Lecture Notes

Smart Contract Development in Ethereum

Ganache

In this session, you learnt about the following:

- How to deploy smart contracts using Ganache.

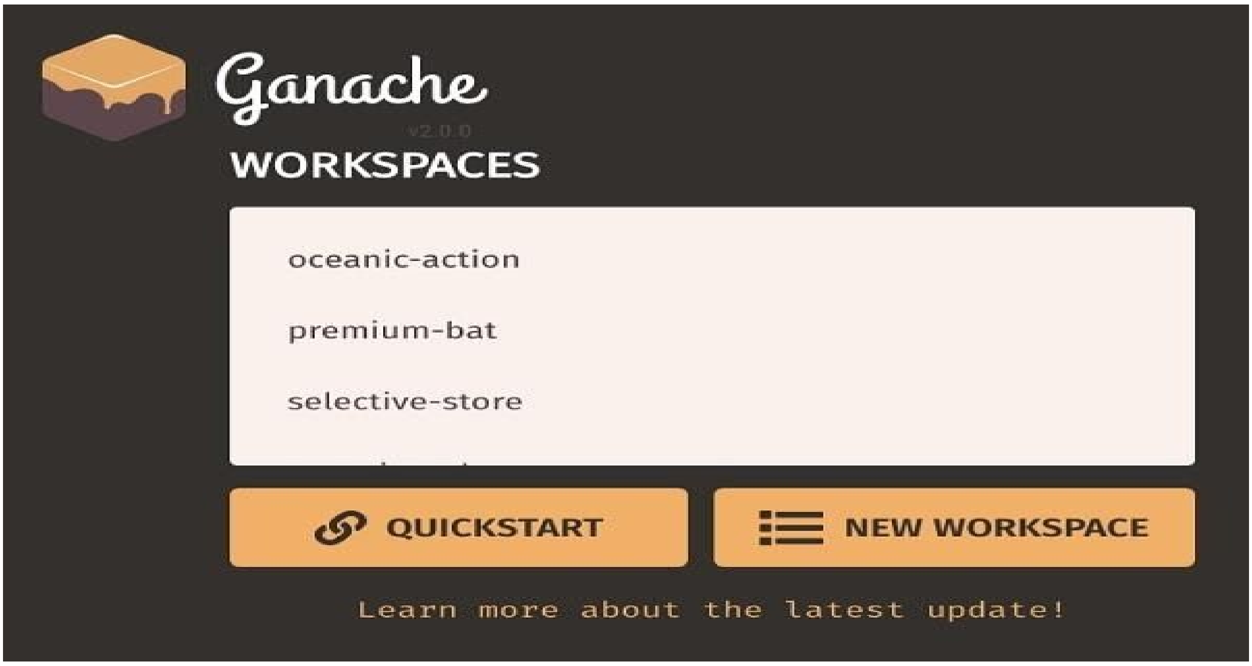
Introduction to Ganache

The deployment process using Truffle is a little complicated. Ganache provides a simplified way to deploy a smart contract on a network. The best thing about Ganache is that it comes with a graphical user interface, or GUI. This makes Ganache much more convenient to use as compared with Truffle's CLI options.

