data was sent successfully.

Now let's look at interacting with the blockchain via the console.

To do this, there are two commands: `callcontract` to receive data and `sendtocontract` to send data to smart. Below is an example of how to use it.

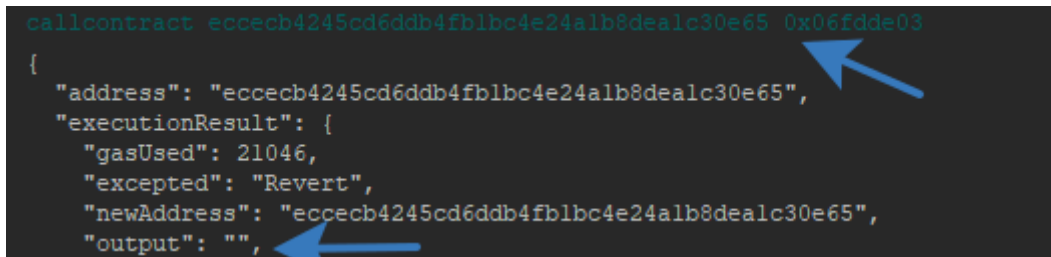
```
sendtocontract efa851007505513c91c0f80e3e9f571a544d5245
a9059cbb00000000000000000000000000000001c96419179e5d6e2cb98b0b2c
a658daff6d12cb5000000000000000000000000000000000000000000000000
000000000000003b9aca00 0 100000 0.0000004
ZTwG2DpQNsRoczDJg4jXcZ43gCWoEXmmsc
```

```
callcontract eccecb4245cd6ddb4fb1bc4e24alb8dealc30e65
06fdde03
```

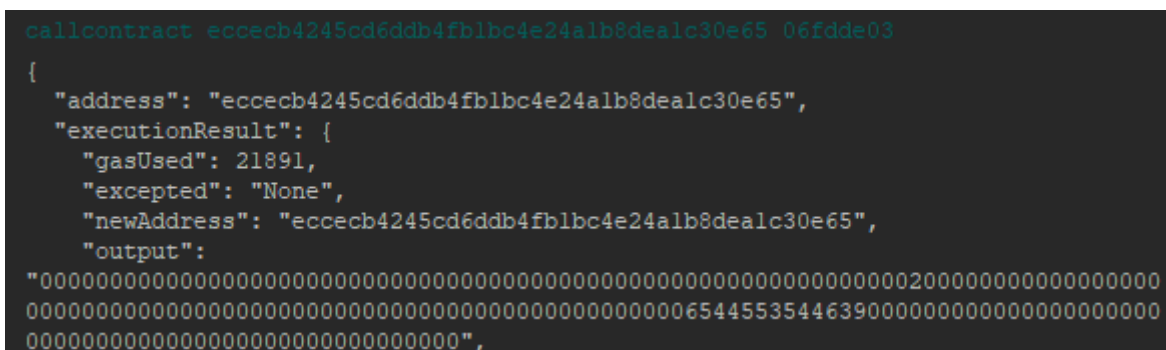
The command for a transaction via the console can be obtained from the remix



Do not write 0 x in the command name. In zh , this is incorrect. This way you won't get anything. You should write what comes after 0x.

