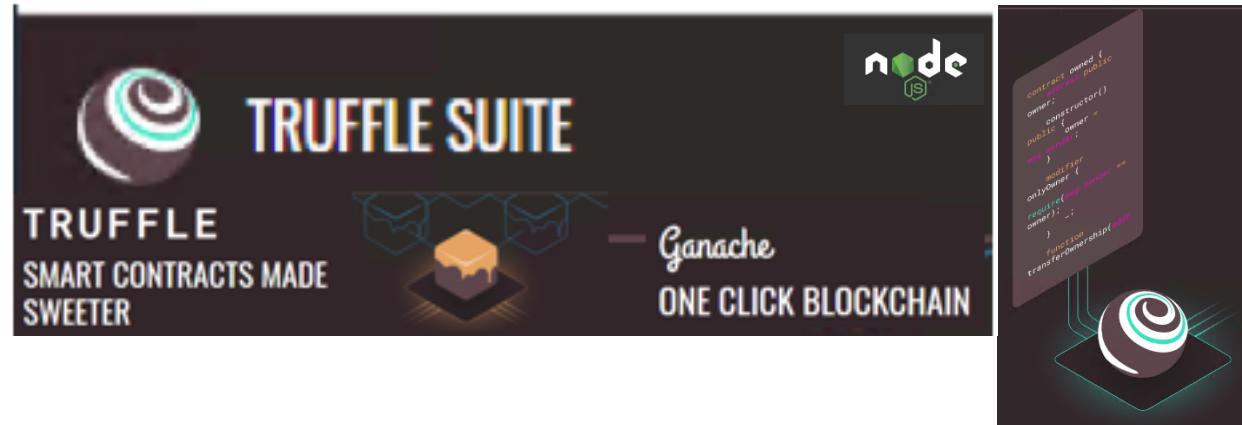




# Creating Blockchain Distributed Apps Tools and Examples



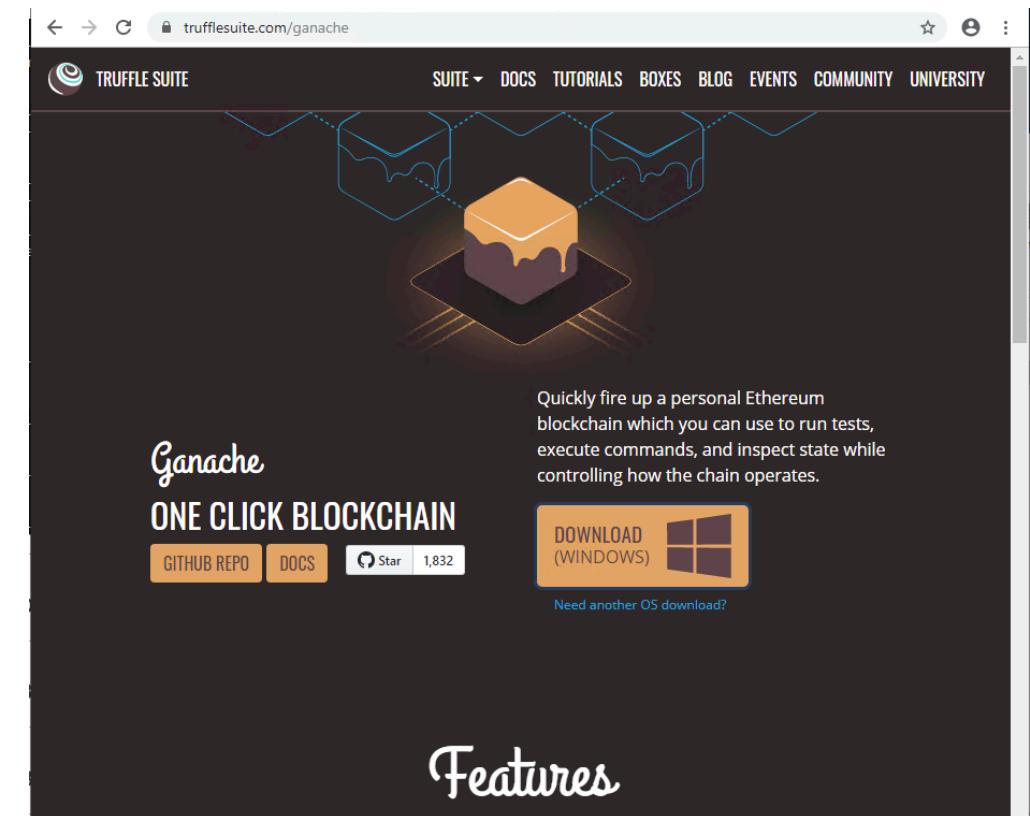
# First Download Tools

<https://nodejs.org/en/download/>



The screenshot shows the Node.js download page. At the top, there's a navigation bar with links for HOME, ABOUT, DOWNLOADS, DOCS, GET INVOLVED, SECURITY, NEWS, and FOUNDATION. Below the navigation bar, the Node.js logo is displayed. A main heading states "Node.js® is a JavaScript runtime built on Chrome's V8 JavaScript engine." Below this, there are two prominent download buttons: one for "12.13.0 LTS" (Recommended For Most Users) and another for "13.1.0 Current" (Latest Features). At the bottom of the page, there are links for "Other Downloads | Changelog | API Docs" and "Or have a look at the Long Term Support (LTS) schedule." A newsletter sign-up link for "Node.js Everywhere" is also present.

