네트워크 과제

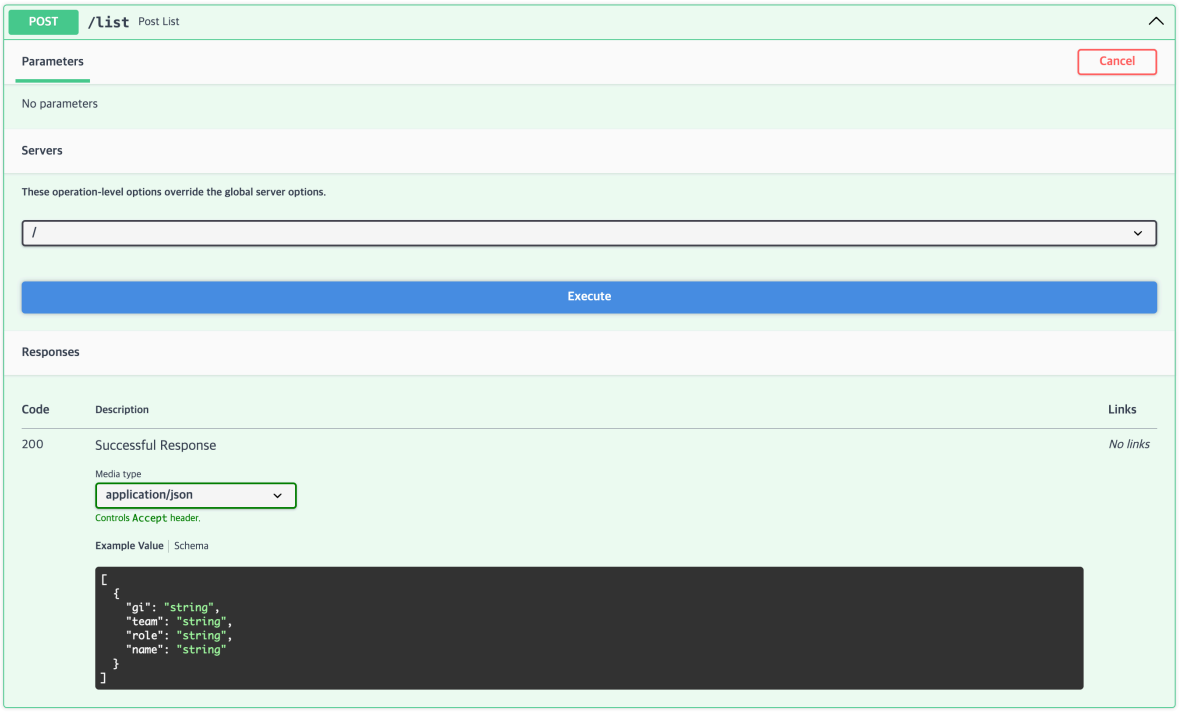
백준호

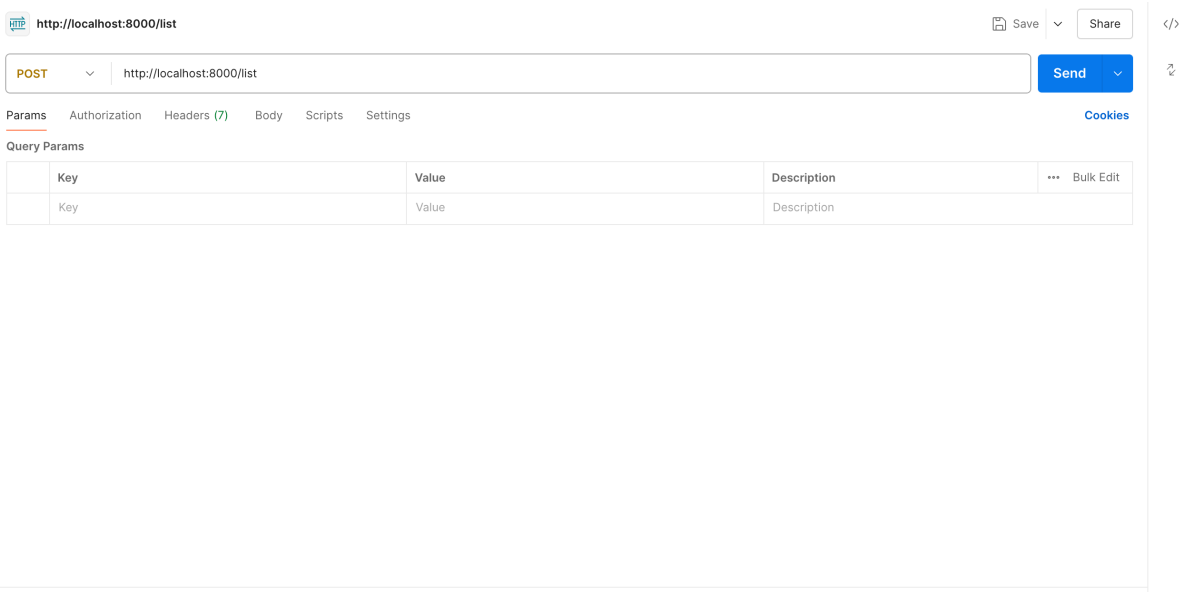
FastAPI 설정. 인코딩 문제를 해결하기 위해 Respose 한국어 인코딩 방식에 통용되는 라이브러리도 가져왔음.

```python
from fastapi import FastAPI, HTTPException
from fastapi.responses import UJSONResponse
from pydantic import BaseModel
from typing import List, Dict

app = FastAPI(default_response_class=UJSONResponse)

class Data(BaseModel):
    gi: str
    team: str
    role: str
    name: str


data_store: List[Data] = []
```