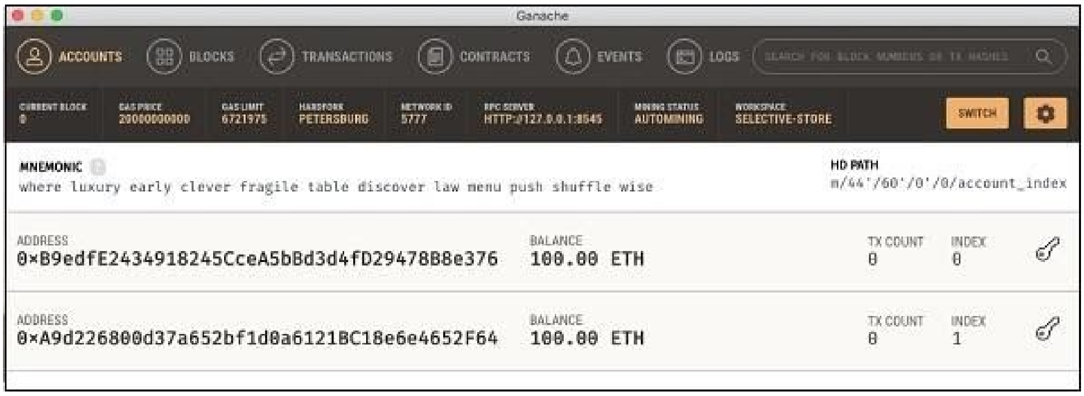
Ganache is basically a test blockchain. It is like Remix IDE where there are five sample accounts and you can easily deploy the smart contract on the blockchain. Ganache comes with GUI, which makes the debugging easier.

Ganache Desktop

When Ganache starts, the screen will appear as shown in the following image.



Click ‘QUICKSTART’ to start Ganache. You will see the Ganache console as shown in the following image.



CURRENT BLOCK 0
GAS PRICE 20000000000
GAS LIMIT 6721975
HARDFORK PETERSBURG
NETWORK ID 5777
RPC SERVER HTTP://127.0.0.1:8545
MINING STATUS AUTOMINING
WORKSPACE SELECTIVE-STORE

MNEMONIC
where luxury early clever fragile table discover law menu push shuffle wise

HD PATH
m/44'/60'/0'/0/account_index

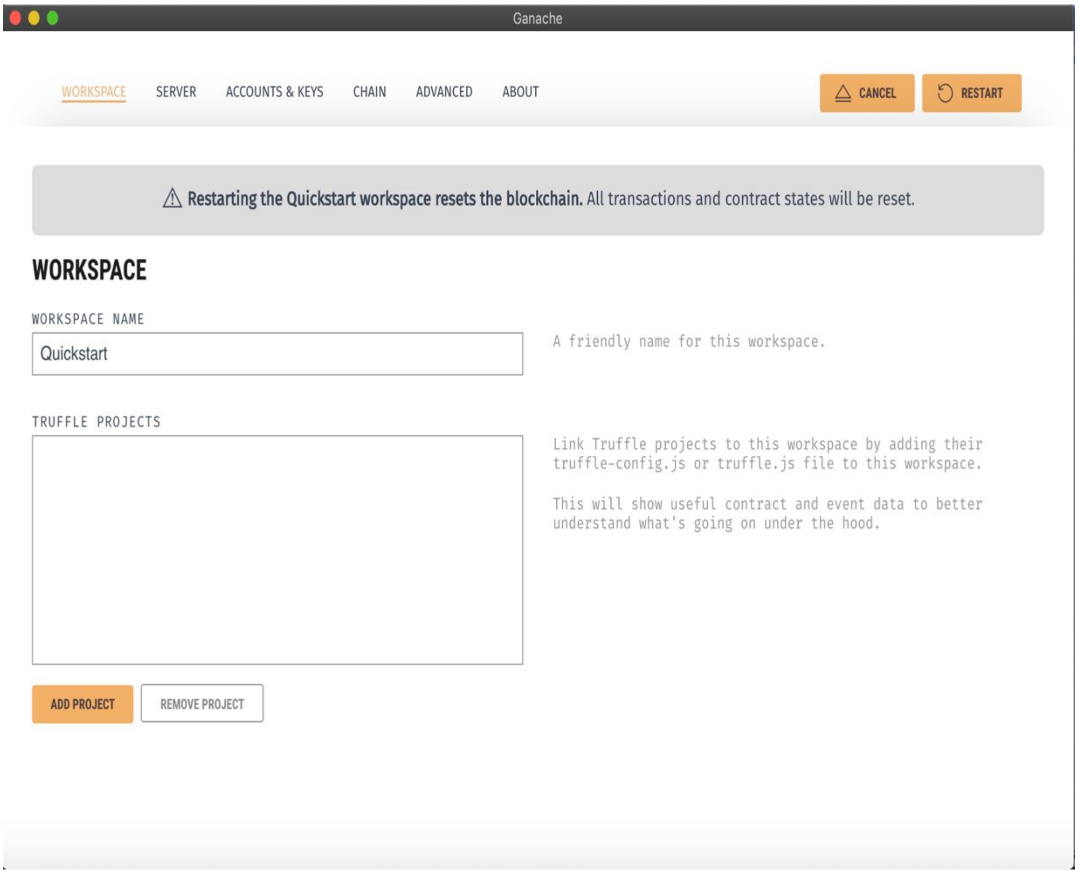
ADDRESS 0×B9edfE2434918245CceA5bBd3d4fD29478B8e376
BALANCE 100.00 ETH
TX COUNT 0
INDEX 0