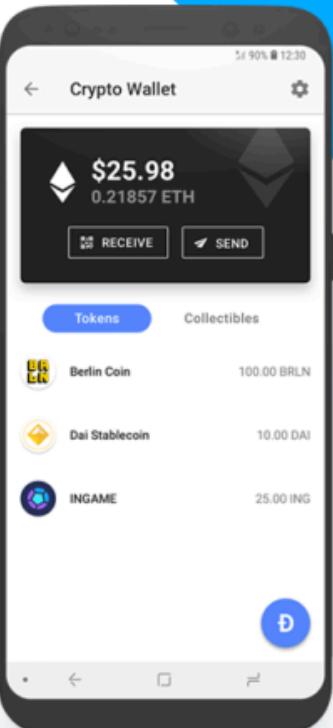
<https://www.trufflesuite.com/ganache>



The screenshot shows the Truffle Ganache download page. The header includes the Truffle Suite logo and navigation links for SUITE, DOCS, TUTORIALS, BOXES, BLOG, EVENTS, COMMUNITY, and UNIVERSITY. The main visual is a 3D rendering of three Ethereum blocks connected by dashed lines, with one block highlighted in orange. Below this, the text "Ganache ONE CLICK BLOCKCHAIN" is displayed, along with links for "GITHUB REPO", "DOCS", and a "Star" button with the number "1,832". A "DOWNLOAD (WINDOWS)" button featuring the Windows logo is prominently shown. At the bottom, there's a link for "Need another OS download?" and a large, stylized "Features" section.

# Optional Download Opera Phone App

<https://www.opera.com/crypto>



The screenshot shows the Opera mobile browser's built-in crypto wallet interface. At the top, it displays a balance of \$25.98 and 0.21857 ETH. Below this are buttons for 'RECEIVE' and 'SEND'. Underneath, there are sections for 'Tokens' and 'Collectibles'. The 'Tokens' section lists three items: 'Berlin Coin' (100.00 BRLN), 'Dai Stablecoin' (10.00 DAI), and 'INGAME' (25.00 ING). A blue arrow graphic points from the top left towards the text 'Built-in crypto wallet in Opera Touch on iOS and Opera for Android'.

FAQ    f    Twitter    YouTube    in    Instagram

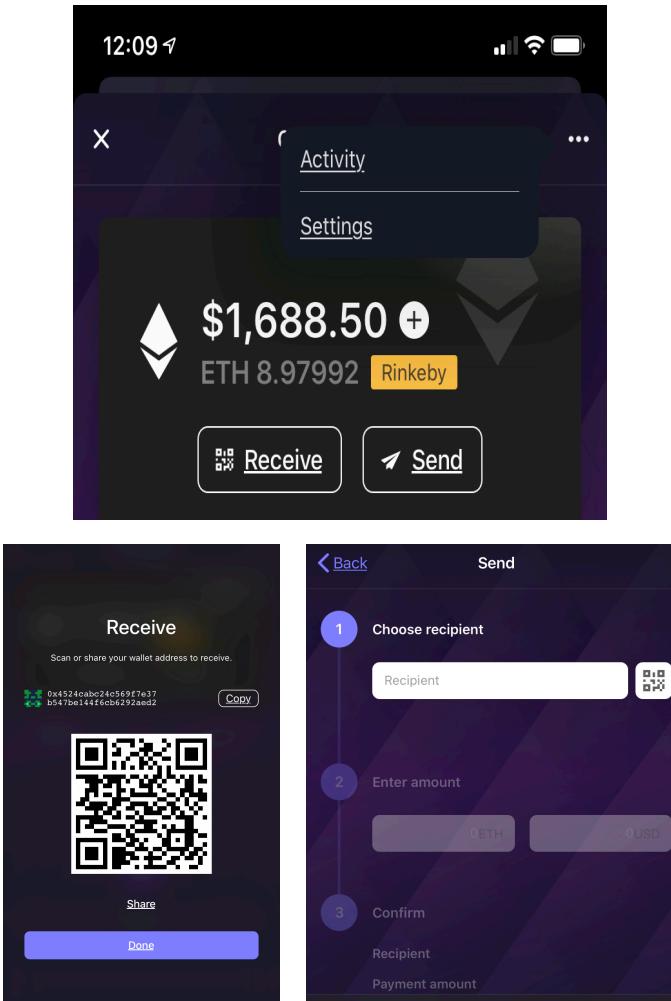
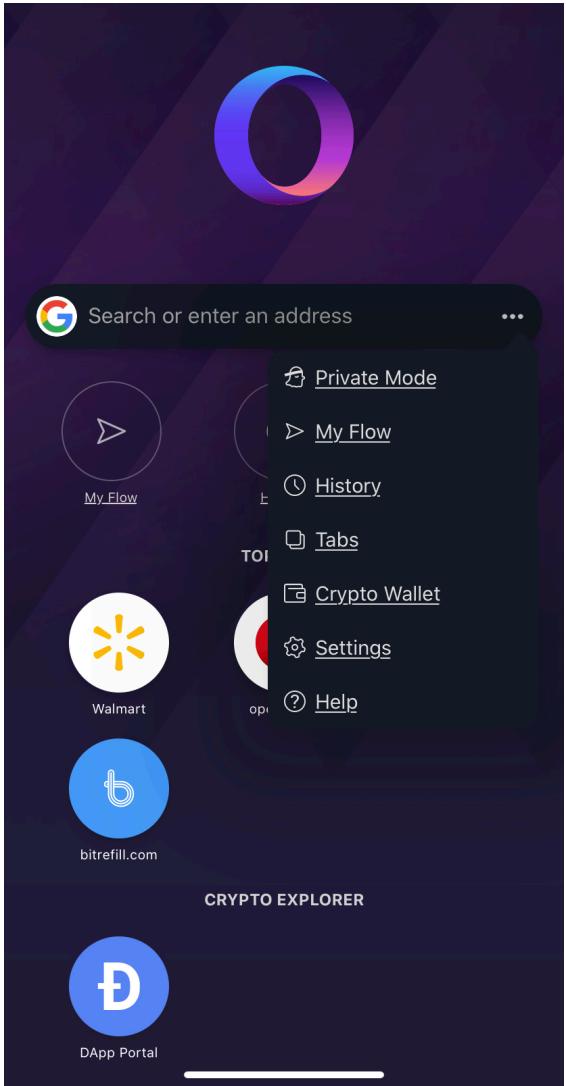
**Built-in crypto wallet in  
Opera Touch on iOS and  
Opera for Android**

The first major browser to integrate a crypto wallet, enabling seamless access to the emerging web of tomorrow (Web 3).

