

```

1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4
5  import sys
6  import json
7
8  from fastapi import APIRouter, File
9  from fastapi.responses import Response
10 from pydantic import BaseModel
11 from typing import Union, List, Optional
12
13 from LIB import db
14 from LIB import storage
15
16 DEBUG = False # True
17
18 router = APIRouter(
19     prefix="/file",
20     tags=["file"],
21 )
22
23 # MinIO へのアクセス情報取得
24 s = db.DB()
25 st_rec = s.select(db.DsStorage, "MinIO")
26 if DEBUG : print("st_rec=",st_rec)
27 if st_rec != None:
28     url = "http://" + st_rec.address + ":" + str(st_rec.port)
29     access_key_id = st_rec.access_key_id
30     secret_access_key = st_rec.secret_access_key
31 else:
32     url = "http://localhost:9000"
33     access_key_id = "admin"
34     secret_access_key = "admin001"
35
36 if DEBUG:
37     print("url = ", url)
38     print("access_key_id = ", access_key_id)
39     print("secret_access_key = ", secret_access_key)
40
41 st = storage.Storage(url, access_key_id, secret_access_key)
42
43 """
44 Storage
45 """
46 @router.get("/{channel_id}")
47 async def list_file(channel_id: str):
48     print("read_file", channel_id)
49     """
50     データを検索
51     """
52     res = st.list_file(channel_id)
53
54     return {"Status": "OK", "OBJECTS":res}
55
56 @router.get("/{channel_id}/{filename}")
57 async def get_file(channel_id: str, filename: str, need_token: bool = False, account: str = None):
58     print("get_file", channel_id, filename, need_token, account)
59     """

```

```
60     データを検索
61     """
62     d = st.get_file(channel_id, filename)
63
64     res = Response(content=d, status_code=200)
65
66     return res
67
68 @router.post("/{channel_id}/{filename}")
69 async def put_file(channel_id: str, filename: str, file: bytes = File(...)):
70     print("put_file", channel_id, filename)
71
72     res = st.put_file(channel_id, filename, file)
73
74     return {"Status": "OK", "RESULT": res}
75
76
77 @router.delete("/{channel_id}/{filename}")
78 async def delete_file(channel_id: str, filename: str):
79     print("delete_file", channel_id, filename)
80
81     res = st.delete_file(channel_id, filename)
82
83     return {"Status": "OK", "RESULT": res}
```