```
$ curl https://nodejs.org/dist/v16.17.0/node-v16.17.0-darwin-x64.tar.gz -o node-v16.17.0-darwin-x64.tar.gz
$ tar zxf node-v16.17.0-darwin-x64.tar.gz
$ export PATH=`pwd`/node-v16.17.0-darwin-x64/bin:$PATH
$ npm install keythereum
$ node
> let ac = "0x3590a22d97c4aadfa57f8403fac37011f47dbc0"
> let pass = "user1"
> let ke = require("keythereum");
> let ks = "<geth init で指定したデータパス>"
> let ko = ke.importFromFile(ac, ks)
> let pk = ke.recover(pass, ko)
> pk.toString('hex')
```

リスト 1

```
(mydata_server)   % cd LIB
(mydata_server)   % ./setethereum.py --id Ethereum --name "Etherem private net" --account 0x70f9e0445a572d62eabf6d6669f69283558d7e2f --password dataprovider --privatekey
c2f7b10487844a8d0f708a2397a63e0aa2b2695351775fe5548557bc71280e28 --address=192.168.3.3
Namespace(id='Ethereum', name='Etherem private net', address='192.168.3.3', port=8545, account='0x70f9e0445a572d62eabf6d6669f69283558d7e2f', password='dataprovider',
privatekey='c2f7b10487844a8d0f708a2397a63e0aa2b2695351775fe5548557bc71280e28')
(mydata_server) %
```

リスト 2

```
(mydata_server) % mysql -p -u root
Enter password:
mysql> use mydataserver;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from ds_ethereum;
+----------+---------------------+-----------+------+------------------------------------------+--------------+------------------------------------------------------------------
-------+
| id       | name                | address   | port | account                                  | password     | private_key                                                      |
+----------+---------------------+-----------+------+------------------------------------------+--------------+------------------------------------------------------------------
-------+
| Ethereum | Etherem private net | 192.168.3.3 | 8545 | 0x70f9e0445a572d62eabf6d6669f69283558d7e2f | dataprovider |
c2f7b10487844a8d0f708a2397a63e0aa2b2695351775fe5548557bc71280e28 |
+----------+---------------------+-----------+------+------------------------------------------+--------------+------------------------------------------------------------------
-------+
1 row in set (0.00 sec)

mysql>
```