GET IT ON  
Google Play

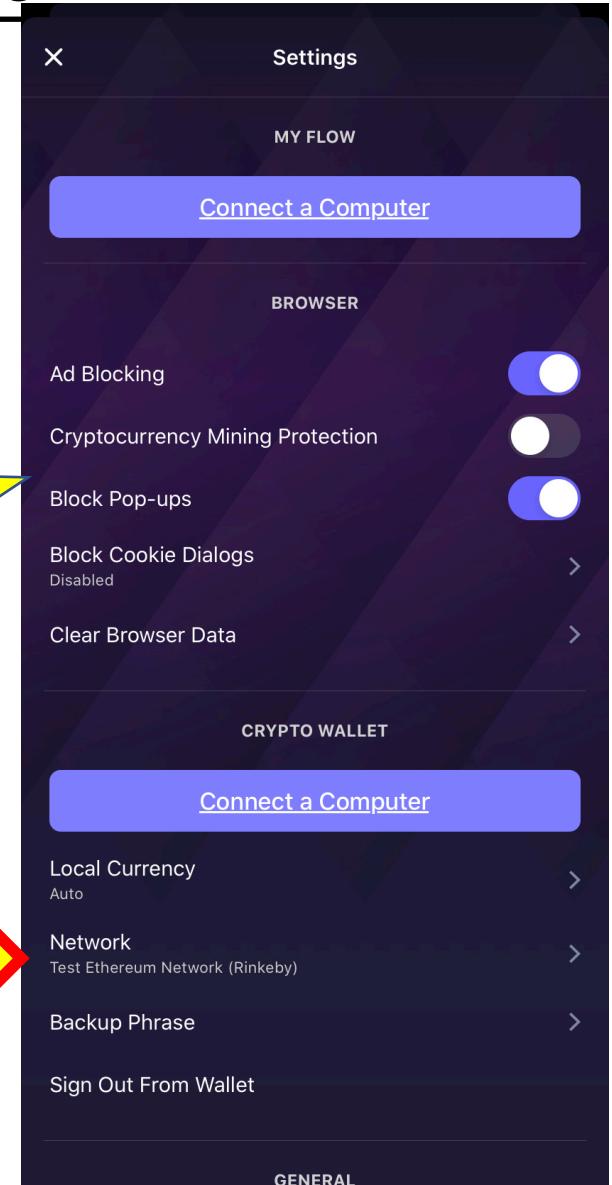
Download on the  
App Store

# Optional Opera iOS Phone Wallet

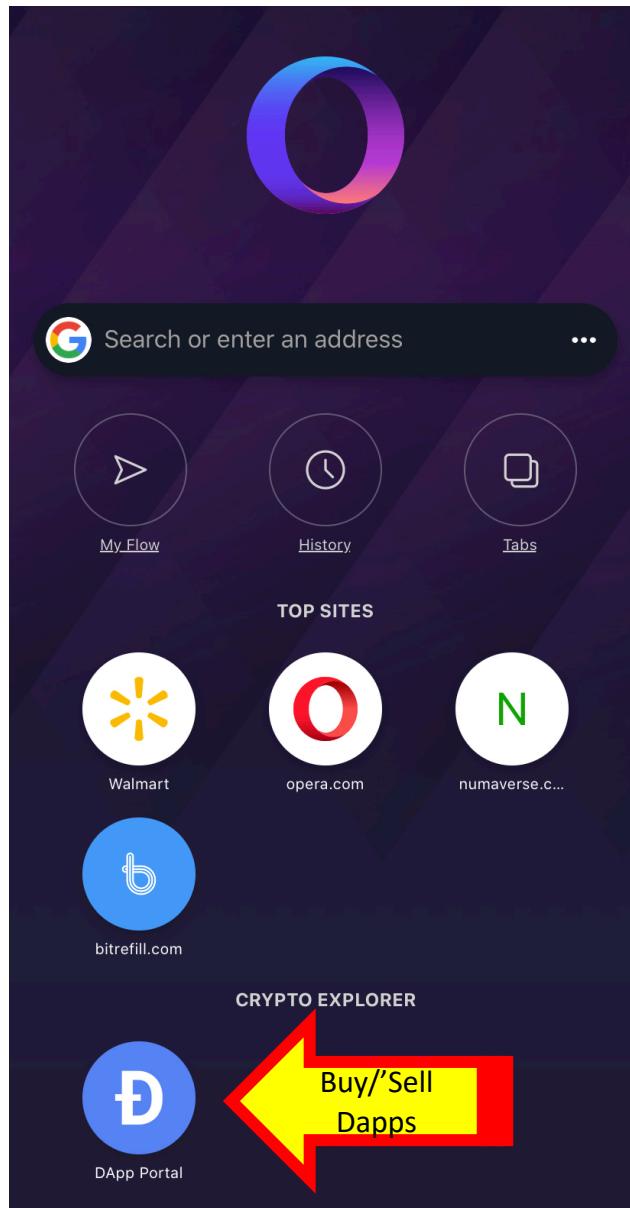


You can test  
With  
Ethereum  
Rinkeby test  
network

Test Network  
Rinkeby



# Optional Opera iOS Phone Wallet



This screenshot shows the "DApps Store" section of the Opera browser on iOS. It features a banner with the text "Discover Web 3 apps in your favourite browser" and an illustration of a smartphone displaying a dapp interface with a cat and a penguin. Below this, there are sections for "Buy with crypto" (listing "Opera Crypto Shop" and "Bitrefill"), "Earn crypto" (listing "Bounties Network" and "Cent"), and a "Crypto Explorer" section. A large red arrow on the left side of the screen points to the "Buy/Sell Dapps" button from the previous screenshot.

