

```

1 // SPDX-License-Identifier: MIT
2 pragma solidity ^0.8.7;
3
4 contract MyDataStore {
5     struct Measurement {
6         string[] data;
7         string[] timestamp;
8     }
9
10    struct Channel {
11        string id;
12        mapping(string => Measurement) measurement;
13    }
14
15    mapping(string => Channel) channel_list;
16
17    function put_data(string calldata channel, string calldata measurement, string calldata timestamp, string calldata
data) public {
18        Channel storage ch = channel_list[channel];
19        ch.id = channel;
20        Measurement storage m = ch.measurement[measurement];
21        m.data.push(data);
22        m.timestamp.push(timestamp);
23    }
24
25    function get_data_num(string calldata channel, string calldata measurement) public view returns(uint) {
26        Measurement memory data = channel_list[channel].measurement[measurement];
27        return data.data.length;
28    }
29
30    function get_data(string calldata channel, string calldata measurement, uint start, uint stop) public view
returns(Measurement memory) {
31        Measurement memory data = channel_list[channel].measurement[measurement];
32        Measurement memory m;
33        uint data_num = data.data.length;
34        if (start > data_num) {
35            return m;
36        }
37        stop++;
38        if (stop > data_num) {
39            stop = data_num;
40        }
41        uint num = stop - start;
42        m.data = new string[](num);
43        m.timestamp = new string[](num);
44        for (uint i = 0; i < num; i++) {
45            m.data[i] = data.data[start+i];
46            m.timestamp[i] = data.timestamp[start+i];
47        }
48        return m;
49    }
50 }
51 }
52

```

```
1 #!/usr/bin/env python
2 # -*- coding: utf-8 -*-
3
4
5 import sys
6 import json
7
8 from fastapi import APIRouter
9 from pydantic import BaseModel
10 from typing import Union, List, Optional
11
12 from LIB import db
13 from LIB import ethereum
14
15 DEBUG = False # True
16
17 router = APIRouter(
18     prefix="/bchain_data",
19     tags=["bchain_data"],
20 )
21
22 # Ethereumへのアクセス情報取得
23 s = db.DB()
24 eth_rec = s.select(db.DsEthereum, "Ethereum")
25 if DEBUG : print("eth_rec=", eth_rec)
26 if eth_rec != None:
27     url = "http://" + eth_rec.address + ":" + str(eth_rec.port)
28     account = eth_rec.account
29     password = eth_rec.password
30     private_key = eth_rec.private_key
31 else:
32     url = "http://192.168.3.3:8545"
33     account = "0x70f9e0445a572d62eabf6d6669f69283558d7e2f"
34     password = "dataprovider"
35     private_key = "c2f7b10487844a8d0f708a2397a63e0aa2b2695351775fe5548557bc71280e28"
36
37
38
39 if DEBUG:
40     print("url = ", url)
41     print("account = ", account)
42     print("password = ", password)
43     print("private_key = ", private_key)
```

```

44
45
46 eth = ethereum.Ethereum(url, account, password, private_key)
47
48 # コントラクト設定
49 func_rec = s.select(db.DsFunction, "MyDataStore")
50 if DEBUG : print("func_rec = ", func_rec)
51 eth.set_contract(func_rec.address, func_rec.abi)
52
53 """
54 Ethereum
55 """
56
57 class DList(BaseModel):
58     timestamp: Optional[str]
59     source: str
60     data: List[dict]
61
62 @router.get("/")
63 async def get_data(channel_id: str, measurement: str, start: int = 0, stop: int = 100):
64     print("get_data", channel_id, measurement, start, stop)
65     """
66     データを取得する
67     """
68     res = eth.get_data(channel_id, measurement, start, stop, None)
69     if DEBUG : print(res)
70
71     return {"Status": "OK", "data":res}
72
73 @router.get("/length")
74 async def get_data_number(channel_id: str, measurement: str):
75     print("get_data_number", channel_id, measurement)
76     """
77     データ数を取得する
78     """
79     res = eth.get_data_number(channel_id, measurement)
80     if DEBUG : print(res)
81
82     return {"Status": "OK", "Result":res}
83
84
85 @router.post("/{channel_id}/{measurement}")
86 async def put_data(channel_id: str, measurement: str, data : DList):
87     print("put_data:", channel_id, measurement, data)
88     """
89     データを登録する
90     """
91     eth.put_data(channel_id, measurement, data.json())
92     return {"Status": "OK"}
93
94