ADDRESS 0×A9d226800d37a652bf1d0a6121BC18e6e4652F64
BALANCE 100.00 ETH
TX COUNT 0
INDEX 1

The console in the screenshot shows user accounts with a balance of 100 ETH each. It also shows a transaction count of zero for each account. As the user has not performed any transactions so far, this count is obviously zero.

Click on the settings icon at the top right hand side of the screen to configure Ganache:

1. **Workspace tab**
   Here, you can name the project or link truffle projects. (To link truffle project, you need to provide a **truffle-config.js** file).



⚠ **Restarting the Quickstart workspace resets the blockchain.** All transactions and contract states will be reset.

WORKSPACE

WORKSPACE NAME

Quickstart

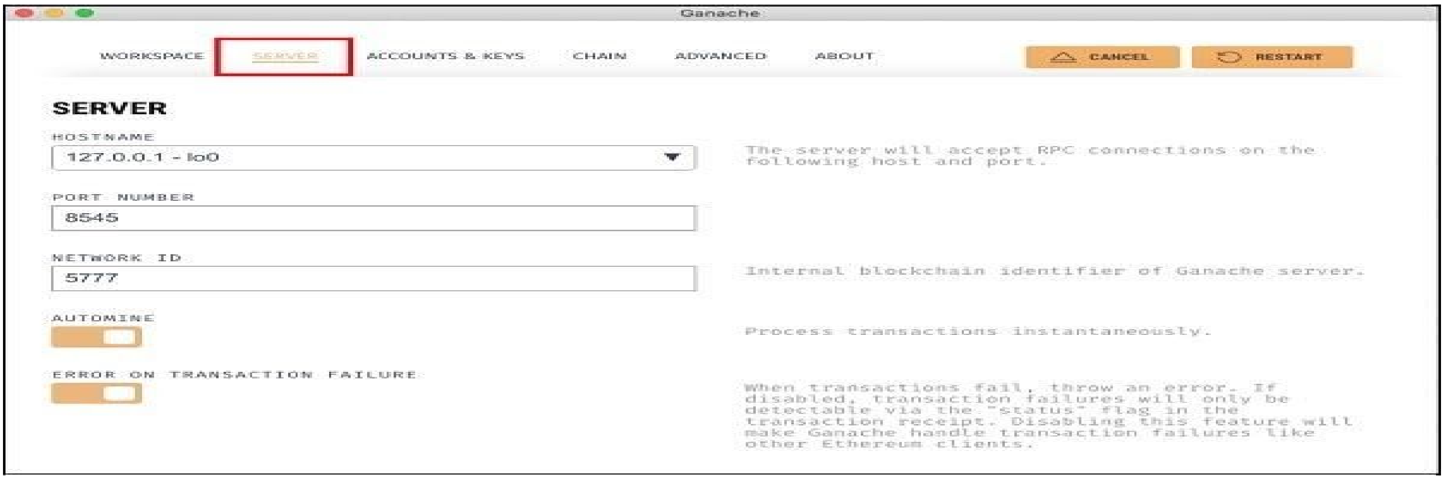
A friendly name for this workspace.