This screenshot shows the Bitrefill app interface on iOS. At the top, it says "12:22" and shows signal strength and battery level. The main header reads "Bitrefill" with a shopping cart icon. Below the header, the text "Live on Crypto" is displayed. It explains that users can purchase gift cards or mobile refills from over 1650 businesses in 170 countries using Bitcoin and other cryptocurrencies. A prominent green "Get Started" button with a right-pointing arrow is centered. Further down, it lists "Supported payment methods" including Bitcoin, Ethereum, Litecoin, and others. At the bottom, there's a section for "Popular Gift Cards in USA" with options for Google Play, App Store & iTunes, and Hotels.com.

This screenshot shows the Bitrefill app displaying various gift card options. It includes sections for "Rixty (Razer Gold) USA" and "PlayStation Network USA". Below these are sections for "Best Buy USA" (with a "BEST BUY" logo), "Uber Eats USA" (with a "Uber Eats" logo), "Airbnb USA" (with an Airbnb logo), "Target USA" (with a Target logo), "Roblox USA" (with a Roblox logo), and "eBay USA" (with an eBay logo). Each option has a small image and the text "USA" followed by the company name.

# Basic Blockchain Key Terms

Taken from: <https://objectcomputing.com/expertise/blockchain/glossary>

## **ETHERUM**

Ethereum is a decentralized Blockchain 2.0 chain. It was the first major smart contract platform and has widespread support from Fortune 500 companies through the Ethereum Enterprise Alliance (EEA). Ethereum currently uses a Proof-of-Work (PoW) consensus algorithm, but future changes to the protocol will update it to a more scalable algorithm, most likely based on Proof-of-Stake (PoS).

## **ETHER (ETH)**

The base cryptocurrency for the Ethereum blockchain network. Ether is used as the currency to pay transaction fees to miners. It is stored on an account basis, rather than in Unspent Transaction Outputs (UTXOs). Ether's base unit is wei, and one ETH is equivalent to  $10^{18}$  wei.

## **GAS**

In the blockchain industry, a measure of the computational difficulty required to process a smart contract function. More complex functions use more gas. Gas can be hardcoded values for each opcode (as is the case for Ethereum) or subjective values based on the preferences of the miner (as is the case for EOSIO).

## **GAS PRICE**

In the blockchain industry, the number of tokens that will be charged as a fee for each unit of gas consumed by a smart contract's function. Gas prices allow a network to dynamically respond to changes in bandwidth demand based on market forces.

# Basic Blockchain Key Terms

Taken from: <https://objectcomputing.com/expertise/blockchain/glossary>

## **BLOCKCHAIN**

A method of storing data in discrete sections (blocks) that are linked together. Blockchains specify criteria for what data can be stored in a block and reject invalid data.

## **TOKEN – COIN – BITCOIN**

In the blockchain industry, tokens are the generalized base unit of a cryptocurrency. A token is the lowest unit possible; it cannot be divided further.

## **CURRENCY**

A system of abstract representations of the ability to reconcile debts that is generally accepted or in use. Money is a currency. In the United States of America, the U.S. Dollar is the national currency.

## **CRYPTOCURRENCY**

Digitally distributed and traded [currencies](#) for which proof of ownership is established via [cryptographic](#) methods. For example, [Ether](#) cannot be transferred from an account without having control of the [private key](#) that is associated with that account.

# Basic Blockchain Key Terms

Taken from: <https://objectcomputing.com/expertise/blockchain/glossary>

## **SOLIDITY**

A smart contract programming language built for the Ethereum Virtual Machine. Syntactically it resembles C++ and Javascript and compiles to eWASM.

## **SMART CONTRACT**

Code that is executable within the environment of a virtual machine. Blockchains use smart contracts in the context of the chain's state to extend the functionality of the chain and provide trustless program execution. Smart contracts in blockchains are particularly useful because their outputs are deterministic, meaning anyone who processes a function in a smart contract will get the same output as anyone else performing the same function.

## **DAPP**

A decentralized application. DApps operate similarly to regular web applications; however, they retrieve their state and data from a blockchain network (or multiple blockchain networks). DApps do not require a central web server to function and can communicate to each other over the messaging protocol of the blockchain network(s) to which they're connected.

## **DECENTRALIZATION**

The movement of data, actions, and other interests away from a single actor in favor of distribution amongst all actors. In a decentralized system, no actor or group of actors can control the system without the consent of the rest of the actors.

# Basic Blockchain Key Terms

Taken from: <https://objectcomputing.com/expertise/blockchain/glossary>

## **TRANSACTION**

A singular input into a blockchain that affects some change in the blockchain's data.

Depending on the blockchain's implementation, transactions can transfer cryptocurrencies, create log events, or execute smart contract functions.

## **TRANSACTION FEE**

An amount of cryptocurrency, typically a blockchain's native cryptocurrency, that is charged by a miner to manipulate data on the chain. Transaction fees are generally variable and heavily dependent upon market conditions.