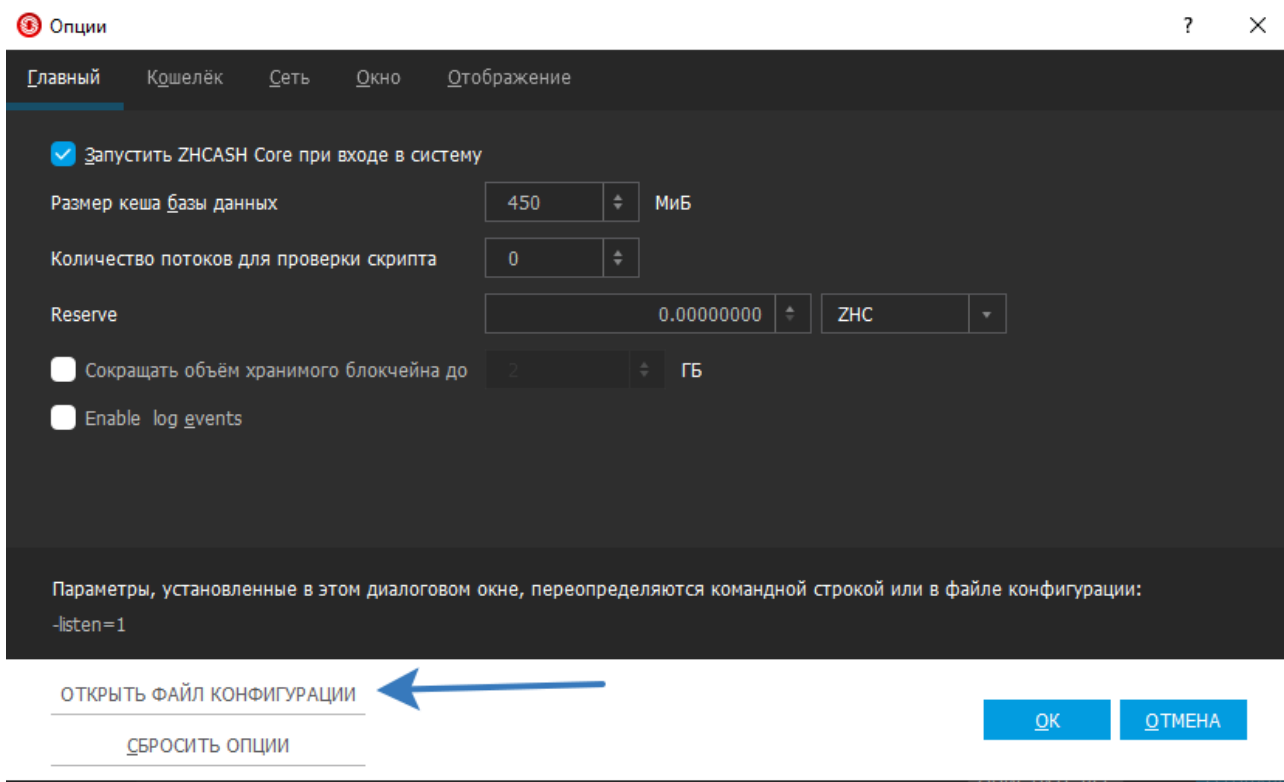


This is the correct way:



We will get some data that can then be distilled into a string. Below is a screenshot for sendtocontract



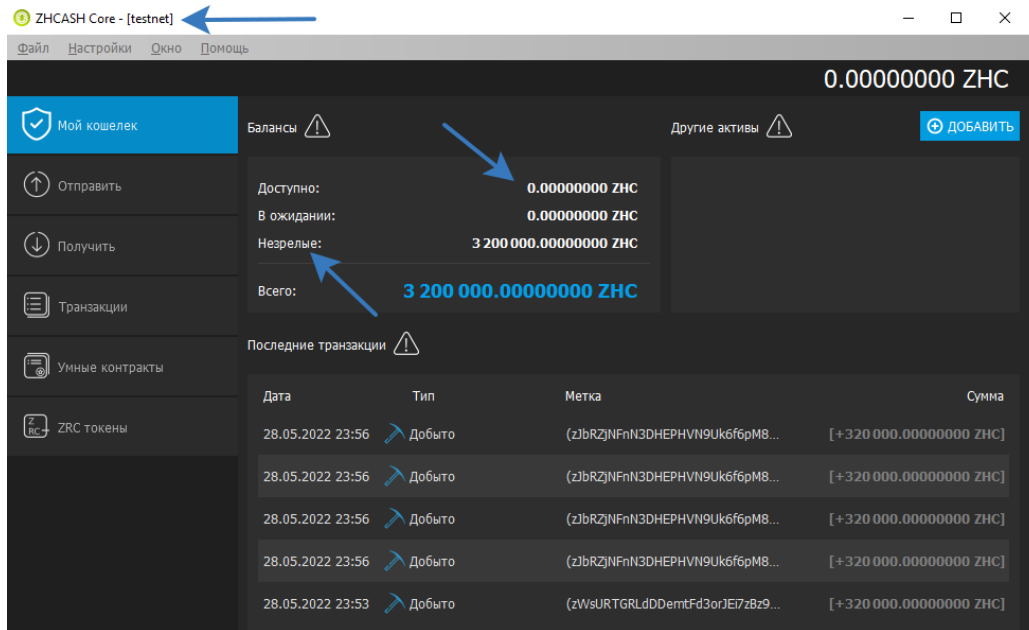


And save the following text, where `rpcpassword` you can set your wallet password in the last argument of `rpcpassword`

```
accounting=1
server=1
daemon=1
gen=0
irc=0
rpcport=3889
port=8003
listen=1
#staking=0
#rpcbind=17.2.7.11
#reservebalance=9999999999
#rpccallowip=17.2.7.12
#rpccallowip=17.2.7.11
rpccallowip=127.0.0.1
rpcuser=zerohour-rpcuser
rpcpassword=
```

```
accounting=1
server=1
daemon=1
gen=0
irc=0
rpcport=3889
port=8003
listen=1
#staking=0
#rpcbind=17.2.7.11
reservebalance=9999999999
#rpccallowip=17.2.7.12
```