과제 1 (POST): /list 라는 라우터에 포스트 요청을 하면 student_list가 respose_model에 저장됨. /list에 get 요청시 해당 데이터를 불러오기 위해서 data_store에도 저장함. Form 형식의 html을 채워서 submit 해서 데이터를 저장하는 방식이 웹에서 일반적이지만, 과제내용을 수행할 때 postman이나 swaggerUI를 사용해서 post 요청을 보낼 수 있기에 데이터를 다 적어둠.

```python
@app.post("/list", response_model=List[Data])
def post_list():
    student_list = [
        Data(gi='23기', team='DE팀', role='전 부대장', name='이어흥'),
        Data(gi='24기', team='DE팀', role='팀장', name='임채림'),
        Data(gi='22기', team='DS팀', role='교육부장', name='김지훈'),
        Data(gi='24기', team='DE팀', role='부회장', name='조윤영'),
        Data(gi='24기', team='DS팀', role='회장', name='이동진')
    ]
    data_store.extend(student_list)
    return student_list
```

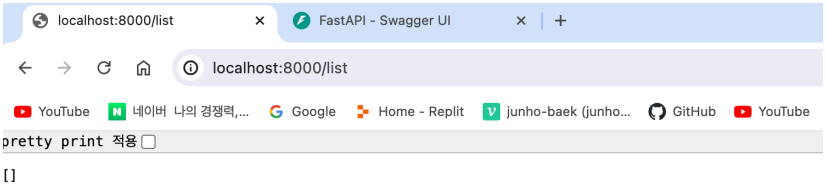




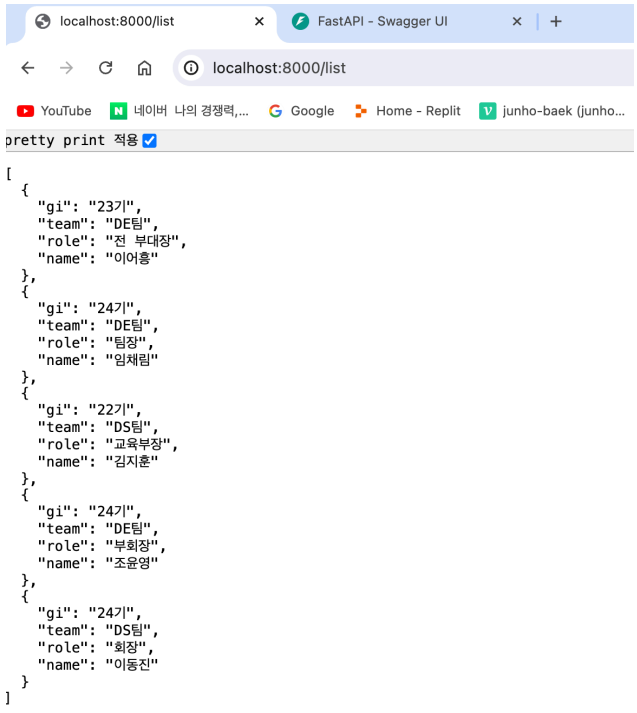
둘 중 아무거나 사용해서 post 요청을 보냄. 여러번 누르면 데이터가 중복해서 저장됨.

