# Forking Ethereum Mainnet

When you want to interact with contracts that are deployed on the Ethereum mainnet for development purpose without spending any ether you can fork the mainnet from a particular block number and then interact with that forked mainnet on your local machine.

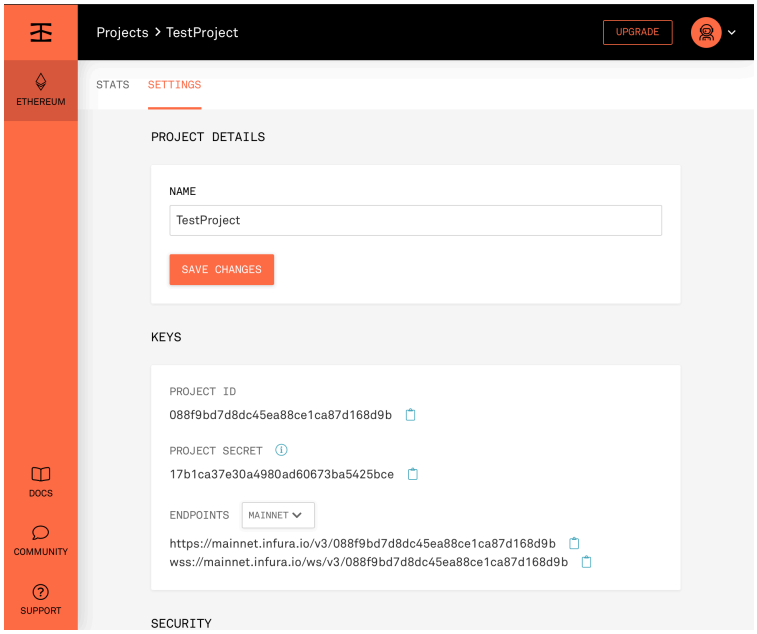
So, today we will take a look at how you can fork the mainnet and interact with DSA contracts and spells.

## Step: 1  Open terminal on your machine and run

`ganache-cli --fork https://mainnet.infura.io/v3/{Project_Id} --unlock {Ethereum_Address} -p {Port_Number} --networkId 1`

Use the latest version of ganache-cli to run this command which currently is v6.9.1.

{Project_Id} : This Id comes from infura. If you don't know about how to get this, just open infura and create a free account. After creating account add a new project and you will get the Project_Id that you need. Sample Project_Id has been shown below.




{Ethereum_Address} : On forked mainnet you can unlock any account of your choice without knowing their private key. So, let's say you want to test  something which requires transaction of DAI or other coin. So, you can choose an account which has lots of DAI. This way you can unlock any account of your choice.

{Port_Number}: It is the port number on which you want to run the ganache-cli. The default port number for ganache-cli is 8545

If you want to fork mainnet at a particular block number you can do that by providing value of Block_Number after infura Project_ID

`ganache-cli --fork https://mainnet.infura.io/v3/{Project_Id}@{Block_Number} --unlock {Ethereum_Address} -p {Port_Number} --networkId 1`

After running the command, it will return you something like this:

```
Ganache CLI v6.9.1 (ganache-core: 2.10.2)

Available Accounts
==================
(0) 0xCCD4360C0828C7312aEFd88a3858C3cd4315D4E3 (100 ETH)
(1) 0xE2Cdd45307Cde68c87243255c2344aabd00CeA45 (100 ETH)
(2) 0x18665DB2035B0358c643902E882fD5Ef0fb7c283 (100 ETH)
(3) 0x28c64E55E86B49E730830AF92a90727411e7188e (100 ETH)
(4) 0xB29a69E9b39256Cf82D25a94c7545F6Cbb40220C (100 ETH)
(5) 0x5b412fe4aF46BF389E8C0c693DB2B2069dD63B90 (100 ETH)
(6) 0xb203E8Cb654BaD5200F67578F9f67b33bAd8538B (100 ETH)
(7) 0x052EA1e36d810e825016529DF6408c76d69A8982 (100 ETH)
(8) 0x9042fc8C70A0bF6797b3D706dE81c6ce278B5DD1 (100 ETH)
(9) 0x0BE40C39fd6C915Dc18CBc6d740e2800E8b389C7 (100 ETH)

Private Keys
==================
(0) 0x40870d0c4897098b69a51632ef839704c14c1f4bbe8ca5a44774672cdbc0d2dc
(1) 0x5b5e687f77800667f59bb01b206d735f2b58253bdd9b41da30b98e90648638e8
(2) 0x5cc3bf01f2f6a95623098424084dc3d4b9e3f736022888ab1fe26f889d2cac4c
(3) 0xcb1bbfca1132d1d180f6bfe5fa23f3612377a3f12ce3b9d20376f3fd722a8522
(4) 0xa20f39183be0d12740bc8b6a743e91cbd2d49a1af399abce42e6a9e937c7689c
(5) 0x3d560aa715ad8a886ff479757af7514d1d7d1051b3358298a3870f86b80d5d13
(6) 0x66b283c0bec386a2b9facb164ff0aedb56698eafe00a4b7ac008eb996d6e348a
(7) 0x1d6f28bc8a99e5c4fdf23083768471ba346df16457a2fd771c0d35658f144739
(8) 0x49d116463ca90cc8eed4585c72ce06b1b3228f03fea16af9beb080e7ea0d356f
(9) 0x16bd153409fbd18548f13cd1d00a49f511fa51591fe28a347c9d575e4f6a788b

HD Wallet
==================
Mnemonic:      uniform dish awake under churn hammer skirt supply rib forum ginger accuse
Base HD Path:  m/44'/60'/0'/0/{account_index}

Gas Price
==================
20000000000

Gas Limit
==================
6721975

Call Gas Limit
==================
9007199254740991

Forked Chain
==================
Location:    https://mainnet.infura.io/v3/cfae4c61fef54529a22942124af4f4a3
Block:       10226733
Network ID:  1
Time:        Mon Jun 08 2020 23:51:02 GMT+0530 (India Standard Time)

Listening on 127.0.0.1:7545
```