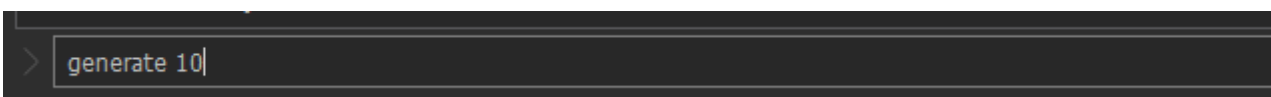




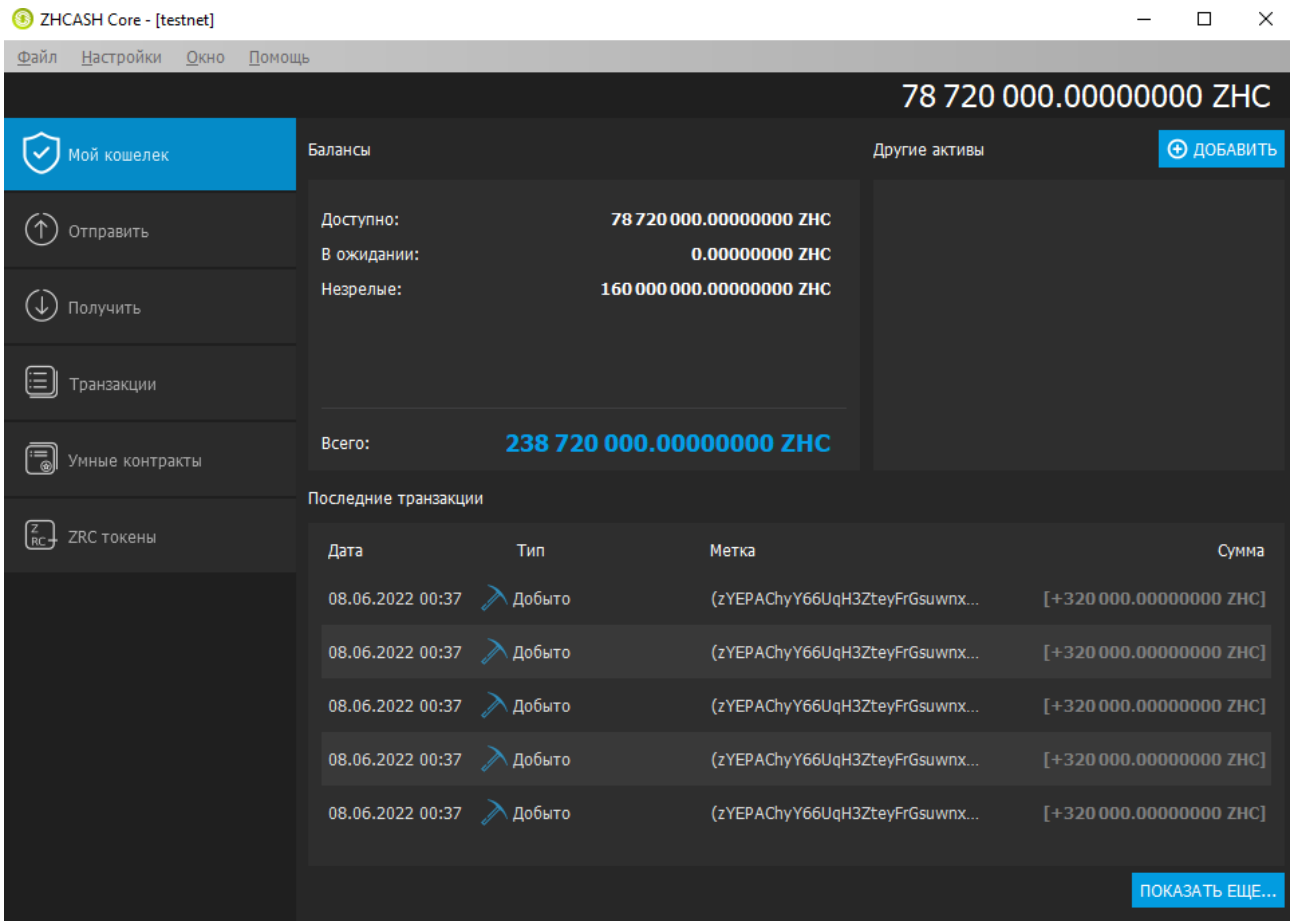
We go to the console, write generate 10 and get an error. In order to generate blocks, you need to run with the keys-testnet -deprecatedrpc=generate. Repeat again.

```
PS C:\Users\Yoga\Desktop> .\ZeroHour-Qt.exe -testnet
PS C:\Users\Yoga\Desktop> .\ZeroHour-Qt.exe -testnet -deprecatedrpc=generate
PS C:\Users\Yoga\Desktop> █
```

We poke консолетhis command in the console many,many times, until something happens to the balance in the "Available" section. Unripe shekels make the blockchain indigestible.

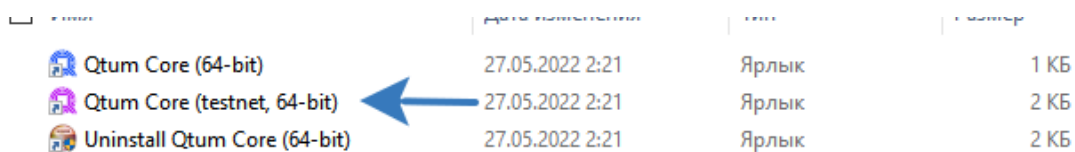


After repeatedly pressing the up key (to repeat the previous command) and enter , the blockchain is split up and ready to go



You can load smart cards, check everything, then generate a block and test everything.

2) Using the QTUM test network. They have a tap for the test network, where you can enter your address and you will be credited with  $50 \pm 20$  test coins. When downloading a wallet, a separate testnet wallet is immediately available.



To synchronize with the test blockchain takes about 2 hours to synchronize with the testing blockchain. Guide on testnet qtum testnet Guide <https://docs.qtum.site/en/Testnet-User-Guide.html>