Example

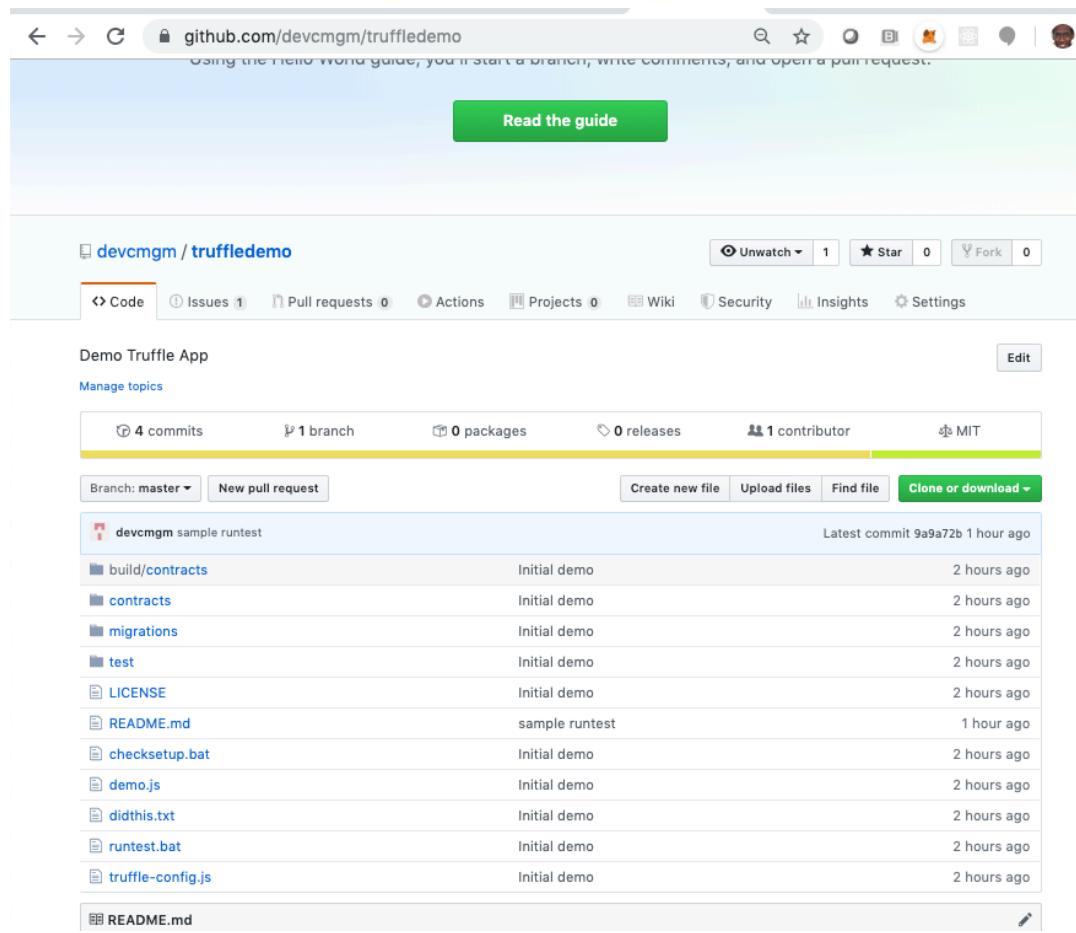
If many transactions are being requested, and a limited number can be included in a block, a miner chooses to include whatever transactions have the highest profit margins, i.e., the transactions with the highest fees. Higher transaction fees can be set to incentivize miners to prioritize including a specific transaction in a block. In the reverse, if fewer transactions are being requested than will fit in a block, miners include all transactions that meet a minimum profitability requirement, and lower transaction fees can be set.

## **WALLET**

A file or software that contains the private keys for interacting with a Private Key Infrastructure (PKI). Typical wallet software has functionality for signing messages and transactions for the corresponding network.