TRUFFLE PROJECTS

Link Truffle projects to this workspace by adding their truffle-config.js or truffle.js file to this workspace.

This will show useful contract and event data to better understand what's going on under the hood.

2. **Server tab**



SERVER

HOSTNAME

127.0.0.1 - lo0

The server will accept RPC connections on the following host and port.

PORT NUMBER

8545

NETWORK ID

5777

Internal blockchain identifier of Ganache server.

AUTOMINE

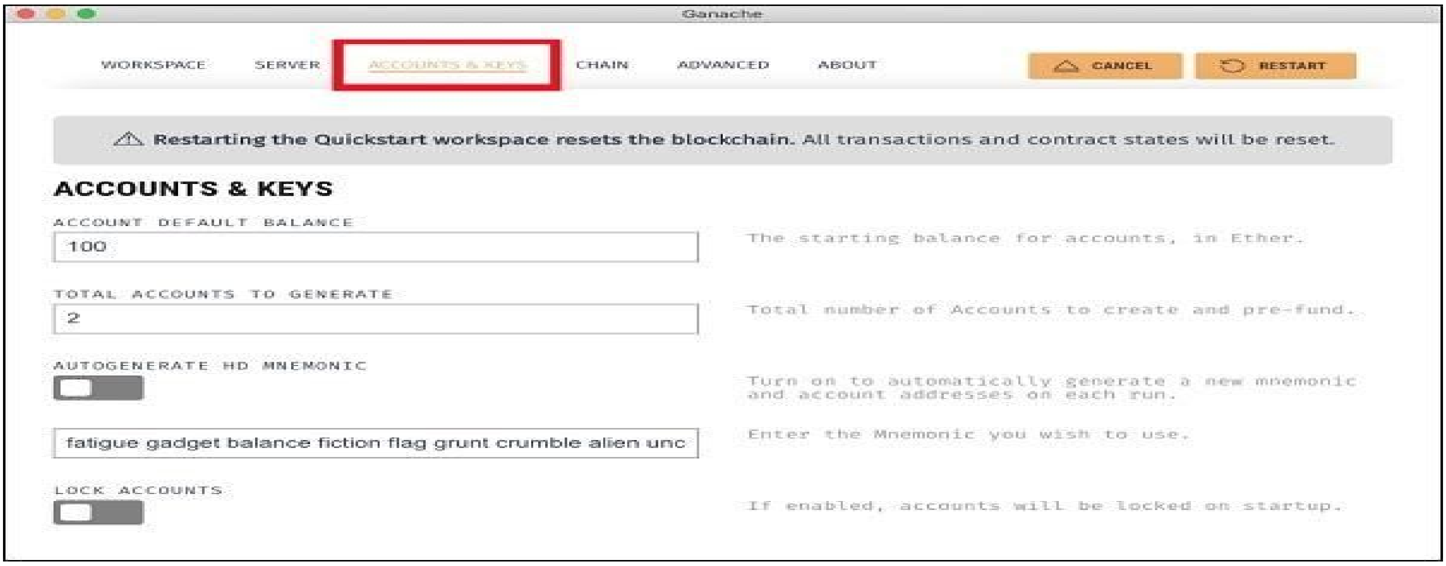
Process transactions instantaneously.

ERROR ON TRANSACTION FAILURE

When transactions fail, throw an error. If disabled, transaction failures will only be detectable via the "status" flag in the transaction receipt. Disabling this feature will make Ganache handle transaction failures like other Ethereum clients.

Here, you will be able to set the values of the server address and the port number for your Ganache server. For the time being, leave these to their default values. The network id is an internal blockchain identifier of Ganache server. Leave this to its default value.
The **Automine** button is in the ON state, indicating that the transactions would be processed instantly. If you switched this off, it would ask you to enter the time in seconds after which the blocks would be mined.