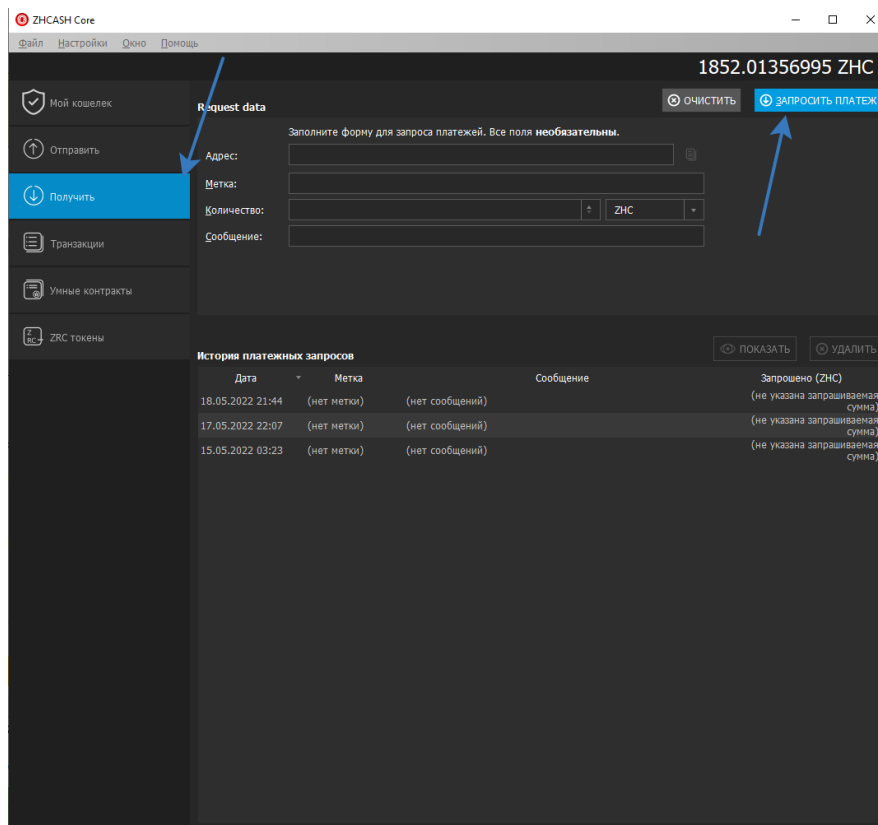


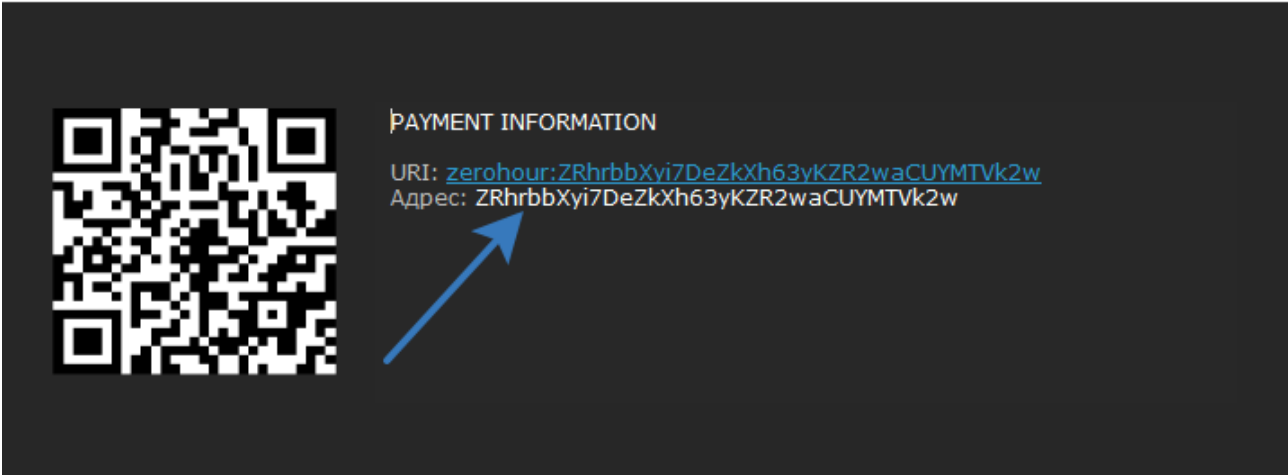
3) Creating dozens of test smarts on the main network (as the author did at the beginning. So he decided to write a guide), but this is frowned upon. The result of the third approach in testing is shown below.

All  ZRC TEST10 (ZRC10)  TESTF8 (TESTF8)  QRC FINAL (QRCF)  ZRC TEST10 (ZRC10)  QRC TEST6 (QRC6)  ZRC TEST 13 (ZRC13)  ZRC TEST10 (ZRC10)  QRC TEST6 (QRC6)  QRC TEST (QTC)  ZRC TEST10 (ZRC10)  QRC TEST6 (QRC6)  ZRC TEST 3 (ZRC3)  QRC TEST6 (QRC6)  ZRC TEST 3 (ZRC3)  QRC TEST6 (QRC6)  QRC TEST6 (QRC6)  QRC TEST6 (QRC6)  ZRC TEST10 (ZRC10)  QRC TEST6 (QRC6)  ZRC TEST 2 (ZRC2)  TEST FINAL (TESTF)  TEST FINAL 4 (TESTF4)  QRC TEST6 (QRC6)  ZRC TEST 11 (ZRC11)  ZRC TEST10 (ZRC10)  QRC TEST5 (QRC5)  ZRC TEST10 (ZRC10)  QRC TEST6 (QRC6)  QRC TEST6 (QRC6)  ZRC TEST 2 (ZRC2)  TEST FINAL (TESTF)  ZRC TEST 3 (ZRC3)  ZRC TEST 3 (ZRC3)  TEST FINAL 6 (TESTF6)  TEST FINAL 3 (TESTF3)  ZRC TEST 3 (ZRC3)  ZRC TEST (ZRC)  QRC TEST6 (QRC6)  TEST FINAL 5 (TESTF5)  ZRC TEST 3 (ZRC3)  QRC TEST6 (QRC6)  QRC TEST6 (QRC6)  QRC TEST6 (QRC6)  QRC TEST6 (QRC6)  ZRC TEST 3 (ZRC3)  ZRC TEST10 (ZRC10)  ZRC TEST 3 (ZRC3)  QRC TEST4 (QRC4)  QRC TEST6 (QRC6)  TESTF9 (TESTF9)  TEST FINAL (TESTF)  QRC TEST6 (QRC6)  ZRC TEST9 (ZRC9)

