

```
1  #!/usr/bin/env python
2  # -*- coding: utf-8 -*-
3
4
5  import sys
6  import json
7  import datetime
8  import uvicorn
9
10 from fastapi import FastAPI
11 from fastapi import Request
12 from fastapi.responses import JSONResponse
13 from fastapi.security import OAuth2PasswordBearer, OAuth2PasswordRequestForm
14 from fastapi.middleware.cors import CORSMiddleware
15 from pydantic import BaseModel
16 from fastapi.staticfiles import StaticFiles
17 from routers import channel_rt, data_rt, file_rt, bchain_data_rt, nft_rt
18 from APP import event
19
20
21 IP="0.0.0.0"
22 PORT=8000
23
24
25 # 初期化
26 app = FastAPI()
27
28 # イベント処理ミドルウェア
29 evt = event.Event()
30 @app.middleware("http")
31 async def process_event(request: Request, call_next):
32     print("START process_event", request)
33
34     # 処理前イベント
35     res = evt.pre_event(request)
36
37     # NFT 対応の特殊処理
38     if 'MyPhotoNFT' in res:
39         r = res['MyPhotoNFT']
40         print(r)
```

```
41
42     if r['status'] != 0:
43         print("NO call_next")
44         return JSONResponse(content=r['data'], status_code=403)
45
46     # 処理実態
47     response = await call_next(request)
48
49     # 処理後イベント
50     res = evt.post_event(request, response)
51
52     print("END process_event", response)
53
54     return response
55
56
57 # CORS 対応
58 origins = [
59     "http://localhost",
60     "http://localhost:"+str(PORT),
61 ]
62 app.add_middleware(
63     CORSMiddleware,
64     allow_origins=origins,
65     allow_credentials=True,
66     allow_methods=["*"],
67     allow_headers=["*"],
68 )
69
70
71 # API 処理)の登録
72 app.include_router(data_rt.router)
73 app.include_router(channel_rt.router)
74 app.include_router(bchain_data_rt.router)
75 app.include_router(file_rt.router)
76 app.include_router(nft_rt.router)
77
78
79 # システム情報
80 class SystemInfo(BaseModel):
```

```
81     System = "My Data Server"
82     Version = "1.0b"
83     Release = "2022/08/31"
84
85 @app.get("/info", response_model=SystemInfo)
86 def read_root():
87     return SystemInfo()
88
89
90 # 画面表示用の静的ファイル
91 app.mount("/dataserver", StaticFiles(directory="html"), name="html")
92
93
94 @app.get("/")
95 async def read_root():
96     return {"My Data Server"}
97
98 if __name__ == "__main__":
99     uvicorn.run(app, host=IP, port=PORT)
```