# Simple Demo Code using MetaCoin

<https://github.com/devcmgm/truffledemo>



The screenshot shows a GitHub repository page for 'truffledemo' by devcmgm. The repository has 4 commits, 1 branch, 0 packages, 0 releases, and 1 contributor. The README.md file is open, displaying the following content:

```
truffledemo

Demo Truffle App

This demo give you a simple command line app using the demo unbox code from MetaCoin.

NOTE: You should download and install Ganache https://www.trufflesuite.com/ganache and NODEJS https://nodejs.org/en/download/

Start up Ganache form Windows Start menu and ensure setup for http://127.0.0.1:7545

First run the checksetup.bat script to see what pre-reqs are needed.

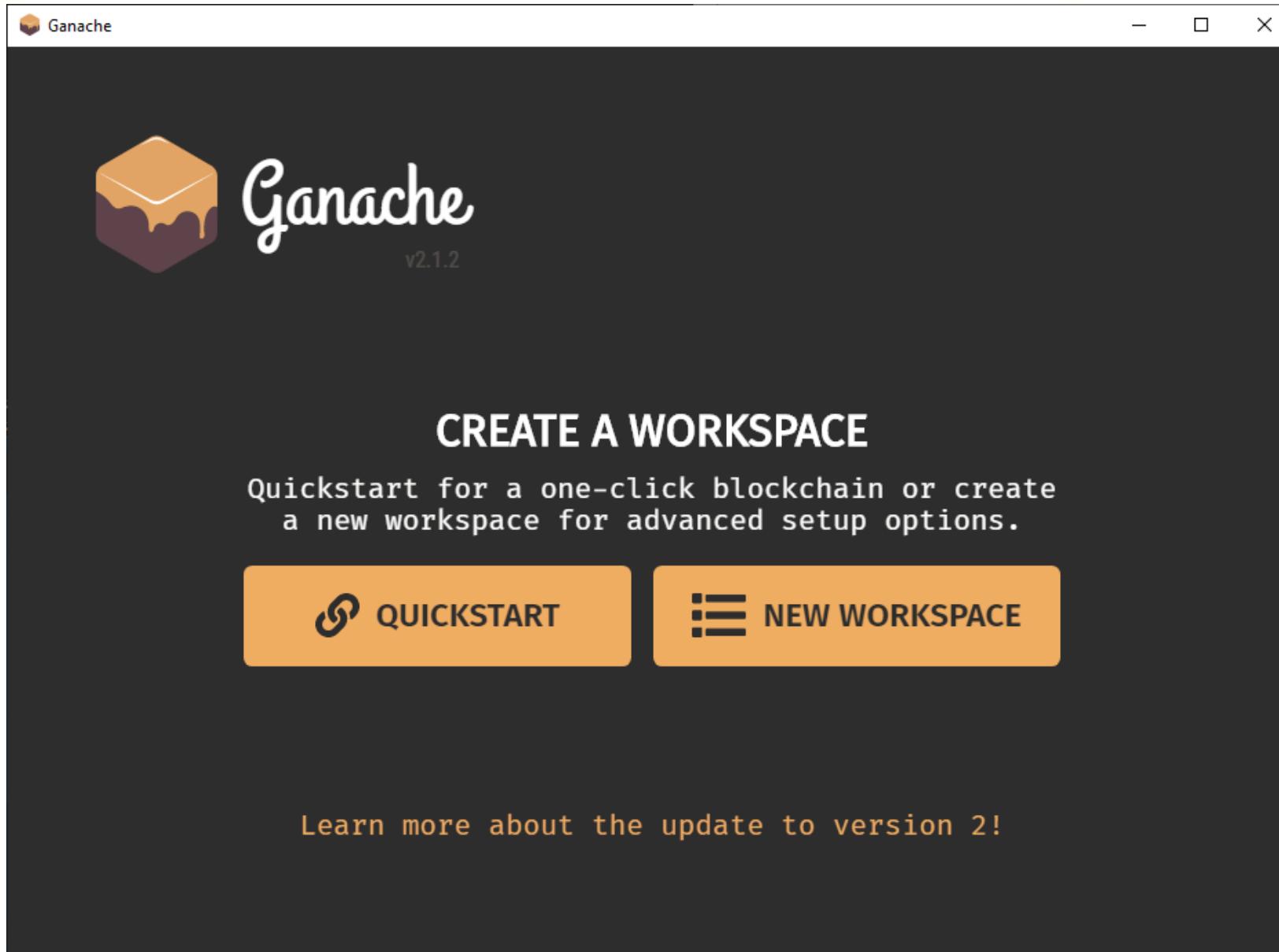
Then run the runtest.bat to see example.

+++++ Sample Run +++++

Your environment has been set up for using Node.js 12.13.0 (x64) and npm.

C:\Users\demo>cd C:\Users\demo\Downloads\truffledemo-master\truffledemo-master
C:\Users\demo\Downloads\truffledemo-master\truffledemo-master>checksetup
C:\Users\demo\Downloads\truffledemo-master\truffledemo-master>node --version
2>NUL || echo "FIRST: You must download and install Node JS !"
&& goto enderr v12.13.0
C:\Users\demo\Downloads\truffledemo-master\truffledemo-master>REM npm -version
2>NUL || echo "SECOND download install Node NPM Package manager !"
&& goto enderr
C:\Users\demo\Downloads\truffledemo-master\truffledemo-master>node --version
2>NUL && echo "Node JS is Installed" v12.13.0 "Node JS is Installed"
```

<https://www.trufflesuite.com/ganache>



Ganache

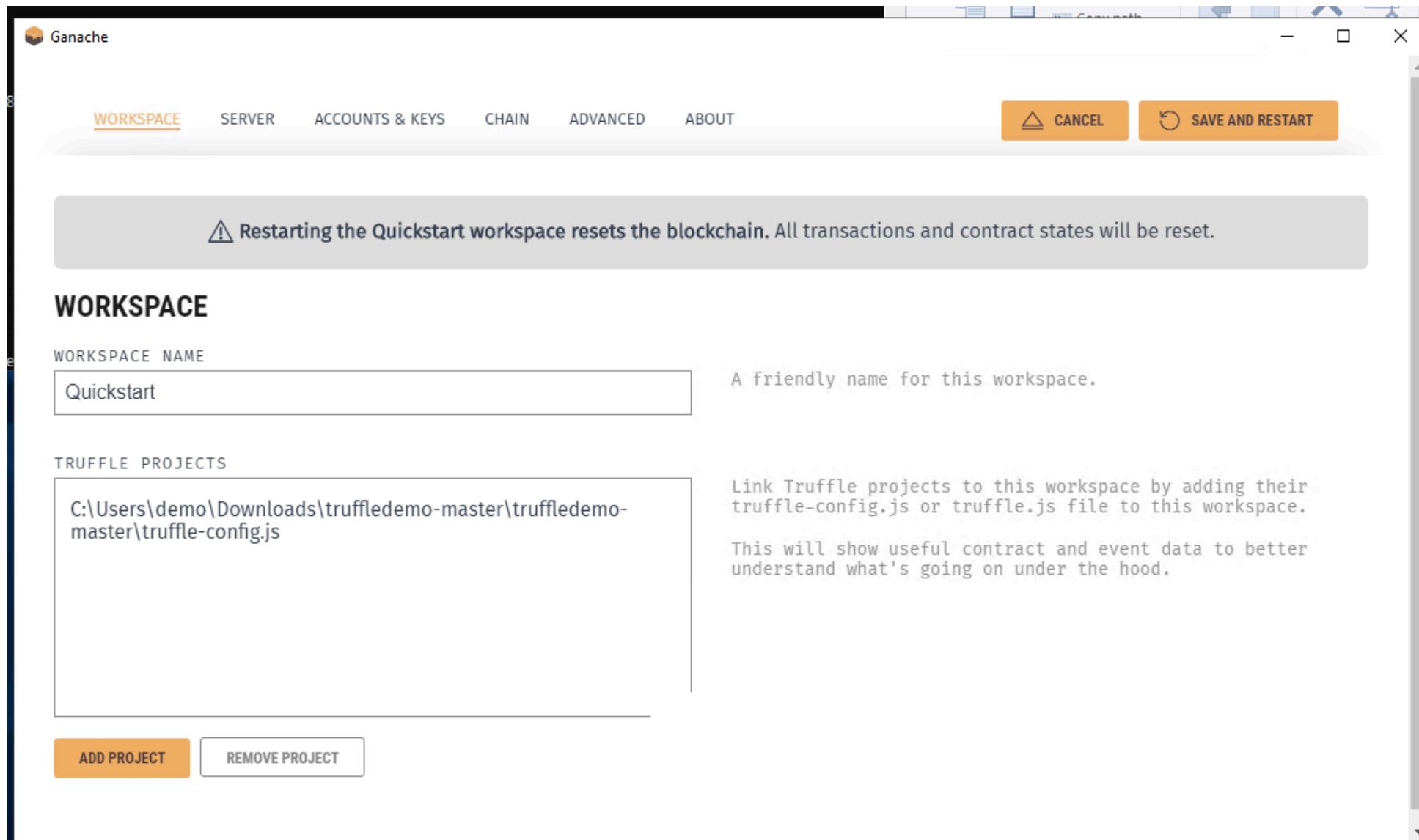
8 ACCOUNTS BLOCKS TRANSACTIONS CONTRACTS EVENTS LOGS SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK 0 GAS PRICE 20000000000 GAS LIMIT 6721975 HARDFORK PETERSBURG NETWORK ID 5777 RPC SERVER HTTP://127.0.0.1:7545 MINING STATUS AUTOMINING WORKSPACE QUICKSTART SAVE SWITCH