3. **Account and keys tab**



⚠ **Restarting the Quickstart workspace resets the blockchain.** All transactions and contract states will be reset.

ACCOUNTS & KEYS

ACCOUNT DEFAULT BALANCE: 100 — The starting balance for accounts, in Ether.

TOTAL ACCOUNTS TO GENERATE: 2 — Total number of Accounts to create and pre-fund.

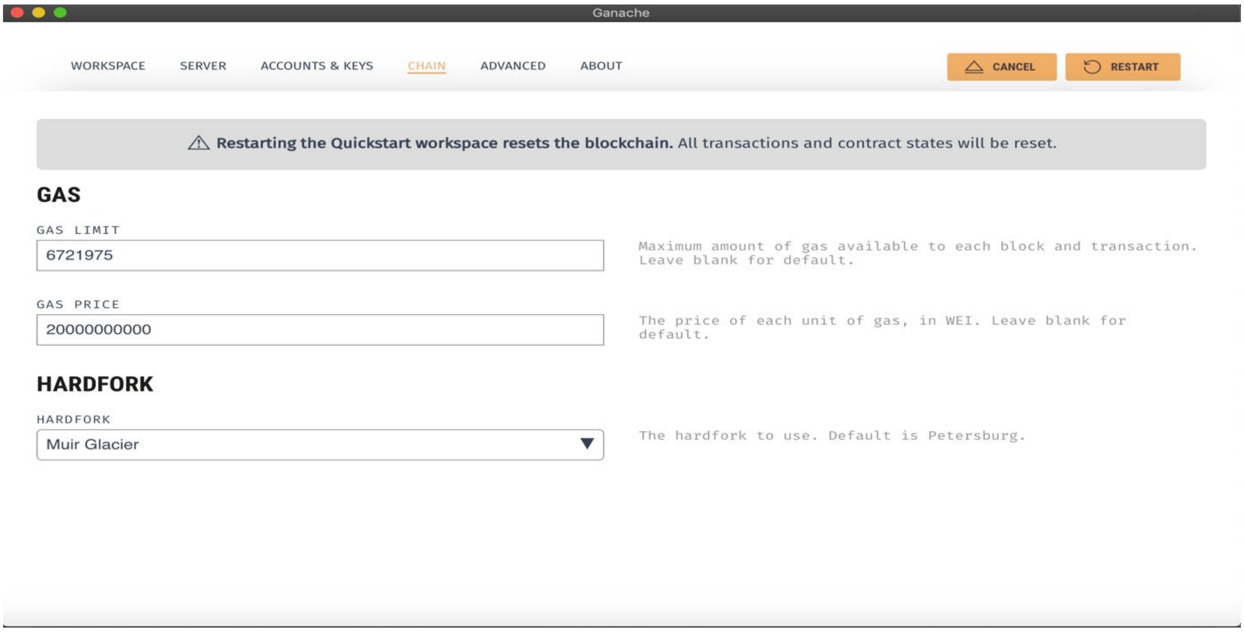
AUTOGENERATE HD MNEMONIC — Turn on to automatically generate a new mnemonic and account addresses on each run.

fatigue gadget balance fiction flag grunt crumble alien unc — Enter the Mnemonic you wish to use.

LOCK ACCOUNTS — If enabled, accounts will be locked on startup.

Here, you would be able to **set** the default balance for each account. The default value is 100. This now explains why you saw 100 ETH displayed for each account in the desktop screenshot. You can also set the number of accounts on this screen; the default value is 10. The value displayed in this screenshot is 2 and that is why the desktop showed only two accounts.

4. **Chain tab**



⚠ **Restarting the Quickstart workspace resets the blockchain.** All transactions and contract states will be reset.

GAS

GAS LIMIT: 6721975 — Maximum amount of gas available to each block and transaction. Leave blank for default.

