

wallet

August 6, 2020

```
[182]: from constants import *
import os
import subprocess
import json
from web3 import Web3
from dotenv import load_dotenv
from web3.middleware import geth_poa_middleware
from eth_account import Account

from pathlib import Path
from getpass import getpass

from bit import wif_to_key
from bit import PrivateKeyTestnet
from bit.network import NetworkAPI

load_dotenv()
```

[182]: True

```
[183]: # enter your personal mnemonic here

mnemonic = os.getenv('MNEMONIC', 'work want afford grace rubber build chair_
↳ sudden valley certain capital club')
```

```
[184]: # Function to provide your wallet address

def wallet_address(coins, depth):

    command = f"./derive -g --mnemonic='{mnemonic}'_
↳ --cols=path,address,privkey,pubkey --coin={coins} --numderive={depth}_
↳ --format=json"

    p = subprocess.Popen(command, stdout=subprocess.PIPE, shell=True)
    (output, err) = p.communicate()
    keys = json.loads(output)
```

```
return keys
```

```
[185]: #ETH wallets  
wallet_address("ETH",3)
```

```
[185]: [{ 'path': "m/44'/60'/0'/0/0",  
        'address': '0x9eDeBC9f308848F247595e663e0796CFF62A965a',  
        'privkey':  
'0x75ade74425d7272a2a82f77efb632fe7e5b706f6fbe7da72637da3f795e0ab7e',  
        'pubkey':  
'032da0dccf8596d52496fe849671caae79748d17dfe8a77f6f97d35deb99bd498e'},  
  { 'path': "m/44'/60'/0'/0/1",  
    'address': '0x146ff600884a2c5e25216e30b52EC7D033E797d6',  
    'privkey':  
'0xe7a541bd8483c99a489baeaf337ac3acc7175d4aa1b444c1632a45db21b1a039',  
    'pubkey':  
'02656ea756cf62d73dc3575bca882a59315c8fd7553334e295feef3694e0aae8d'},  
  { 'path': "m/44'/60'/0'/0/2",  
    'address': '0xDdAAEDa1d157C6fFFE39D1405292F480c9f74667',  
    'privkey':  
'0xa365b8fbe04d615293db50d379c397d68898306e5845edf3a4fd435fd62a2c7c',  
    'pubkey':  
'03b520019fb871ceb18b7271f11f314965a86cf64b534774bc360cfcbc55816954' } ]
```

```
[186]: #BTC Test Wallets  
wallet_address("btc-test",3)
```

```
[186]: [{ 'path': "m/44'/1'/0'/0/0",  
        'address': 'mfcheUzJTkJ4tHHz2HyGqfDxL9uUkQR9W6',  
        'privkey': 'cRSFSFMH242PYahfUaM3HMMkRwGMc8jyALZSkVnh6QEtaQJXo8dw',  
        'pubkey':  
'036f354c0dba78ce405f394ae9422991db609fd500eb2de3ba1bbf7ce726f206'},  
  { 'path': "m/44'/1'/0'/0/1",  
    'address': 'mnHT7DTSDV5LpSkz6RNBoQV4GpP1s78Pup',  
    'privkey': 'cNN53odghuVBttKNZDxWon2ziYXNqburhEUoRNPUuTkt3WHSF1re',  
    'pubkey':  
'03dd0e3a3cfd42a88562c37b5dc469786431ab6e5c2090c7447bb2b4ffc19bb84c'},  
  { 'path': "m/44'/1'/0'/0/2",  
    'address': 'n4aL4JJVbthRSHWNPQ8uJ4nkoM8E9GUBcM',  
    'privkey': 'cUTPsFTUJNGRwSw8TTySSdGhoWatToCoAnzNXnTcDkMUT2aGA2DN',  
    'pubkey':  
'022a1972f906e4c3aafd33993e1aa98e9ccdd59914c8eaa26b00c646b49bd70f1e' } ]
```

```
[187]: #BTC wallets  
wallet_address("btc",3)
```

```
[187]: [{ 'path': "m/44'/0'/0'/0/0",
        'address': '1zH27dQfYV7iwJtdoZ6hVj1jpTVVitoWi',
        'privkey': 'Kzh65oY6VdngSogcGNBh1BM4ZAfDVwZxzkcMcVoKBgPDhvXGxVSQV',
        'pubkey':
'020862ffac75af0536eb646f096ba804f4817a0ac2028ce73a2f015d321b7115df'},
  { 'path': "m/44'/0'/0'/0/1",
    'address': '1XMxjdGgA8ZQUqsHzxWtR5MMAeXWg93xR',
    'privkey': 'KzpooJXccq9X8wpooTHcRRBw9L4mVShyJQWapEhpnuJi6hiBowgs',
    'pubkey':
'02096e1edaf47ef43339b4c1d166c6f513d273b2dc5504e364d5670bed8755d58c'},
  { 'path': "m/44'/0'/0'/0/2",
    'address': '1KbYQsKfjgcqXS24EuQJRQjm4a2qMg1hXV',
    'privkey': 'L1jwpSyeYaZao58XTodLBDiSSXUayhxjE6p5wzSPiuyVggSJfsR4',
    'pubkey':
'03832e4ffcf77745c7a35de2f11de1eee280f7289e914535e3c3555202f328e8c' }]
```