과제 2 (GET): data_store를 respose로 보냄.

```python
@app.get("/list", response_model=List[Data])
def get_list():
    return data_store
```



Post 이전 get



Post 이후 get

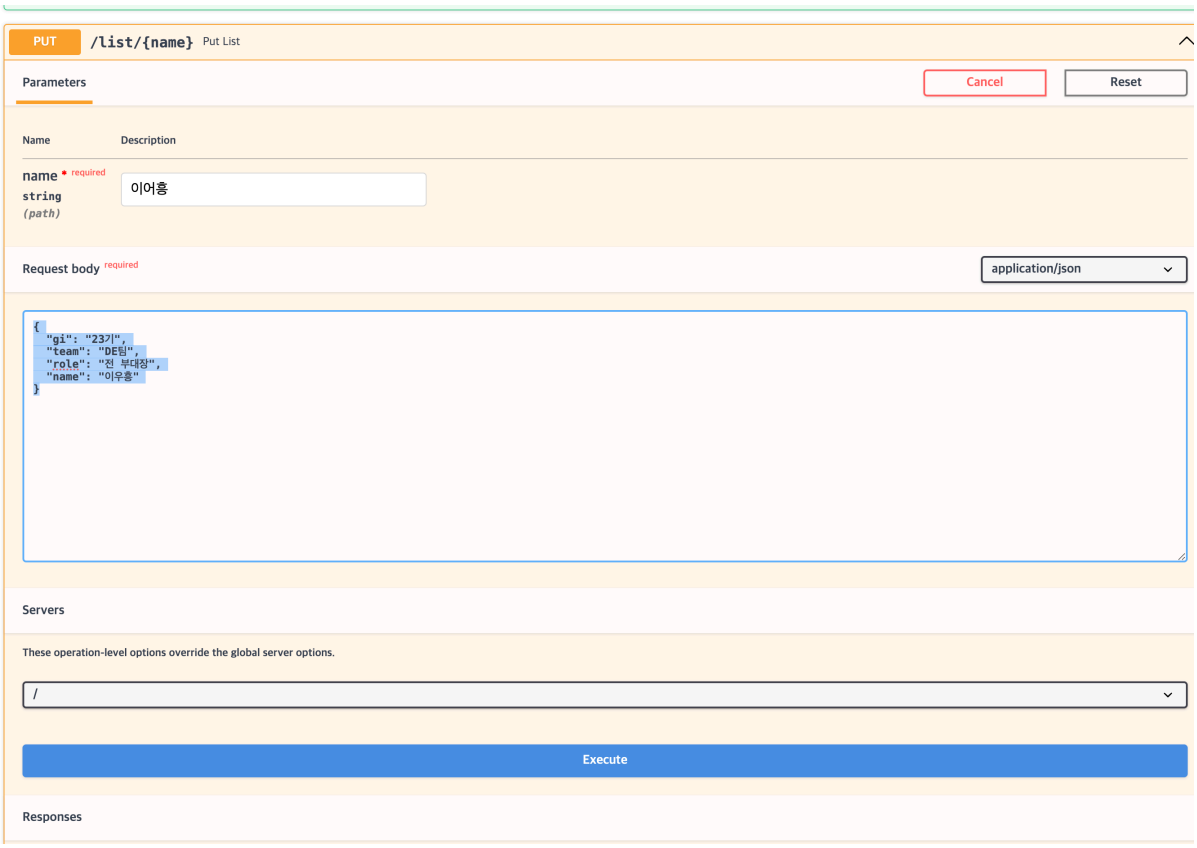
과제 3 (PUT): put 요청시 name을 넣고 데이터에 해당 name 프로퍼티가 같은 값을 가지고 있는 데이터를 새로 받은 데이터로 수정하는 코드.