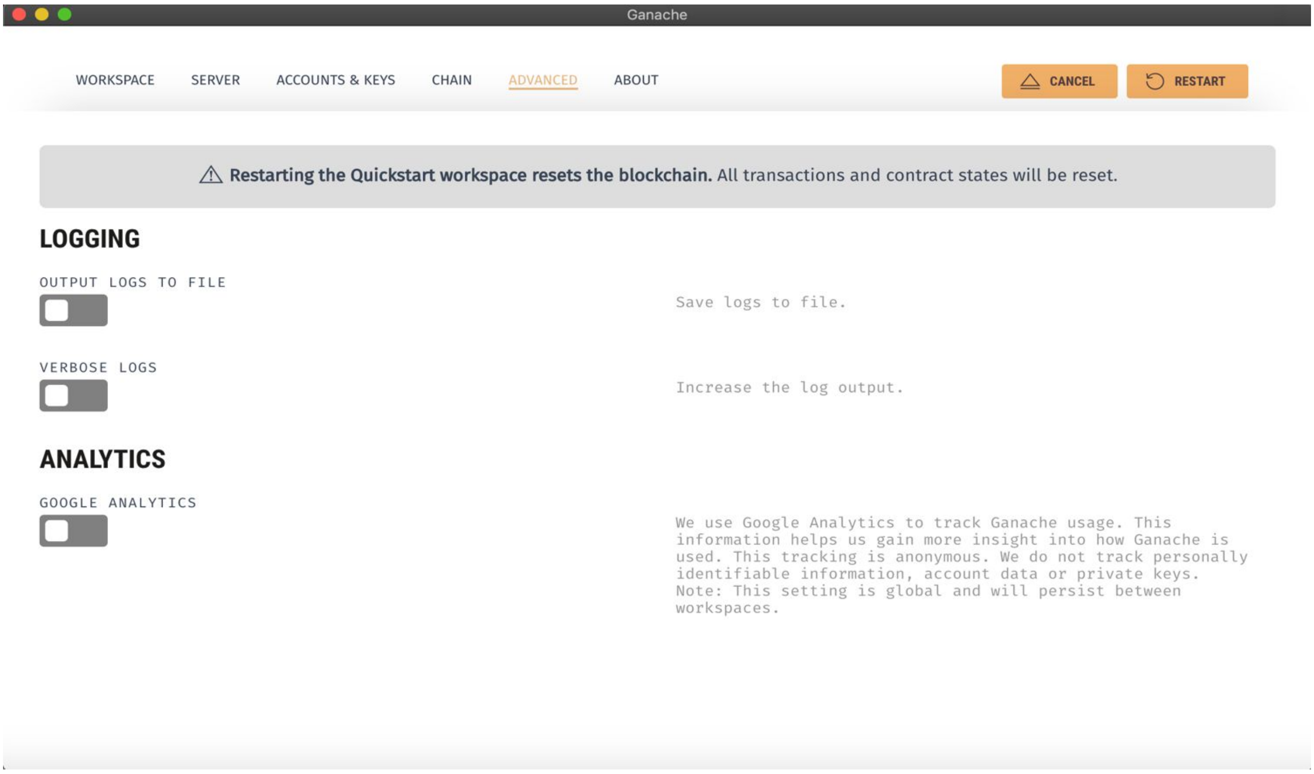
GAS PRICE: 20000000000 — The price of each unit of gas, in WEI. Leave blank for default.

HARDFORK

HARDFORK: Muir Glacier ▼ — The hardfork to use. Default is Petersburg.

In the chain tab, you can define the gas limit and the gas price for your private network.

5. **Advanced tab**



⚠ **Restarting the Quickstart workspace resets the blockchain.** All transactions and contract states will be reset.

LOGGING

OUTPUT LOGS TO FILE ○ — Save logs to file.