リスト 3

```
(mydata_server) % pip install web3
Collecting web3
  Downloading web3-5.30.0-py3-none-any.whl (501 kB)
     |████████████████████████████████| 501 kB 2.1 MB/s
Collecting eth-abi<3.0.0,>=2.0.0b6
  Downloading eth_abi-2.2.0-py3-none-any.whl (28 kB)
Collecting eth-hash[pycryptodome]<1.0.0,>=0.2.0
  Downloading eth_hash-0.5.0-py3-none-any.whl (8.9 kB)
Collecting eth-account<0.6.0,>=0.5.7
  Downloading eth_account-0.5.9-py3-none-any.whl (101 kB)
     |████████████████████████████████| 101 kB 4.2 MB/s
Collecting ipfshttpclient==0.8.0a2
  Downloading ipfshttpclient-0.8.0a2-py3-none-any.whl (82 kB)
```

```
     |████████████████████████████████| 82 kB 723 kB/s
Collecting aiohttp<4,>=3.7.4.post0
  Downloading aiohttp-3.8.1-cp39-cp39-macosx_10_9_x86_64.whl (574 kB)
     |████████████████████████████████| 574 kB 3.2 MB/s
Collecting hexbytes<1.0.0,>=0.1.0
  Downloading hexbytes-0.3.0-py3-none-any.whl (6.4 kB)
Collecting jsonschema<5,>=3.2.0
  Downloading jsonschema-4.14.0-py3-none-any.whl (82 kB)
     |████████████████████████████████| 82 kB 1.1 MB/s
Requirement already satisfied: requests<3.0.0,>=2.16.0 in /Users/tsuchiya/opt/anaconda3/envs/mydata_server/lib/python3.9/site-packages (from web3) (2.28.1)
Collecting websockets<10,>=9.1
  Downloading websockets-9.1-cp39-cp39-macosx_10_9_x86_64.whl (88 kB)
     |████████████████████████████████| 88 kB 4.6 MB/s
Collecting protobuf<4,>=3.10.0
  Downloading protobuf-3.20.1-cp39-cp39-macosx_10_9_x86_64.whl (962 kB)
     |████████████████████████████████| 962 kB 5.1 MB/s
Collecting eth-utils<2.0.0,>=1.9.5
  Downloading eth_utils-1.10.0-py3-none-any.whl (24 kB)
Collecting lru-dict<2.0.0,>=1.1.6
  Downloading lru_dict-1.1.8-cp39-cp39-macosx_10_9_x86_64.whl (9.9 kB)
Collecting eth-typing<3.0.0,>=2.0.0
  Downloading eth_typing-2.3.0-py3-none-any.whl (6.2 kB)
Collecting eth-rlp<0.3
  Downloading eth_rlp-0.2.1-py3-none-any.whl (5.0 kB)
Collecting multiaddr>=0.0.7
  Downloading multiaddr-0.0.9-py2.py3-none-any.whl (16 kB)
Collecting yarl<2.0,>=1.0
  Downloading yarl-1.8.1-cp39-cp39-macosx_10_9_x86_64.whl (61 kB)
     |████████████████████████████████| 61 kB 1.6 MB/s
Collecting async-timeout<5.0,>=4.0.0a3
  Downloading async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Collecting multidict<7.0,>=4.5
  Downloading multidict-6.0.2-cp39-cp39-macosx_10_9_x86_64.whl (28 kB)
Collecting aiosignal>=1.1.2
  Downloading aiosignal-1.2.0-py3-none-any.whl (8.2 kB)
Collecting frozenlist>=1.1.1
  Downloading frozenlist-1.3.1-cp39-cp39-macosx_10_9_x86_64.whl (36 kB)
Requirement already satisfied: charset-normalizer<3.0,>=2.0 in /Users/tsuchiya/opt/anaconda3/envs/mydata_server/lib/python3.9/site-packages (from aiohttp<4,>=3.7.4.post0->web3)
(2.1.1)
Collecting attrs>=17.3.0
  Downloading attrs-22.1.0-py2.py3-none-any.whl (58 kB)
     |████████████████████████████████| 58 kB 4.2 MB/s
Collecting parsimonious<0.9.0,>=0.8.0
  Downloading parsimonious-0.8.1.tar.gz (45 kB)
     |████████████████████████████████| 45 kB 443 kB/s
Collecting eth-keys<0.4.0,>=0.3.4
  Downloading eth_keys-0.3.4-py3-none-any.whl (21 kB)
Collecting bitarray<3,>=1.2.1
  Downloading bitarray-2.6.0-cp39-cp39-macosx_10_9_x86_64.whl (105 kB)
     |████████████████████████████████| 105 kB 7.8 MB/s
Collecting rlp<3,>=1.0.0
  Downloading rlp-2.0.1-py2.py3-none-any.whl (20 kB)
Collecting eth-keyfile<0.6.0,>=0.5.0
  Downloading eth_keyfile-0.5.1-py3-none-any.whl (8.3 kB)
Collecting pycryptodome<4,>=3.6.6
  Downloading pycryptodome-3.15.0-cp35-abi3-macosx_10_9_x86_64.whl (1.6 MB)
     |████████████████████████████████| 1.6 MB 2.7 MB/s
Collecting cytoolz<1.0.0,>=0.9.0
  Downloading cytoolz-0.12.0-cp39-cp39-macosx_10_9_x86_64.whl (410 kB)
     |████████████████████████████████| 410 kB 2.5 MB/s
Collecting toolz>=0.8.0
  Downloading toolz-0.12.0-py3-none-any.whl (55 kB)
     |████████████████████████████████| 55 kB 929 kB/s
Collecting eth-utils<2.0.0,>=1.9.5
```

```
  Downloading eth_utils-1.9.5-py3-none-any.whl (23 kB)
Collecting pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0
  Downloading pyrsistent-0.18.1-cp39-cp39-macosx_10_9_universal2.whl (81 kB)
     |████████████████████████████████| 81 kB 7.1 MB/s
Collecting varint
  Downloading varint-1.0.2.tar.gz (1.9 kB)
Requirement already satisfied: six in /Users/tsuchiya/opt/anaconda3/envs/mydata_server/lib/python3.9/site-packages (from multiaddr>=0.0.7->ipfshttpclient==0.8.0a2->web3) (1.16.0)
Collecting netaddr
  Downloading netaddr-0.8.0-py2.py3-none-any.whl (1.9 MB)
     |████████████████████████████████| 1.9 MB 2.8 MB/s
Collecting base58
  Downloading base58-2.1.1-py3-none-any.whl (5.6 kB)
Requirement already satisfied: certifi>=2017.4.17 in /Users/tsuchiya/opt/anaconda3/envs/mydata_server/lib/python3.9/site-packages (from requests<3.0.0,>=2.16.0->web3)
(2021.10.8)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /Users/tsuchiya/opt/anaconda3/envs/mydata_server/lib/python3.9/site-packages (from requests<3.0.0,>=2.16.0->web3) (1.26.9)
Requirement already satisfied: idna<4,>=2.5 in /Users/tsuchiya/opt/anaconda3/envs/mydata_server/lib/python3.9/site-packages (from requests<3.0.0,>=2.16.0->web3) (3.3)
Building wheels for collected packages: parsimonious, varint
  Building wheel for parsimonious (setup.py) ... done
  Created wheel for parsimonious: filename=parsimonious-0.8.1-py3-none-any.whl size=42724 sha256=9c07df9b9048ef106b7f4f8c774edfbcee3e1642e30f453479a0e772eff0bd3c
  Stored in directory: /Users/tsuchiya/Library/Caches/pip/wheels/ae/3f/87/24298980f6c0436680b8b64caac154f542e138487ed0f90f2a
  Building wheel for varint (setup.py) ... done
  Created wheel for varint: filename=varint-1.0.2-py3-none-any.whl size=1979 sha256=c52c0a8e1ef866ea398ff7a31ebd4ab9a959d9d852beb87d297927e214bfc2cd
  Stored in directory: /Users/tsuchiya/Library/Caches/pip/wheels/4b/47/bb/e4fd5cf6101ed8d6a9c52ff50e37bfb908ffdf330ddbbf9550
Successfully built parsimonious varint
Installing collected packages: toolz, eth-typing, eth-hash, cytoolz, eth-utils, varint, rlp, pycryptodome, parsimonious, netaddr, multidict, hexbytes, frozenlist, eth-keys, base58, yarl, pyrsistent,
multiaddr, eth-rlp, eth-keyfile, eth-abi, bitarray, attrs, async-timeout, aiosignal, websockets, protobuf, lru-dict, jsonschema, ipfshttpclient, eth-account, aiohttp, web3
  Attempting uninstall: protobuf
    Found existing installation: protobuf 4.21.2
    Uninstalling protobuf-4.21.2:
      Successfully uninstalled protobuf-4.21.2
Successfully installed aiohttp-3.8.1 aiosignal-1.2.0 async-timeout-4.0.2 attrs-22.1.0 base58-2.1.1 bitarray-2.6.0 cytoolz-0.12.0 eth-abi-2.2.0 eth-account-0.5.9 eth-hash-0.5.0
eth-keyfile-0.5.1 eth-keys-0.3.4 eth-rlp-0.2.1 eth-typing-2.3.0 eth-utils-1.9.5 frozenlist-1.3.1 hexbytes-0.3.0 ipfshttpclient-0.8.0a2 jsonschema-4.14.0 lru-dict-1.1.8 multiaddr-0.0.9
multidict-6.0.2 netaddr-0.8.0 parsimonious-0.8.1 protobuf-3.20.1 pycryptodome-3.15.0 pyrsistent-0.18.1 rlp-2.0.1 toolz-0.12.0 varint-1.0.2 web3-5.30.0 websockets-9.1 yarl-1.8.1
(mydata_server) %
```