MNEMONIC ? liberty stick obscure husband upon almost cheap glue brown midnight domain romance HD PATH m/44'/60'/0'/0/account\_index

ADDRESS	BALANCE	TX COUNT	INDEX	
0x5757121ff2cfE720703D20240A9eF5000c7A7A63	100.00 ETH	0	0	🔑
0xb01573b79bED2D5Dd58Ff5c3CFe5C2F6cB12bFf4	100.00 ETH	0	1	🔑
0xb4EF99a222323f6829b71FcFD28C8076e375986	100.00 ETH	0	2	🔑
0xa6398834A2f8309927c27ECe7DD6d51f14E313Db	100.00 ETH	0	3	🔑
0xF7cdd868122375AD88533bD76a68f87D914789A2	100.00 ETH	0	4	🔑

# Load Contracts from config file



Ganache

— □ ×

ACCOUNTS BLOCKS TRANSACTIONS CONTRACTS EVENTS LOGS SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK 0 GAS PRICE 200000000000 GAS LIMIT 6721975 HARDFORK PETERSBURG NETWORK ID 5777 RPC SERVER HTTP://127.0.0.1:7545 MINING STATUS AUTOMINING WORKSPACE QUICKSTART SAVE SWITCH

**truffledemo-master** C:\Users\demo\Downloads\truffledemo-master\truffledemo-master

NAME	ADDRESS	TX COUNT
ConvertLib	Not Deployed	0
MetaCoin	Not Deployed	0
Migrations	Not Deployed	0

# After truffle Migrate

The screenshot shows the Ganache interface with the following details:

NAME	ADDRESS	TX COUNT	DEPLOYED
ConvertLib	0xA5cf418eeeea178ac53f51D2786049B15cD0B7eD9	0	DEPLOYED
MetaCoin	0xc6d082CeD775f752cC3ff1fe4207278345dc9dd4	2	DEPLOYED
Migrations	0xB5C042622C776f1d2c962eF1eC4702527dc2C288	1	DEPLOYED

# Perform a Send Coin Transaction

```
demo:pwd
/Users/zacharylewis/Desktop/truffledemo/MetaCoin
demo:truffle console
truffle(development)> let instance = await MetaCoin.deployed()
undefined
truffle(development)> let accounts = await web3.eth.getAccounts()
undefined
truffle(development)> instance.sendCoin(accounts[1], 10, {from: accounts[0]})
{ tx:
  '0x959177ff9a746e357595a961a739e9dd77cac20395953c169b017a912a98fed8',
  receipt:
    { transactionHash:
        '0x959177ff9a746e357595a961a739e9dd77cac20395953c169b017a912a98fed8',
      transactionIndex: 0,
      blockHash:
        '0x9534cd71ea2483ba3b567c46b62c089219f4ce35b140bb6a5b318d4dafe67805',
      blockNumber: 12,
      from: '0xfa946c43d9b453bb06b3bf427caa7e05c17d3a92',
      to: '0xc6d082ced775f752cc3ff1fe4207278345dc9dd4',
      gasUsed: 36008,
      cumulativeGasUsed: 36008,
      contractAddress: null,
```

Ganache

ACCOUNTS BLOCKS TRANSACTIONS CONTRACTS EVENTS LOGS SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK 12 GAS PRICE 20000000000 GAS LIMIT 6721975 HARDFORK PETERSBURG NETWORK ID 5777 RPC SERVER HTTP://0.0.0:7545 MINING STATUS AUTOMINING WORKSPACE QUICKSTART SAVE SWITCH

TX HASH **0x959177ff9a746e357595a961a739e9dd77cac20395953c169b017a912a98fed8** CONTRACT CALL

FROM ADDRESS 0xfA946c43d9B453Bb06B3bF427cAA7e05C17D3a92 TO CONTRACT ADDRESS MetaCoin GAS USED 36008 VALUE 0

TX HASH **0xc778cac9657c0339a33a702df0eb10ed291e0eec99d1dad11559ef186accc9f2** CONTRACT CALL

FROM ADDRESS 0xfA946c43d9B453Bb06B3bF427cAA7e05C17D3a92 TO CONTRACT ADDRESS MetaCoin GAS USED 51008 VALUE 0

TX HASH **0x33099b48b492aa1a7dd1ea8a68766a06d964dd19eecca78511f088bdd967d5c6** CONTRACT CALL