```

\$ geth --networkid "504" --nodiscover --datadir `pwd`/DATA/ --http --http.addr "192.168.3.3" --http.port "8545" --http.corsdomain
"*" --http.api "eth,net,web3,personal,admin,rpc,txpool,miner" --maxpeers 0 --allow-insecure-unlock --unlock

0x70f9e0445a572d62eabf6d6669f69283558d7e2f console

INFO [08-30|15:47:22.190] Maximum peer count ETH=0 LES=0 total=0
INFO [08-30|15:47:22.398] Set global gas cap cap=50,000,000
INFO [08-30|15:47:22.406] Allocated trie memory caches clean=154.00MiB dirty=256.00MiB
INFO [08-30|15:47:22.407] Allocated cache and file handles
database=/Users/tsuchiya/CQ/MyDataServer/Ethereum/DATA/geth/chaindata cache=512.00MiB handles=5120
INFO [08-30|15:47:22.860] Opened ancient database
database=/Users/tsuchiya/CQ/MyDataServer/Ethereum/DATA/geth/chaindata/ancient readonly=false
INFO [08-30|15:47:22.909]
INFO [08-30|15:47:22.909]

INFO [08-30|15:47:22.909] Chain ID: 504 (unknown)
INFO [08-30|15:47:22.909] Consensus: unknown
INFO [08-30|15:47:22.909]
INFO [08-30|15:47:22.909] Pre-Merge hard forks:
INFO [08-30|15:47:22.909] - Homestead: 0
(<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/homestead.md>)
INFO [08-30|15:47:22.909] - Tangerine Whistle (EIP 150): 0
(<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/tangerine-whistle.md>)
INFO [08-30|15:47:22.909] - Spurious Dragon/1 (EIP 155): 0
(<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md>)
INFO [08-30|15:47:22.909] - Spurious Dragon/2 (EIP 158): 0
(<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/spurious-dragon.md>)
INFO [08-30|15:47:22.909] - Byzantium: 0
(<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/byzantium.md>)
INFO [08-30|15:47:22.909] - Constantinople: 0
(<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/constantinople.md>)
INFO [08-30|15:47:22.909] - Petersburg: 0
(<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/petersburg.md>)
INFO [08-30|15:47:22.909] - Istanbul: 0
(<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/istanbul.md>)
INFO [08-30|15:47:22.909] - Berlin: 0
(<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/berlin.md>)
INFO [08-30|15:47:22.909] - London: 0
(<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/london.md>)
INFO [08-30|15:47:22.909]
INFO [08-30|15:47:22.909] The Merge is not yet available for this network!
INFO [08-30|15:47:22.909] - Hard-fork specification:
<https://github.com/ethereum/execution-specs/blob/master/network-upgrades/mainnet-upgrades/paris.md>
INFO [08-30|15:47:22.909]