リスト4

```
 1 #!/usr/bin/env python
 2 # -*- coding: utf-8 -*-
 3
 4
 5 import sys
 6 import json
 7 import datetime
 8 from web3 import Web3
 9
10 DEBUG = False #True
11
12
13 # Ethereum にアクセスするライブラリ
14
15 class Ethereum:
16     # コンストラクタ
17     def __init__(self, url, account, password, private_key):
18         print("Constructor of Ethereum")
19         self.url = url
20         self.account = account
21         self.password = password
22         self.private_key = private_key
23         self.contracts = {}
24
25         # Ethereum へ接続するクライアント作成
```

```
26          self.w3 = Web3(Web3.HTTPProvider(url))
27
28
29
30      # デストラクタ
31      def __del__(self):
32          print("Destructor of Ethereum")
33
34
35      # 実行するコントラクトの情報
36      def set_contract(self, addr, abi):
37          print("set_contract")
38          self.cnt_addr = addr
39          self.cnt_abi = abi
40          self.cnt = self.w3.eth.contract(address=self.w3.toChecksumAddress(addr), abi=abi)
41
42
43      # データ数取り出し
44      def get_data_number(self, channel_id,   measurement):
45          print("get_data_number")
46
47          res = self.cnt.functions.get_data_num(channel_id,   measurement).call()
48          return res
49
50      # データ取り出し
51      def get_data(self, channel_id,   measurement, start=0, stop=10, source=None):
52          print("get_data")
53
54          res = self.cnt.functions.get_data(channel_id,   measurement, start, stop).call()
55          if DEBUG : print("RES = ",res)
56          return res
57
58      # データ保存
59      def put_data(self, channel_id, measurement, data):
60          print("put_data")
61          """
62          data  を  timestamp とデータに分ける
63          """
64          d = json.loads(data)
65          if DEBUG:
66              print(d)
67
68          timestamp = d['timestamp']
69          if timestamp == None:
70              timestamp = datetime.datetime.utcnow().strftime('%Y%m%d%H%M%S')
71
72          ch_ac = self.w3.toChecksumAddress(self.account)
73          tx = self.cnt.functions.put_data(channel_id, measurement, timestamp, data).buildTransaction({'nonce': self.w3.eth.getTransactionCount(ch_ac)})
74          s_tx = self.w3.eth.account.signTransaction(tx, self.private_key)
75          tx_hash = self.w3.eth.sendRawTransaction(s_tx.rawTransaction)
76          tx_receipt = self.w3.eth.wait_for_transaction_receipt(tx_hash)
77          if DEBUG:
78              print(tx_hash)
79              print(self.w3.eth.get_transaction(tx_hash))
80
81          #  トランザクション情報を残しておく
82          self.tx = tx
83          self.s_tx = s_tx
84          self.tx_hash = tx_hash
85          self.tx_receipt = tx_receipt
86
87
88      #   イベント処理コントラクトの登録
89      def set_event_contract(self, event, addr, abi):
```

```
90          if DEBUG : print("set_contract", event, addr,abi)
91
92          self.contracts[event] = {"address":addr, "abi":abi, "contract":self.w3.eth.contract(address=self.w3.toChecksumAddress(addr), abi=abi)}
93
94
95      #   イベント処理コントラクトの実行
96      def put_event_contract(self, event, account, params):
97          if DEBUG : print("put_event_contract", event, account, params)
98          if event not in self.contracts:
99              print("no event:", event)
100             return None
101         if account == None:
102             print("use self.account:", self.account)
103             account = self.account
104         cnt = self.contracts[event]['contract']
105         ch_ac = self.w3.toChecksumAddress(self.account)
106         keys = []
107         values = []
108         for k in params:
109             keys.append(k)
110             if k == 'account':
111                 #values.append(self.w3.toChecksumAddress(params[k]))
112                 values.append(params[k].lower())
113                 account = params[k].lower()
114             else:
115                 values.append(params[k])
116         tx = cnt.functions.put(self.w3.toChecksumAddress(account), keys, values).buildTransaction({'nonce': self.w3.eth.getTransactionCount(ch_ac)})
117         s_tx = self.w3.eth.account.signTransaction(tx, self.private_key)
118         tx_hash = self.w3.eth.sendRawTransaction(s_tx.rawTransaction)
119         tx_receipt = self.w3.eth.wait_for_transaction_receipt(tx_hash)
120
121     #   イベント処理コントラクトの情報取得
122     def get_event_contract(self, event, params):
123         if DEBUG : print("get_event_contract", event, params)
124         if event not in self.contracts:
125             print("no event:", event)
126             return None
127         cnt = self.contracts[event]['contract']
128         keys = []
129         values = []
130         for k in params:
131             keys.append(k)
132             if k == 'account':
133                 #values.append(self.w3.toChecksumAddress(params[k]))
134                 values.append(params[k].lower())
135             else:
136                 values.append(params[k])
137         res = cnt.functions.get(keys, values).call()
138         if DEBUG : print("get_event_contract:res:",res)
139         return res
```