And as you can see it is mentioning that it is Forked Chain and also provides details about chain like Block number you forked that mainnet from and also the network id.

**Note:** If Block's value in the Forked Chain characteristics is '0' then it means that you have not been able to fork the chain properly.

## Step 2: Interacting with DSA using the Forked Chain

Create a new Directory on your machine and Clone this repo on your machine.

After cloning run `npm install` in the cloned repo directory.

Before interacting with DSA make sure you have configured .env file which contains the private key, Ethereum address and Infura project Id.

Now, open **dsa.js** file and check if the port number of your ganache-cli matches with the web3 providers.

```
//  Instantiation of web3 object
const web3 = new Web3(new Web3.providers.HttpProvider('http://localhost:7545'))
```

Now, you can interact with the DSA by using the **dsa.js** file. If you don't have any DSA account you can create one by uncommenting this section from **dsa.js** file.

```
//  BUILD FUNCTIONALITY


/*

    If you want to create a DSA account uncomment this section

    async function getTxnCount() {
        return await web3.eth.getTransactionCount(process.env.PUBLIC_ADDRESS);
    }

    async function buildWallet() {
      const nonce = await getTxnCount();
      dsa.build({
        gasPrice: web3.utils.toHex(web3.utils.toWei(gasPrice, 'gwei')),
        gasLimit: web3.utils.toHex(gasLimit),
        nonce: nonce
      }).then(txHash => {
        console.log(`https://etherscan.io/tx/${txHash}`)
      })
    }

*/
```

The file contains a section where you can add your spells by default it contains spells for deposit and withdraw methods of Compound Connector.

File also contains methods for account setup and transfer.

## Step 3: executing the file

```
// Adding the spells

// These are just the Sample Spells
// You can add any spell of your choice

spells.add({
    connector: "compound",
    method: "deposit",
    args: [dsa.tokens.info.eth.address,dsa.tokens.fromDecimal(0.01,"eth") , 0, 0]
})

spells.add({
    connector: "compound",
    method: "withdraw",
    args: [dsa.tokens.info.eth.address,dsa.tokens.fromDecimal(0.01,"eth") , 0, 0]
})
```

After making all the changes in the file execute the file using command:

`node dsa.js`

```
[
  {
    id: '431',
    address: '0x015710BC1dFc33640D9cD7e3673779b917e10382',
    version: '1'
  },
  {
    id: '707',
    address: '0x4B343019202783fC701a331F872000AC70273814',
    version: '1'
  },
  {
    id: '708',
    address: '0x2d40bc02a801257B006aCb8a21AFe95d1D39F653',
    version: '1'
  },
  {
    id: '709',
    address: '0xce7776965EdBB4b53a4d748F1f77F93d2739C2bd',
    version: '1'
  }
]
Account has been set !!!
0xbce91355ebc215b0f40e2b2a8ca9e1c9855d3e71c63533346521d9bfdc2feb22
```

This may take few minutes as ganache has to fetch contracts from mainnet and then work on them so be patient, if you want to know if it has stuck or not then head to the tab on console where the ganache-cli is working and check if it is making some calls. This will return something like shown in the image.

Array in the output shows all the DSA that are linked to your Ethereum Address.
And the hash in the last line is the Transaction Hash that *dsa.cast()* function returns.

