

1.

```
Request body required

{
  "gi": "24기",
  "team": "DS팀",
  "role": "회장",
  "name": "이동진"
}
```

2.

```
Response body

{
  "data": {
    "이어흥": {
      "gi": "23기",
      "team": "DE팀",
      "role": "전 부대장",
      "name": "이어흥"
    },
    "임채림": {
      "gi": "24기",
      "team": "DE팀",
      "role": "DE 팀장",
      "name": "임채림"
    },
    "김재훈": {
      "gi": "22기",
      "team": "DS팀",
      "role": "교육부장",
      "name": "김재훈"
    },
    "조윤영": {
      "gi": "24기",
      "team": "DE팀",
      "role": "부회장",
      "name": "조윤영"
    },
    "이동진": {
      "gi": "24기",
      "team": "DS팀",
      "role": "회장",
      "name": "이동진"
    }
  }
}
```

3.

Server response	
Code	Details
200	<div>Response body<pre>{ "status": "success", "updated_data": { "gi": "string", "team": "string", "role": "string", "name": "이우흥" }}</pre></div>

4.

Name	Description
name <small>★ required</small> string (path)	<input type="text" value="이우흥"/>

Servers

These operation-level options override the global server options.

Execute

Responses

Curl

```
curl -X 'DELETE' \
  'http://localhost:8000/list/%EC%9D%B4%EC%9A%B0%ED%9D%A5' \
  -H 'accept: application/json'
```

Request URL

```
http://localhost:8000/list/%EC%9D%B4%EC%9A%B0%ED%9D%A5
```

코드 설명

```
# 귀찮으니깐 한번에...ㅎㅎ
@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
```

데이터를 입력받아서 넣을 수 도 있지만 처음부터 넣을 수 있고 매칭만 시켜주면 된다

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

get 데이터 조회

전체를 조회하는 get 이고 data_store로 반환 값을 줘서 내역을 다 보여준다

```
@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")
```

put 데이터 수정

enumerate 함수를 사용해서 d.name이 {name} 과 일치하는지를 검사하고 일치하는 경우 해당 인덱스 i에 새 데이터를 할당!

```

@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")

```

del 데이터 삭제

put 과 비슷하게 del 도 순회하면서 d.name 이 name 과 같다면 pop해서 데이터 없앤다

```

"""
@app.post("/list", response_model=Data)
def post_list(data: Data):
    data_store.append(data)
    return data
"""

```

post 데이터 입력

입력받은 데이터를 data_store.append(data) 로 추가한다