## Getting the wallet address

To get your wallet number, go to the "Get" tab and click the "REQUEST PAYMENT" button





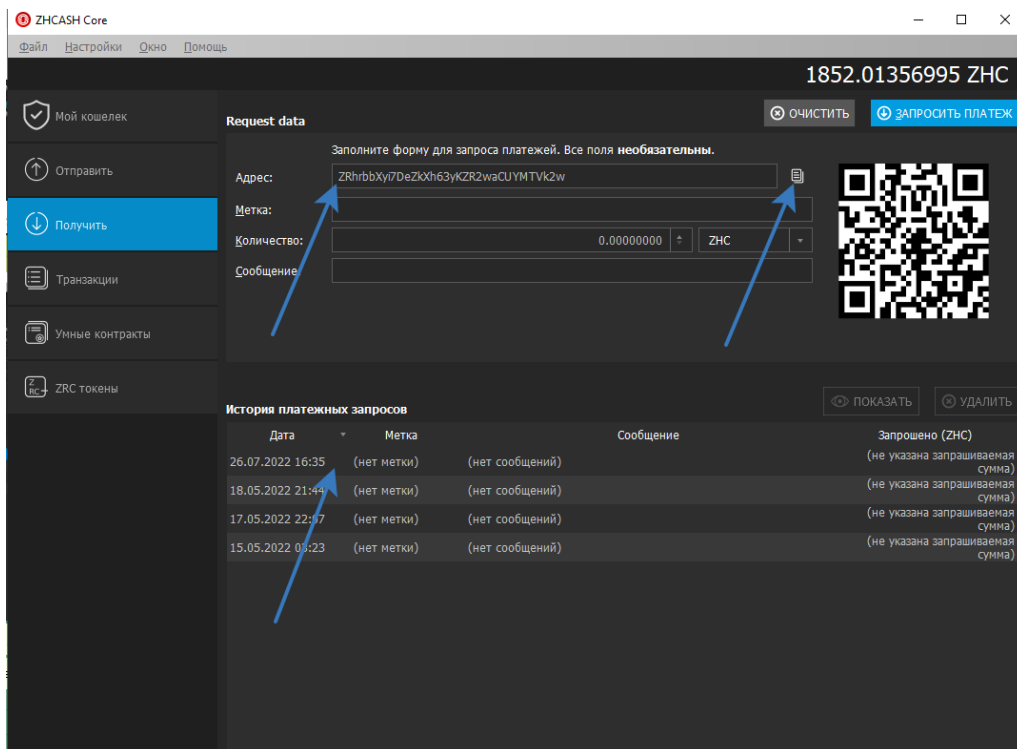
КОПИРОВАТЬ URI

СОХРАНИТЬ ИЗОБРАЖЕНИЕ...