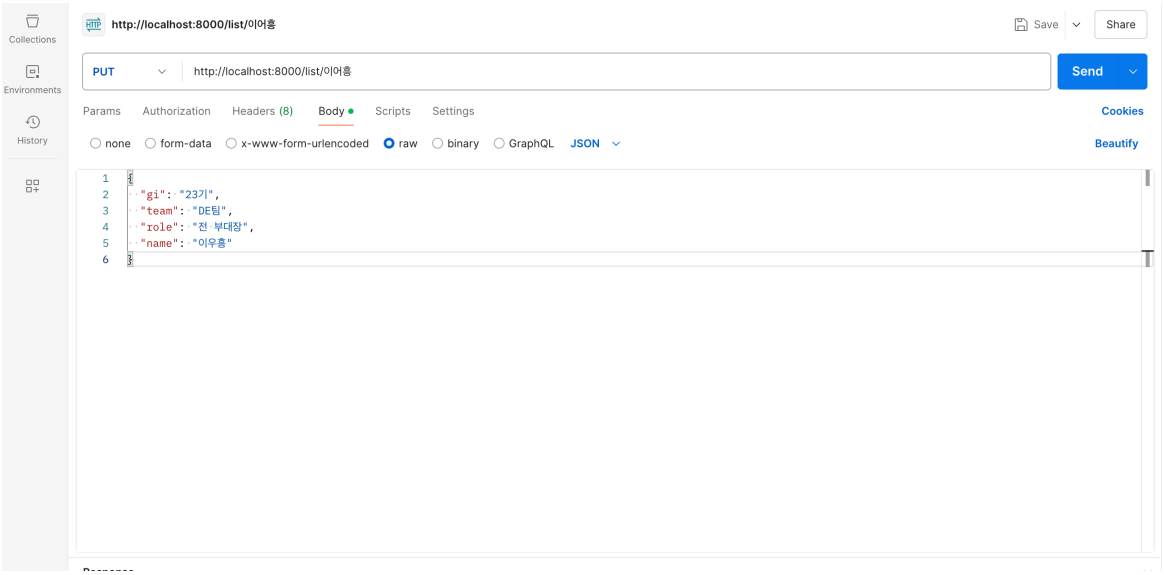
이어흥, {

```
    "gi": "23기",
    "team": "DE팀",
    "role": "전 부대장",
    "name": "이우흥"
}
```

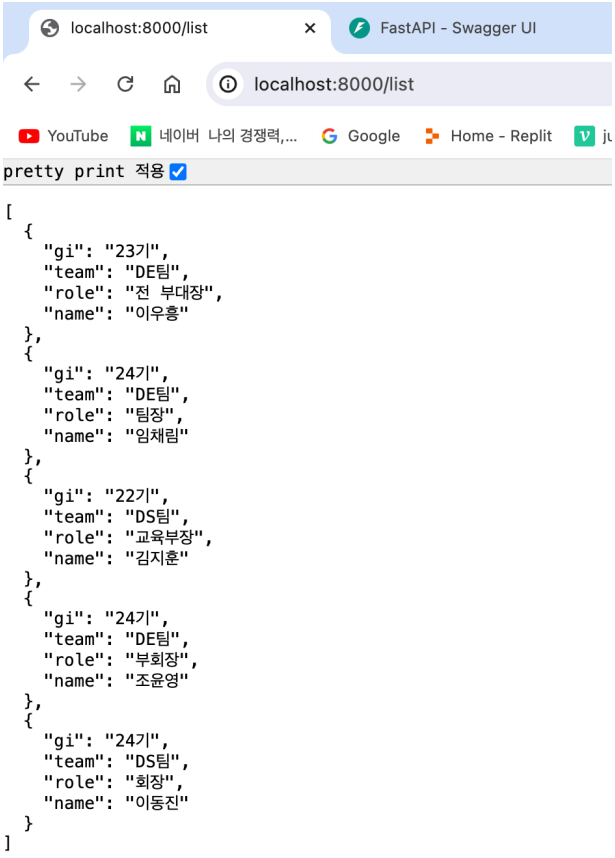
을 데이터로 넣어 수정

```python
@app.put("/list/{name}", response_model=Data)
def put_list(name: str, data: Data):
    for i, d in enumerate(data_store):
        if d.name == name:
            data_store[i] = data
            return data
    raise HTTPException(status_code=404, detail="Item not found")
```





