

```
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 influxdb
14
15 DEBUG = False # True
16
17 router = APIRouter(
18     prefix="/data",
19     tags=["data"],
20 )
21
22 # InfluxDB へのアクセス情報取得
23 s = db.DB()
24 idb_rec = s.select(db.DsInfluxDB, "InfluxDB")
25 if DEBUG : print("idb_rec=",idb_rec)
26 if idb_rec != None:
27     url = "http://" + idb_rec.address + ":" + str(idb_rec.port)
28     organization = idb_rec.organization
29     token = idb_rec.token
30 else:
31     url = "http://localhost:8086"
32     organization = "MyDataServer"
33     token = "jG99-jU4YqMZixUN3IGoBCdQ7iAbekCXPkaM2SEvHXt0IXEQKMDkJppaiXYn66bnsOuaEDZ323k1p3EYIHw-zg=="
34
35 if DEBUG:
36     print("url = ", url)
37     print("organization = ", organization)
38     print("token = ", token)
39
40 idb = influxdb.InfluxDB(url, organization, token)
```

```

41
42
43 """
44 InfluxDB
45 """
46 class DList(BaseModel):
47     timestamp: Optional[str]
48     source: str
49     data: List[dict]
50
51 @router.get("/")
52 async def get_data(channel_id: str, measurement: Union[str, None] = None, start: Union[str, None] = None, stop: Union[str, None] =
None):
53     print("get_data", channel_id, measurement, start, stop)
54     """
55     データを取得する
56     """
57     res = idb.get_data(channel_id, measurement, start, stop, None)
58     if DEBUG: print(res)
59
60     return {"Status": "OK", "data": res}
61
62
63
64 @router.post("/{channel_id}/{measurement}")
65 async def put_data(channel_id: str, measurement: str, data : DList):
66     print("put_data:", channel_id, measurement, data)
67     """
68     データを登録する
69     """
70     idb.put_data(channel_id, measurement, data.json())
71     return {"Status": "OK"}
72
73

```