1 Linking the transaction signing libraries

```
[188]: def priv_key_to_account(coin,priv_key):
        if coin == "ETH":
            w3 = Web3(Web3.HTTPProvider("http://127.0.0.1:8545"))
            w3.middleware_onion.inject(geth_poa_middleware, layer=0)

            #account_one = Account.from_key(os.getenv("PRIVATE_KEY"))
            #account_one = Account.from_key(PRIVATE_KEY)
            account_one = Account.privateKeyToAccount(priv_key)
            print(account_one)
            return account_one

        elif coin == "btc-test":
            btc_account= PrivateKeyTestnet(priv_key)

            return btc_account
```

```
[189]: def create_tx(coin, account, recipient, amount):
        if coin == "ETH":
            gasEstimate = w3.eth.estimateGas(
                {"from": account.address, "to": recipient, "value": amount}
            )
            return {
                "from": account.address,
                "to": recipient,
                "value": amount,
                "gasPrice": round(w3.eth.gasPrice),
                "gas": gasEstimate,
                "nonce": w3.eth.getTransactionCount(account.address),
```

```

    }

    elif coin == "btc-test":
        btc_tx = PrivateKeyTestnet.prepare_transaction(account.address,
→ [(recipient, amount, BTC)])

        return btc_tx

```

```

[190]: def send_tx(coin, account, recipient, amount):
        if coin == "ETH":
            tx = create_tx(coin, account, recipient, amount)
            signed_tx = account.sign_transaction(tx)
            result = w3.eth.sendRawTransaction(signed_tx.rawTransaction)
            print(result.hex())
            return result.hex()

        elif coin == "btc-test":
            btc_tx = create_tx(coin, account, recipient, amount)
            tx_hex = key.sign_transaction(btc_tx)
            result = NetworkAPI.broadcast_tx_testnet(tx_hex)

            return tx_hex

```

2 Sending ETH Transaction

```

[195]: # Enter your ETH private key with funding

eth_address1 = "0x9eDeBC9f308848F247595e663e0796CFF62A965a"
eth_priv_key1 =
→ '0x75ade74425d7272a2a82f77efb632fe7e5b706f6fbe7da72637da3f795e0ab7e'
#priv_key2 =
→ '0xe7a541bd8483c99a489baeaf337ac3acc7175d4aa1b444c1632a45db21b1a039'
account_one = priv_key_to_account("ETH", eth_priv_key1)
account_one

```

<eth_account.signers.local.LocalAccount object at 0x7ff96b2a3b90>

[195]: <eth_account.signers.local.LocalAccount at 0x7ff96b2a3b90>

```

[194]: # Check the balance

w3 = Web3(Web3.HTTPProvider("http://127.0.0.1:8545"))
w3.eth.getBalance(eth_address1)

