1. **uvicorn으로 돌릴 index.py**

from fastapi import FastAPI, Form, Request

from fastapi.responses import HTMLResponse

#HTML을 templates폴더에서 꺼내 사용하기위해 Jinja2Templates 사용

from fastapi.templating import Jinja2Templates

app = FastAPI()

#HTML들어있는 폴더 정의해줌

templates = Jinja2Templates(directory="templates")

@app.get("/login/")

#request보내줘야해서 req: Request 인자로 넣어준다.

async def login(req: Request):

return templates.TemplateResponse("login.html", {"request": req})

# form형식에서 post방식으로 데이터를 보낼 것이다. 이떄 get과 post의 다른점은 post가 더욱 보안이 강화되어 있고, 대용량을 보낼때 좋다고 한다.

#그래서 post방식으로 전달하고 받지 않으면 not allowed에러가 나온다.

@app.post("/login/loginProc/")

async def loginProc(req:Request, id: str = Form(...), pw: str=Form(...)):

#return {'username': id, 'password':pw}

#id를 출력할 것이기 때문에 req먼저 보낸 후, id를 보내준다.

return templates.TemplateResponse("loginProc.html", {"request": req, "id": id})

@app.get('/')

async def main():

content = """

<html>

<head>

<title>Main</title>

</head>

<body>

<a href="/login"><h1> 로그인</h1></a>

<h1>로그아웃</h1>

</body>

</html>

"""

return HTMLResponse(content=content)

1. **위 @app.get(“/login/”)에서 사용할 login.html**

<html>

<head>

<title>Login</title>

</head>

<body>

<h1> Login!</h1>

<form action = '/login/loginProc' method='post'>

<input type='text' name='id' value='Id입력'><br/>

<input type='text' name='pw' value='pw입력'><br/>

<input type='submit' value='Login'>

</form>

</body>

</html>

1. **@app.post("/login/loginProc/") 에서 사용할 loginProc.html**

<html>

<body>

<h1> ID : {{id}}로 로그인</h1>

</body>

</html>