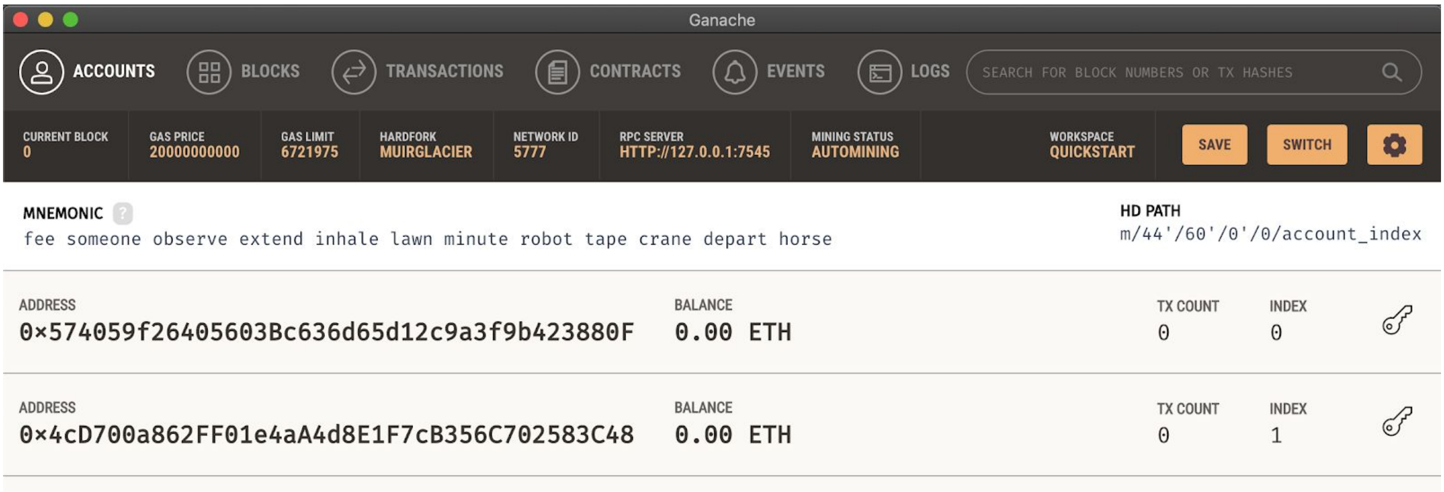
VERBOSE LOGS ○ — Increase the log output.

ANALYTICS

GOOGLE ANALYTICS ○ — We use Google Analytics to track Ganache usage. This information helps us gain more insight into how Ganache is used. This tracking is anonymous. We do not track personally identifiable information, account data or private keys. Note: This setting is global and will persist between workspaces.

In the Advanced tab, you can choose to output the log that Ganache creates.

Ganache dashboard



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

MNEMONIC
fee someone observe extend inhale lawn minute robot tape crane depart horse

HD PATH
m/44'/60'/0'/0/account_index

ADDRESS	BALANCE	TX COUNT	INDEX
0x574059f26405603Bc636d65d12c9a3f9b423880F	0.00 ETH	0	0
0x4cD700a862FF01e4aA4d8E1F7cB356C702583C48	0.00 ETH	0	1

In the main dashboard window of Ganache, you can see six tabs:

1. **Accounts**
   - Here, you can see the details of user accounts.

2. **Blocks**
    - Here, you will see the details of the block that will be created.

3. **Transactions**
    - Here, you can see the transactions details.

4. **Contracts**
    - Here, you will see the list of contracts that you have created for this project.

5. **Events**
    - Here, you will see the list of the events triggered by the smart contract.

6. **Logs**
    - Here, you will see the logs of your entire project.

Smart Contract Deployment Using Ganache

Steps involved in deployment using Ganache:

- Create a workspace and link your Truffle project to this workspace.
- Configure your *truffle-config.js* file to connect to the *Ganache* network.
- Migrate your Truffle project to this network and create an instance of *Voting* smart contract.
- Run various functions inside this smart contract using this instance.

Refer to this document for detailed steps and commands used for each step.