リスト 5

・put_data(channel_id, measurement, timestamp, data)

入力：

channel_id	チャネル ID 文字列
measurement	measurement 文字列
timestamp	日時
data	以下の形式のデータ

```
{
    "timestamp":＜日時＞,
    "source": ＜データソース＞,
    "data":[{"キー":値,"キー":値, ...]
}
```

出力：なし

・get_data(channel_id, measurement, start, stop)

入力：

channel_id	チャネル ID 文字列
measurement	measurement 文字列
star	データ開始インデックス
stop	データ終了インデックス

出力：

```
{
    data:データの配列
    timestamp:日時の配列
}
```

・get_data_num(channel_id, measurement)

入力：

channel_id	チャネル ID 文字列
measurement	measurement 文字列

出力：データ数(uint)

リスト 6

put(account, keys, values)

入力：

account	処理を実行するアカウント
keys	パラメータ名の配列
values	パラメータ値の配列

※keys と values は同一インデックスの値がペアで、key.value 形式のパラメータ情報となる

出力：なし

・get(keys, values)

入力：

keys	パラメータ名の配列
values	パラメータ値の配列

※keys と values は同一インデックスの値がペアで、key.value 形式のパラメータ情報となる

出力：任意の構造体データ（※辞書形式で表現可能な形式）

リスト 7