INFO [08-30|15:47:22.909]

```
INFO [08-30|15:47:22.911] Disk storage enabled for ethash caches   dir=/Users/tsuchiya/CQ/MyDataServer/Ethereum/DATA/geth/ethash
count=3
INFO [08-30|15:47:22.913] Disk storage enabled for ethash DAGS   dir=/Users/tsuchiya/Library/Ethash count=2
INFO [08-30|15:47:22.913] Initialising Ethereum protocol       network=504 dbversion=8
INFO [08-30|15:47:22.965] Loaded most recent local header       number=5230 hash=f973f0..9b90fd td=2,351,883,609 age=3d1h30m
INFO [08-30|15:47:22.965] Loaded most recent local full block   number=5230 hash=f973f0..9b90fd td=2,351,883,609 age=3d1h30m
INFO [08-30|15:47:22.965] Loaded most recent local fast block   number=5230 hash=f973f0..9b90fd td=2,351,883,609 age=3d1h30m
INFO [08-30|15:47:22.992] Loaded local transaction journal       transactions=0 dropped=0
INFO [08-30|15:47:22.996] Regenerated local transaction journal  transactions=0 accounts=0
WARN [08-30|15:47:22.998] Switch sync mode from snap sync to full sync
INFO [08-30|15:47:23.000] Gasprice oracle is ignoring threshold set threshold=2
WARN [08-30|15:47:23.011] Engine API enabled                    protocol=eth
WARN [08-30|15:47:23.011] Engine API started but chain not configured for merge yet
INFO [08-30|15:47:23.014] Starting peer-to-peer node            instance=Geth/v1.10.21-stable-67109427/darwin-amd64/go1.18.4
INFO [08-30|15:47:23.211] New local node record                  seq=1,661,440,642,833 id=876d30f9d19a3090 ip=127.0.0.1 udp=0
tcp=30303
INFO [08-30|15:47:23.211] Started P2P networking
self="enode://1fea05927172a7ddf911a0c84708742e83b464f582506e423170d2d010921f5fc78241ef5aca3a23435cb26933338d688cab39cbd1166fbf52a6
b587dfa52aba@127.0.0.1:30303?discport=0"
INFO [08-30|15:47:23.216] IPC endpoint opened                    url=/Users/tsuchiya/CQ/MyDataServer/Ethereum/DATA/geth.ipc
INFO [08-30|15:47:23.265] Loaded JWT secret file                  path=/Users/tsuchiya/CQ/MyDataServer/Ethereum/DATA/geth/jwtsecret crc32=0xc9c2d640
INFO [08-30|15:47:23.267] HTTP server started                    endpoint=192.168.3.3:8545 auth=false prefix= cors=*
vhosts=localhost
INFO [08-30|15:47:23.270] WebSocket enabled                      url=ws://127.0.0.1:8551
INFO [08-30|15:47:23.270] HTTP server started                    endpoint=127.0.0.1:8551 auth=true prefix= cors=localhost
vhosts=localhost
Unlocking account 0x70f9e0445a572d62eabf6d6669f69283558d7e2f | Attempt 1/3
Password: INFO [08-30|15:47:26.201] Mapped network port                    proto=tcp extport=30303 intport=30303 interface="UPNP
IGDV1-IP1"

Password: □★★ここでパスワードを入力★★
INFO [08-30|15:47:51.854] Unlocked account                       address=0x70f9e0445a572d62eabf6d6669f69283558d7e2f
INFO [08-30|15:47:51.983] Etherbase automatically configured     address=0x70f9e0445a572d62eabf6d6669f69283558d7e2f
welcome to the Geth JavaScript console!

instance: Geth/v1.10.21-stable-67109427/darwin-amd64/go1.18.4
coinbase: 0x70f9e0445a572d62eabf6d6669f69283558d7e2f
at block: 5230 (Sat Aug 27 2022 14:17:16 GMT+0900 (JST))
datadir: /Users/tsuchiya/CQ/MyDataServer/Ethereum/DATA
modules: admin:1.0 debug:1.0 engine:1.0 eth:1.0 ethash:1.0 miner:1.0 net:1.0 personal:1.0 rpc:1.0 txpool:1.0 web3:1.0

To exit, press ctrl-d or type exit
>
```

リスト3

```
(mydata_server) % cd LIB
(mydata_server) % ./setfunc.py --id MyDataStore --name "data store contract" --address
'0x61031415293772674cab0435e8981c3e8c7e800' --abi '[
    {
        "inputs": [
```

~~ 省略 ~~

```
        "stateMutability": "view",
        "type": "function"
    }
]
```

リスト4

```
mysql> select id,address from ds_function where id = 'MyDataStore';
```

id	address
MyDataStore	0x61031415293772674cab0435e8981c3e8c7e800

1 row in set (0.00 sec)

```
mysql>
mysql> select abi from ds_function where id = 'MyDataStore';
```

abi
[{ "inputs": [{ "internalType": "string", "name": "channel", "type": "string" }] }]

~~ 省略 ~~

```
        "stateMutability": "view",
        "type": "function"
    }
]
```

1 row in set (0.00 sec)

mysql>

リスト5

```
-----  
{  
  "timestamp": "20220910113015",  
  "source": "test",  
  "data": [  
    {"temperature":30.0,"pressure":1000.1, "humidity":40.2}  
  ]  
}
```

リスト6

```
-----  
{  
  "timestamp": "20220910113115",  
  "source": "test",  
  "data": [  
    {"temperature":31.5,"pressure":1000.1, "humidity":39.2}  
  ]  
}
```

リスト7

```
-----  
{  
  "status": "OK",  
  "data": [  
    [  
      "{ \"timestamp\": \"20220910113015\", \"source\": \"test\", \"data\": [{ \"temperature\": 30.0, \"pressure\": 1000.1, \"humidity\": 40.2} ] }",  
      "{ \"timestamp\": \"20220910113115\", \"source\": \"test\", \"data\": [{ \"temperature\": 31.5, \"pressure\": 1000.1, \"humidity\": 39.2} ] }"  
    ],  
    [  
      "20220910113015",  
      "20220910113115"  
    ]  
  ]  
}
```

```
}
```

リスト8

```
{  
  "status": "OK",  
  "data": [  
    [{"timestamp": "20220910113115", "source": "test", "data": [{"temperature": 31.5, "pressure": 1000.1,  
"humidity": 39.2}]}]  
  ],  
  ["20220910113115"  
]  
]  
}
```

リスト9