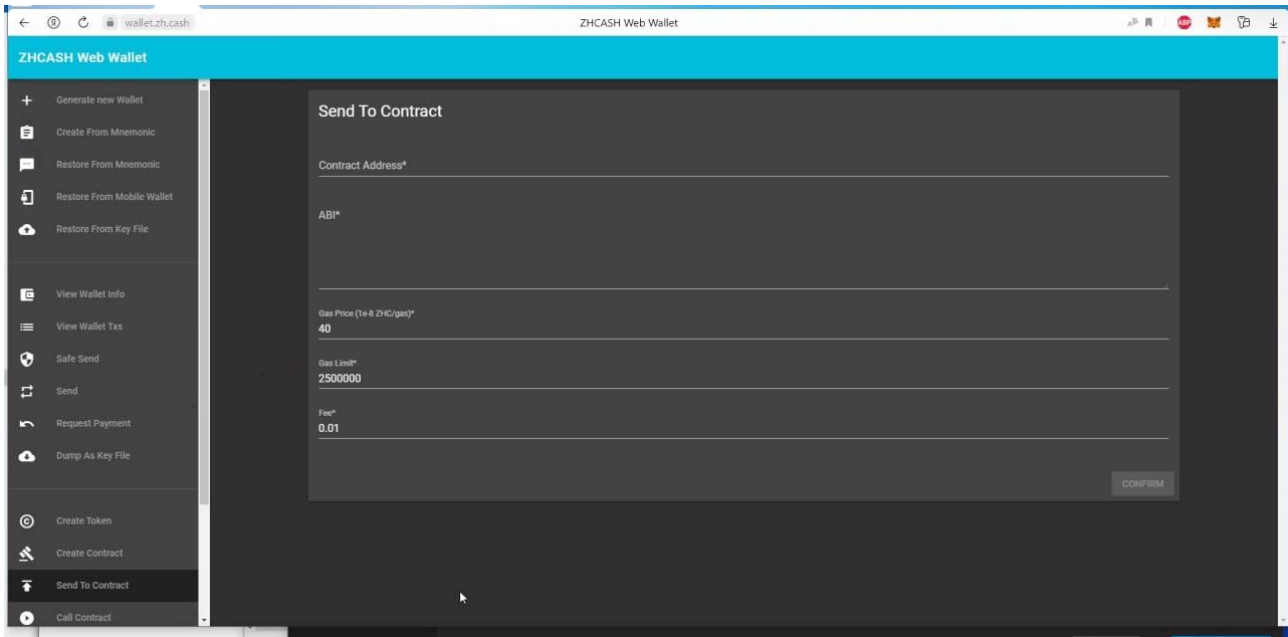
КОПИРОВАТЬ АДРЕС

CLOSE

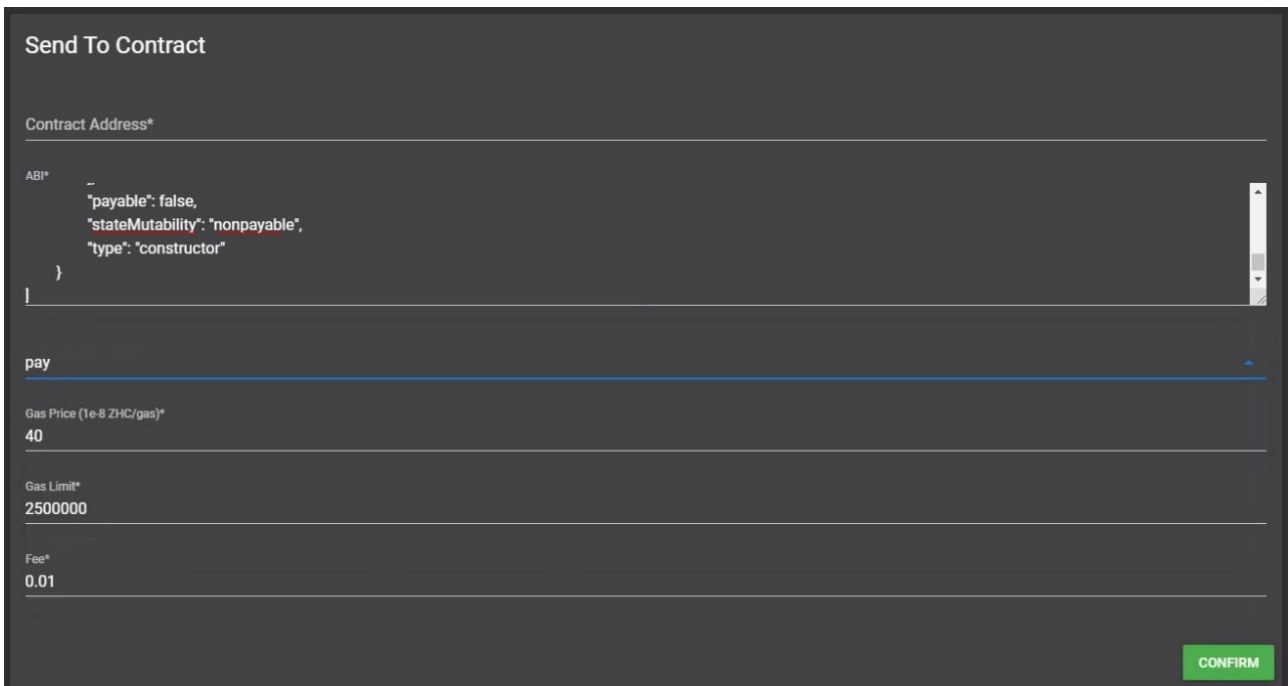
To copy the wallet number again, select a line in the payment request history and copy the Address that appears



# WEB WALLET



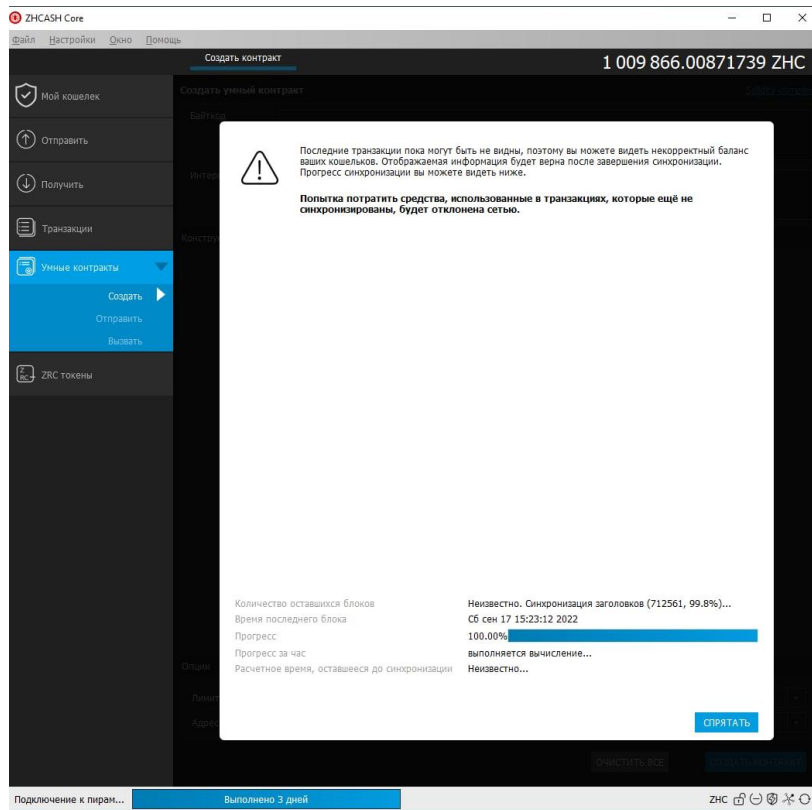
On the web version you can't send a po with the sendtocontract command



