

1.

New Collection / New Request

GETlocalhost:8000/docs

ParamsAuthorizationHeaders (7)BodyScriptsSettings

Query Params

Key	Value	Description
Key	Value	Description

BodyCookiesHeaders (4)Test Results

Status: 200 OKTime: 10 msSize: 104 K

PrettyRawPreviewVisualizeHTML

1<!DOCTYPE html>

2<html>

3<head>

4<link type="text/css" rel="stylesheet" href="https://cdn.jsdelivr.net/npm/swagger-ui-dist@5/swagger-ui.css">

5<link rel="shortcut icon" href="https://fastapi.tiangolo.com/img/favicon.png">

6<title>FastAPI - Swagger UI</title>

7</head>

8<body>

9<div id="swagger-ui">

10</div>

11<script src="https://cdn.jsdelivr.net/npm/swagger-ui-dist@5/swagger-ui-bundle.js"></script>

12<!-- 'SwaggerUIBundle' is now available on the page -->

13<script>

14const ui = SwaggerUIBundle({

15 url: '/openapi.json',

16 "dom_id": "#swagger-ui",

17 "layout": "BaseLayout",

18 "deepLinking": true,

19 "showExtensions": true,

20

2.

New Collection / New Request

POSThttp://127.0.0.1:8000/listSend

ParamsAuthorizationHeaders (9)BodyScriptsSettings

BodyCookiesHeaders (4)Test Results

Status: 200 OKTime: 26 msSize: 468 BSave as example

PrettyRawPreviewVisualizeJSON

```
1 {
2   {
3     "g1": "23기",
4     "team": "08팀",
5     "role": "전 부대장",
6     "name": "이이훈"
7   },
8   {
9     "g1": "24기",
10    "team": "08팀",
11    "role": "팀장",
12    "name": "김재원"
13  },
14  {
15    "g1": "22기",
16    "team": "09팀",
17    "role": "교목부장",
18    "name": "김지훈"
19  },
20  {
21    "g1": "24기",
22    "team": "08팀",
23    "role": "부회장",
24    "name": "조승영"
25  },
26  {
27    "g1": "24기",
28    "team": "09팀",
29    "role": "회장",
30    "name": "이동진"
31  }
32 }
```

3.

New Collection / New Request

PUT

http://127.0.0.1:8000/list/이여홍

Send

Params

Authorization

Headers (9)

Body

Scripts

Settings

none

form-data

x-www-form-urlencoded

raw

binary

GraphQL

JSON

Beautify

```
1 {
2   "id": "23기",
3   "team": "06팀",
4   "role": "전 부대장",
5   "name": "이우홍"
6 }
```

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 8 ms

Size: 196 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": "23기",
3   "team": "06팀",
4   "role": "전 부대장",
5   "name": "이우홍"
6 }
```

4.

New Collection / New Request

DELETE

http://127.0.0.1:8000/list/이우훈

Send

Params

Authorization

Headers (7)

Body

Scripts

Settings

Query Params

Key	Value	Description	Bulk Edit
Key	Value	Description	

Body

Cookies

Headers (4)

Test Results

Status: 200 OK

Time: 7 ms

Size: 196 B

Save as example

Pretty

Raw

Preview

Visualize

JSON

```
1 {
2   "id": "23기",
3   "team": "DE팀",
4   "role": "전 부대장",
5   "name": "이우훈"
6 }
```

5. 코드 내용

```
from fastapi import FastAPI, HTTPException
```

```
from pydantic import BaseModel
```

```
from typing import List, Dict
```

```
app = FastAPI()
```

```
class Data(BaseModel):
```

```
    gi: str
```

```
    team: str
```

```
    role: str
```

```
    name: str
```

```
# 전역 데이터 저장소
```

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

```
# POST 코드
```

```
@app.post("/list", response_model=List[Data])
```

```
def post_list():
```

```
    student_list = [
```

```
        Data(gi='23기', team='DE팀', role='전 부대장', name='이어홍'),
```

```
        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
```

```
# GET 코드
```

```
@app.get("/list", response_model=List[Data])
```

```
def get_list():
```

```
    return data_store
```

```
# PUT 코드
```

```
@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")
```

```
# DELETE 코드
```

```
@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")
```

```
if __name__ == '__main__':
```

```
    import uvicorn
```

```
    uvicorn.run(app, host="localhost", port=8000, log_level="info")
```