FROM ADDRESS 0xfA946c43d9B453Bb06B3bF427cAA7e05C17D3a92 TO CONTRACT ADDRESS Migrations GAS USED 27023 VALUE 0

TX HASH **0x68d9daad5fbd81e699974632e76391ee43ef84ae297d8f4bcb95d9e51c26760e** CONTRACT CREATION

FROM ADDRESS 0xfA946c43d9B453Bb06B3bF427cAA7e05C17D3a92 CREATED CONTRACT ADDRESS 0xc6d082CeD775f752cc3ff1fe4207278345dc9dd4 GAS USED 338349 VALUE 0

TX HASH CONTRACT CREATION



ACCOUNTS



BLOCKS



TRANSACTIONS



CONTRACTS



EVENTS



LOGS

SEARCH FOR BLOCK NUMBERS OR TX HASHES

CURRENT BLOCK  
6GAS PRICE  
20000000000GAS LIMIT  
6721975HARDFORK  
PETERSBURGNETWORK ID  
5777RPC SERVER  
HTTP://127.0.0.1:7545MINING STATUS  
AUTOMININGWORKSPACE  
QUICKSTART

SAVE

SWITCH



## MNEMONIC



liberty stick obscure husband upon almost cheap glue brown midnight domain romance

## HD PATH

m/44'/60'/0'/0/account\_index

## ADDRESS

0x5757121ff2cfE720703D20240A9eF5000c7A7A63

## BALANCE

99.98 ETH

## TX COUNT

6

## INDEX

0



## ADDRESS

0xb01573b79bED2D5Dd58Ff5c3CFe5C2F6cB12bFf4

## BALANCE

100.00 ETH

## TX COUNT

0

## INDEX

1



## ADDRESS

0x0b4EF99a222323f6829b71FcFD28C8076e375986

## BALANCE

100.00 ETH

## TX COUNT

0

## INDEX

2



## ADDRESS

0xa6398834A2f8309927c27ECe7DD6d51f14E313Db

## BALANCE

100.00 ETH

## TX COUNT

0

## INDEX

3



## ADDRESS

0xF7cdd868122375AD88533bD76a68f87D914789A2

## BALANCE

100.00 ETH

## TX COUNT

0

## INDEX

4







# Sample Dapp File Layout

 devcmgm	Update README.md
 build/contracts	Initial demo
 contracts	Initial demo
 migrations	Initial demo
 test	Initial demo
 LICENSE	Initial demo
 README.md	Update README.md
 checksetup.bat	Initial demo
 demo.js	Initial demo
 didthis.txt	Initial demo
 runtest.bat	Initial demo
 truffle-config.js	Initial demo

# Solidity Code

---

```
1 pragma solidity >=0.4.25 <0.6.0;
2
3 import "./ConvertLib.sol";
4
5 // This is just a simple example of a coin-like contract.
6 // It is not standards compatible and cannot be expected to talk to other
7 // coin/token contracts. If you want to create a standards-compliant
8 // token, see: https://github.com/ConsenSys/Tokens. Cheers!
9
10 contract MetaCoin {
11     mapping (address => uint) balances;
12
13     event Transfer(address indexed _from, address indexed _to, uint256 _value);
14
15     constructor() public {
16         balances[tx.origin] = 10000;
17     }
18
19     function sendCoin(address receiver, uint amount) public returns(bool sufficient) {
20         if (balances[msg.sender] < amount) return false;
21         balances[msg.sender] -= amount;
22         balances[receiver] += amount;
23         emit Transfer(msg.sender, receiver, amount);
24         return true;
25     }
26
27     function getBalanceInEth(address addr) public view returns(uint){
28         return ConvertLib.convert(getBalance(addr),2);
29     }
30
31     function getBalance(address addr) public view returns(uint) {
32         return balances[addr];
33     }
34 }
```

# Command Line NODE Javascript DApp

---

```
1
2  'use strict';
3
4  global.artifacts = artifacts;
5  global.web3 = web3;
6
7  async function main(){
8      //web3 = new Web3(new Web3.providers.HttpProvider("http://localhost:7545"));
9      const newtworkType = await web3.eth.net.getNetworkType();
10     const networkId = await web3.eth.net.getId();
11     console.log("network type:"+newtworkType);
12     console.log("network id:"+networkId);
13     const MetaCoin = artifacts.require("MetaCoin");
14     let instance = await MetaCoin.deployed();
15     let accounts = await web3.eth.getAccounts();
16     let balance = await instance.getBalance(accounts[0]);
17     console.log(balance.toNumber());
18     let ether = await instance.getBalanceInEth(accounts[0]);
19     console.log(ether.toNumber());
20     instance.sendCoin(accounts[1], 500);
21     let received = await instance.getBalance(accounts[1]);
22     console.log(received.toNumber());
23     let newBalance = await instance.getBalance(accounts[0]);
24     console.log(newBalance.toNumber());
25 }
26
27 // For truffle exec
28 module.exports = function(callback) {
29     main().then(() => callback()).catch(err => callback(err))
30 };
```

<https://www.trufflesuite.com/docs/truffle/getting-started/interacting-with-your-contracts>

**truffle unbox metacoin**

**Update: truffle-config**

**Truffle compile**

**Truffle migrate**

**Truffle console**

<https://github.com/trufflesuite/truffle/tree/master/packages/contract>

.

# Other Resources

<https://ethereum.org/>

<https://www.trufflesuite.com/tutorials>

<https://www.dappuniversity.com/articles/the-ultimate-ethereum-dapp-tutorial>

[https://www.youtube.com/results?search\\_query=blockchain+tutorial](https://www.youtube.com/results?search_query=blockchain+tutorial)

<https://www.youtube.com/watch?v=2CRI0s6UPfA>