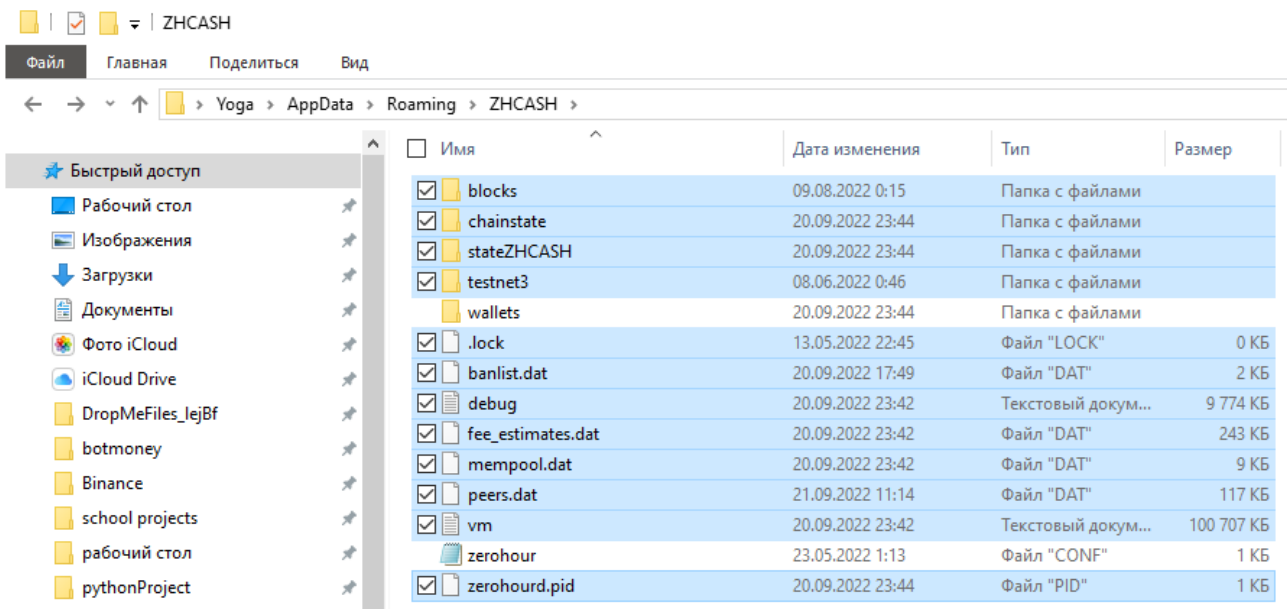
And in general, you can't send  $3x$  на  $2x$  to a smartcontract. This feature has not been added yet (as of 09/21/2022).

Otherwise, the web wallet has the same functionality as the node.





To start a new synchronization, delete all files from the ZHCASH folder ZHCASH (the same folder where the configuration file is located), except for the wallets folder and the zerohour configuration file



To do this, close the qt wallet and run it after deleting it.

### API zeroscan

Zeroscan has API that allows you to pull any information from the blockchain. For example, all transactions for a particular wallet

<https://ws.zerocscan.io/address/ZEFnGiHuwDStHnBA3cvAgPPFhhAKKqXQna/basic-txs>

The API description is similar to QTUM

<https://github.com/qtumproject/qtuminfo-api>

## **Installing the ZHCASH node on the server version of Ubuntu from version 18.04 and higher.**

I recommend that you start updating the node's blockchain in a tmux session

<https://losst.pro/shpargalka-po-tmux://losst.pro/shpargalka-po-tmux>

For connecting via ssh , I recommend MobaXterm

0. Go under the superuser

```
sudo su
```

Installing tmux

```
apt install tmux
```

<https://losst.ru/shpargalka-po-tmux>

in root , create a folder ZHCASH, where everything will spin

go to this folder

```
cd /root/ZHCASH/
```

1. Download the binary archive and unpack it:

```
wget https://zh.cash/download/ZHCash-Console-Linux.zip && unzip ZHCash-Console-Linux.zip
```

```
chmod -R 777 ./
```

Install the unzip archiver (by default , it is available in ubuntu), if it was not unzipped to the folder ZHCash-Console-Linux: apt install unzip

After unzipping, the info-read-me folder will appear , which contains one of the versions of the current guide.