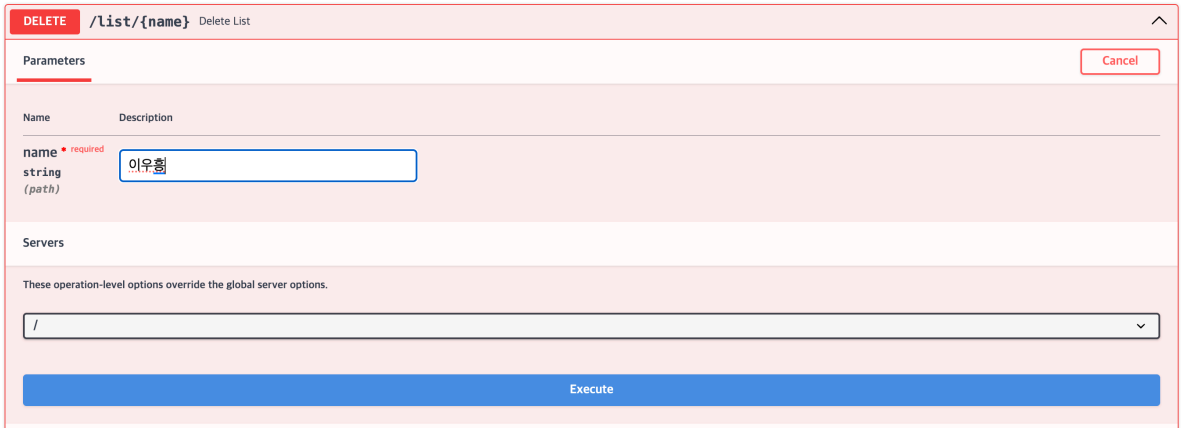
둘 중 아무거나 사용해서 put 요청.

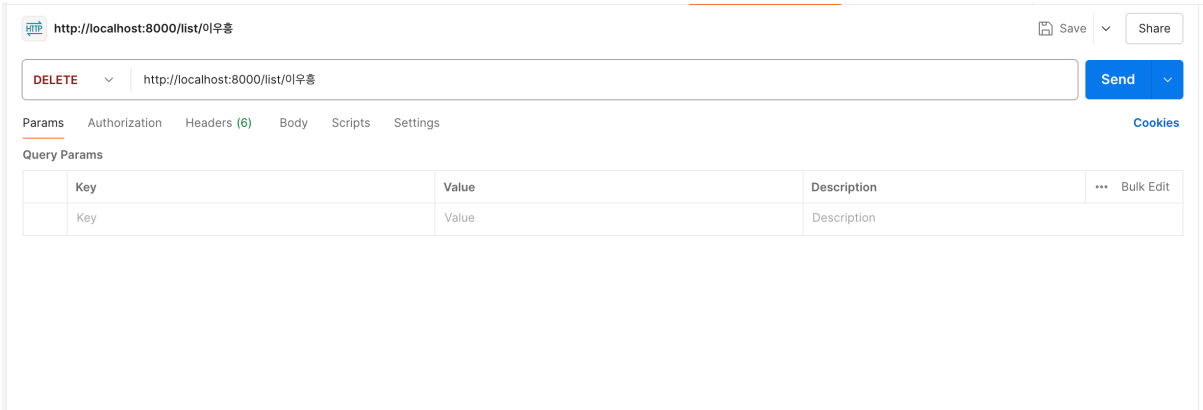


Get 으로 확인

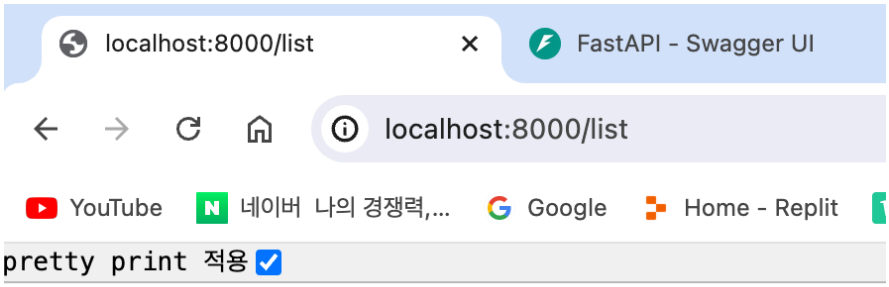
과제 4 (DELETE): name parameter를 받고 put 메서드와 같이 데이터를 탐색 후 데이터를 삭제

```python
@app.delete("/list/{name}", response_model=Data)
def delete_list(name: str):
    for i, d in enumerate(data_store):
        if d.name == name:
            data_store.pop(i)
            return d
    raise HTTPException(status_code=404, detail="Item not found")
```





둘 중 아무 방식으로 delete 수행



get으로 확인

pretty print 적용


```
[
  {
    "gi": "24기",
    "team": "DE팀",
    "role": "팀장",
    "name": "임채림"
  },
  {
    "gi": "22기",
    "team": "DS팀",
    "role": "교육부장",
    "name": "김지훈"
  },
  {
    "gi": "24기",
    "team": "DE팀",
    "role": "부회장",
    "name": "조윤영"
  },
  {
    "gi": "24기",
    "team": "DS팀",
    "role": "회장",
    "name": "이동진"
  }
]
```