To check the transaction receipt you can follow these steps:

Step 1: Open another terminal tab

Step 2: run ‘truffle console’ (make sure you have truffle installed on your machine)

Step 3: In the truffle console, run `*receipt = await web3.eth.getTransactionReceipt({txHash})*`

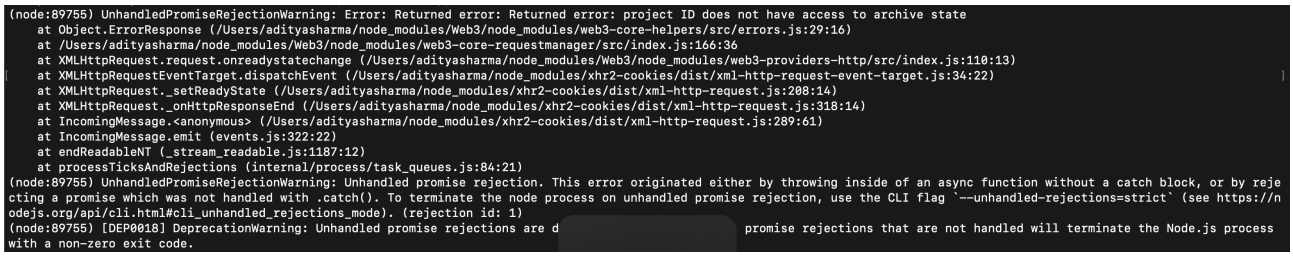
```
truffle(development)> receipt = await web3.eth.getTransactionReceipt('0x9e8ce71c6e8a56cd0d5dc8d6f88d96e7d898d89e7b3979af7ca771b034008a9b')
undefined
truffle(development)> receipt
{
  transactionHash: '0x9e8ce71c6e8a56cd0d5dc8d6f88d96e7d898d89e7b3979af7ca771b034008a9b',
  transactionIndex: 0,
  blockHash: '0x4373772d332152b4ee2d73446777280bd858e9377ea7eaef6c060fab0dc7e49a',
  blockNumber: 10225683,
  from: '0xb116f194179418b24713a6535149ef637fa13325',
  to: '0x015710bc1dfc33640d9cd7e3673779b917e10382',
  gasUsed: '0x46817',
  cumulativeGasUsed: 288791,
  contractAddress: null,
  logs: [
    {
      logIndex: 0,
      transactionIndex: 0,
      transactionHash: '0x9e8ce71c6e8a56cd0d5dc8d6f88d96e7d898d89e7b3979af7ca771b034008a9b',
      blockHash: '0x4373772d332152b4ee2d73446777280bd858e9377ea7eaef6c060fab0dc7e49a',
      blockNumber: 10225683,
      address: '0x4Ddc2D193948926D02f9B1fE9e1daa0718270ED5',
      data: '0x0000000000000000000000000000000000000000000000001790b941b164800000000000000000000000000000000000000000000000000000000000e4378cd9acbf0280000000000000000000000000000000000000000000000000000006829fa1992f0b5d
9e0',
      topics: [Array],
      type: 'mined',
      id: 'log_1514fef1'
    },
    {
      logIndex: 1,
      transactionIndex: 0,
      transactionHash: '0x9e8ce71c6e8a56cd0d5dc8d6f88d96e7d898d89e7b3979af7ca771b034008a9b',
      blockHash: '0x4373772d332152b4ee2d73446777280bd858e9377ea7eaef6c060fab0dc7e49a',
      blockNumber: 10225683,
      address: '0x4Ddc2D193948926D02f9B1fE9e1daa0718270ED5',
      data: '0x000000000000000000000000015710bc1dfc33640d9cd7e3673779b917e1038200000000000000000000000000000000000000000000000000000000002386f26fc1000000000000000000000000000000000000000000000000000000000000002fa7
a56',
      topics: [Array],
      type: 'mined',
      id: 'log_4a4e8542'
    },
    {
      logIndex: 2,
      transactionIndex: 0,
      transactionHash: '0x9e8ce71c6e8a56cd0d5dc8d6f88d96e7d898d89e7b3979af7ca771b034008a9b',
      blockHash: '0x4373772d332152b4ee2d73446777280bd858e9377ea7eaef6c060fab0dc7e49a',
      blockNumber: 10225683,
      address: '0x4Ddc2D193948926D02f9B1fE9e1daa0718270ED5',
      data: '0x0000000000000000000000000000000000000000000000000000000002fa7a56',
      topics: [Array],
      type: 'mined',
      id: 'log_16881f86'
    },
```

This will return you something like this, if your transaction went through.

## Additional advice:

If you want to use any particular token then you can unlock address which holds that token and you can get list of all token holders on
ethplorer.

If you encounter error saying : “Error: Returned error: Returned error: project ID does not have access to archive state”. Trying reloading the ganache-cli as this will most probably fix this.




**NOTE:** This is not the best way to interact with DSA as you might face a lot of errors in the process of interacting on forked mainnet. The best way is still the Ethereum Mainnet.