Name	Size (KB)
..	
Chinese-information-from-the...	1
Deutsch-information-from-the...	1
English-information-from-the-...	1
PDF Guide_ZHCASH_eng.pdf	185
Hindi-information-from-the-de...	1
Korean-information-from-the-...	1
Russian-информация-от-раз...	1
Spanish-information-from-the...	1
Turkish-information-from-the-...	1
PDF Гайд_ZHCASH_rus.pdf	591

2. We grant rights to executable files for the user:

```
chmod u+x zerohourd && chmod u+x zerohour-cli
```

3. Create a folder of wallet data and a config file in it where we specify the parameters for running the daemon in the background and enabling staking mode in the wallet:

```
mkdir .zerohour && cd .zerohour && nano zerohour.conf
```

The nano text editor will start, and we'll set 2 parameters there:

```
daemon=1
```

```
staking=1
```

After that, save the configuration file with CTRL+O, exit the nano editor with CTRL+X

4. Create a session in tmux under the name zh

```
tmux new-session -s zh
```

5. Launch the tmux node in this session:

```
./zerohourd
```

A message will appear that ZHCASH has started. We leave the server for a couple of hours. Next, the wallet will start uploading blocks. You can view how much has already been downloaded by connecting to the server using a new one (not in the tmux session, where the node is being updated):

```
./zerohour-cli getblockchaininfo | grep blocks
```

Когда When the number of blocks equals the last block in the zcash explorerzcash, it means that the wallet has successfully synced and is ready for staking.

To disable staking (this is important for exchanges and if you need to have all zx on your balance) , it is not enough to set `staking = 0`, но also set `reservebalance=9999999999` in the config `zerohour.txt.txt`

### About becoming a node on the server

```
./zerohour-cli stop
```

### To run it, you need to run

```
./zerohourd
```

### Python Life Hacks для Питона

Install pip on ubuntu

```
sudo apt -y install python3-pip
```

### Disable Flask

```
from sys import platform
```

```
try:
```

```
    if platform == "linux" or platform == "linux2":
```

```
        list_files = subprocess.run(["lsof", "-i:5000"], stdout=subprocess.PIPE, text=True)
```

```
        list1 = list_files.stdout.split()
```

```
        pid_index = list1.index('python3') + 1
```

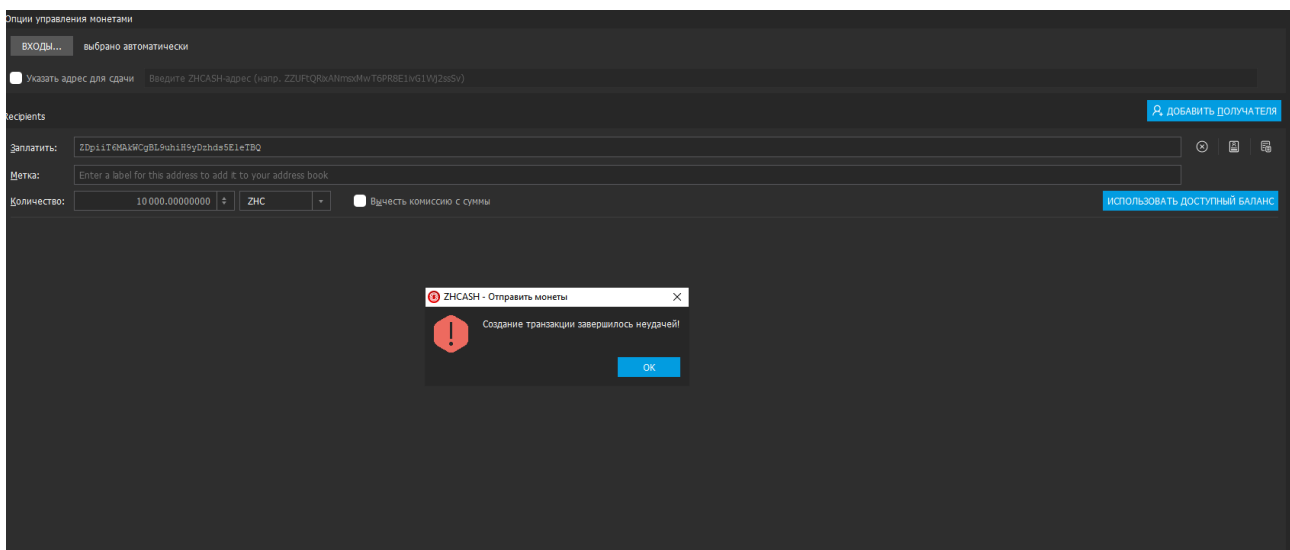
```
        pid = str(list1[pid_index])
```

```
        list_files2 = subprocess.run(["kill", "-9", pid], stdout=subprocess.PIPE, text=True)
```

```
except Exception as e:
```

```
    print(e)
```

**If you create too many inputs, you can get this error:**





To work around this error, select one of the inputs (top left)

## Conclusion

The author wrote a smart contract for the LIFT token. The latest version of the guide is also available there.

<https://github.com/dimaystinov/Token-LIFT-ZHCASH>

The author expresses his gratitude to the initiators of the creation of the LIFT token [https://t.me/lift\\_club](https://t.me/lift_club) with the address

f180d0a911d09853685764a9ad6d366398c50656

To Nikolai, Arjun and Denis.

To Chief Blockchain Engineer zx Roman Graviton, programmer Alex, and developer Mike Gurov for answering the dumb questions that formed the basis of this guide.

@QtumLeandro

(From the chat <https://t.me/qtumofficial> )

for the answer that you still need to send data to smart with the `sendtocontract` command `sendtocontract`.

Raul @kt2090 for the guide to installing a node on the server via ssh

Thank you to Mykola Eremenko for helping me write smarts, programs, and this guide.

Vitale for project the Donation system project

To the Connect project

<http://qtconnect.io>

To the Zero Gravity Foundation

<https://zerogravity.foundation>

Each project contributed to the improvement of this guide

Donations are accepted in shekels ZHC per purse:

ZEFnGiHuwDStHnBA3cvAgPPFhhAKKqXQna