```

[194]: 89968411839965552223


```
[199]: AttributeDict({'hash':
HexBytes('0x92059faea613acf46f772ef649a4afe81198b35cdcefaf1469d93195a1592990'),
'nonce': 15,
'blockHash':
HexBytes('0x4f82eb4ca5d8eff6f355b00deca905b723f7b3a22f4cefb2b3d68d4708816084'),
'blockNumber': 21,
'transactionIndex': 0,
'from': '0x9eDeBC9f308848F247595e663e0796CFF62A965a',
'to': '0x146ff600884a2c5e25216e30b52EC7D033E797d6',
'value': 8888,
'gas': 21000,
'gasPrice': 20000000000,
'input': '0x',
'v': 27,
'r':
HexBytes('0x268cae7a2e89ef849b79060a7bc00672c857599166e3cf6eeaa2054c765dc367'),
's':
HexBytes('0x7657b8f7d60379aae71f331aea76a5a93f03b07eec075fb4ea403894c08497d1')})
```

3 Sending BTC-Test Transaction

```
[200]: #enter your private key and check the balance

btc_priv_key1="cRSFSFMH242PYahfUaM3HMMkRwGMc8jyALZSkVnh6QEtaQJXo8dw"

key = wif_to_key(btc_priv_key1)

key.get_balance('btc')
```

```
[200]: '0.00936432'
```

```
[201]: # get BTC account address from private key

btc_account_one = priv_key_to_account("btc-test",btc_priv_key1)
btc_account_one
```

```
[201]: <PrivateKeyTestnet: mfcheUzJTkJ4tHHz2HyGqfDxL9uUkQR9W6>
```

```
[202]: # Create unsigned transaction ticket using the function above and enter the
↪recipient and amount

recipient_btc1 = "mnHT7DTSdV5LpSkz6RNBoQV4GpP1s78Pup"
create_tx("btc-test",btc_account_one,recipient_btc1,0.003)
```

```
[202]: '{"unspents": [{"amount":1000000,"confirmations":-1804785,"script":"76a914011644b7b5d9b86c6059fe2ca6614bf26494736c88ac","txid":"d8e73759ffac19660cf7b0cc34a4f58df
```

```
e5eb600916b60ac9b84fb35549fab84","txindex":0,"type":"p2pkh","vsize":148,"segwit":false},{ "amount":936432,"confirmations":-1804762,"script":"76a914011644b7b5d9b86c6059fe2ca6614bf26494736c88ac","txid":"6c1d530b6fc30d2d9d5057c85a005c1008e0da27714cb413db7b814057e392be","txindex":1,"type":"p2pkh","vsize":148,"segwit":false}], "outputs": [[ "mnHT7DTSdV5LpSkz6RNBoQV4GpP1s78Pup",300000], [ "mfcheUzJTkJ4tHHz2HyGqfDxL9uUkQR9W6",1555648]]}]'
```

```
[204]: # now signed the contract and send transaction ticket
send_tx("btc-test",btc_account_one,receipient_btc1,0.003)
```

```
[204]: '010000000284ab9f5435fb849bac606b9100b65efe8df5a434ccb0f70c6619acff5937e7d800000
0006b483045022100f2115d25af4a27db3e8cfc778bb412fce9b7dd435a9e645176ed3715acdf908
c02202c5fa6ae94aadf4f539bb0bac035b078bfa73d8a21875ed3b373bf16965b42bf0121036f354
c0dba78ce405f394ae9422991db609fd500eb2de3ba1bbf7ce726f206fffffffff92e35740817
bdb13b44c7127dae008105c005ac857509d2d0dc36f0b531d6c010000006a47304402201246757fe
6151d647e0de1180e74b093aeff2cfa0d0074e1e0d379ae1be9f4a602204a89378f616292012f217
8a4655ef96b85784568b87b8966dd5a63fb2b4e8f720121036f354c0dba78ce405f394ae942299
1db609fd500eb2de3ba1bbf7ce726f206fffffffff02e0930400000000001976a9144a3b1be828b8c
b028c86edca61d8fc12c91237e188acc0bc1700000000001976a914011644b7b5d9b86c6059fe2ca
6614bf26494736c88ac00000000'
```