

2025년 상반기 K-디지털 트레이닝

axios를 이용한 HTTP 통신

[KB] IT's Your Life

✓ axios-test-app 프로젝트를 생성하세요.

- 관련 패키지 설치
- axios 패키지 설치

✓ 프로젝트 만들기

- `npm init axios-test-app`
- `npm install`
- `npm install axios`

- ✔ 프로젝트 루트에 db.json을 다음과 같이 정의하세요.

```
{
  "todos": [
    { "id": "1", "todo": "야구장", "desc": "프로야구 경기도 봐야합니다.", "done": false },
    { "id": "2", "todo": "놀이", "desc": "노는 것도 중요합니다.", "done": false },
    { "id": "3", "todo": "Vue 학습", "desc": "Vue 학습을 해야 합니다", "done": false },
    { "id": "4", "todo": "ES6 공부", "desc": "ES6공부를 해야 합니다", "done": false }
  ]
}
```

- ✔ db.json을 db로 하여 json-server를 rest 서버로 실행하세요.
 - 웹 브라우저로 json-server에 접속함
 - todos 엔드 포인트를 확인하세요.

✓ npx json-server db.json

○ http://localhost:3000

○ http://localhost:3000/todos

JSON Server

[Docs](#) [♡ Sponsor](#)

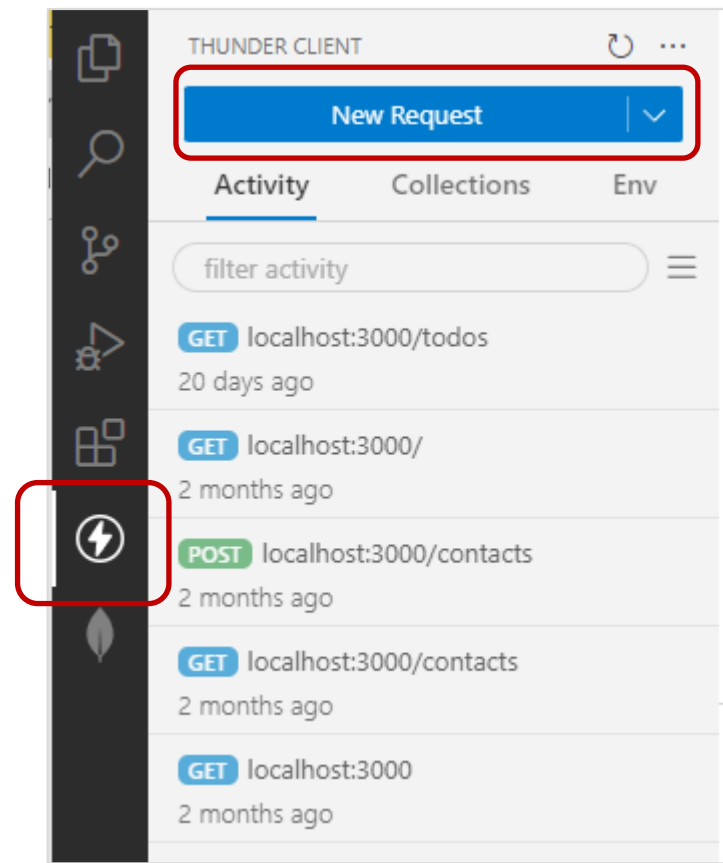
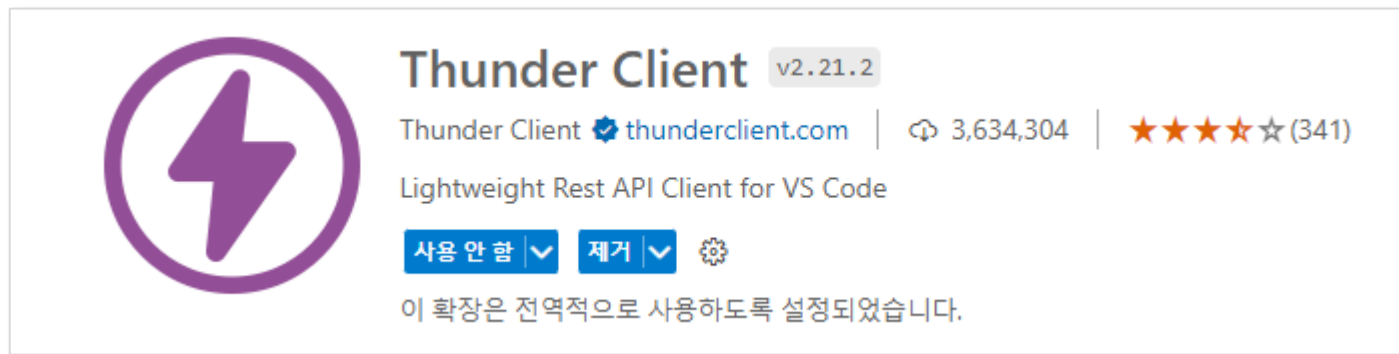
✧*.9(0`*)9✧*.

/todos - 4 items

```
[
  {
    "id": "1",
    "todo": "ES6학습",
    "desc": "설명1",
    "done": false
  },
  {
    "id": "2",
    "todo": "React학습",
    "desc": "설명2",
    "done": false
  },
  {
    "id": "3",
    "todo": "ContextAPI 학습",
    "desc": "설명3",
    "done": true
  },
  {
    "id": "4",
    "todo": "야구경기 관람",
    "desc": "설명4",
    "done": false
  }
]
```

- ✓ VS Code 확장팩 Thunder Client를 설치하세요
- ✓ Thunder Client를 이용해서 todo rest 서버를 대상으로 CRUD 테스트를 진행하세요.

✓ VS Code 확장팩 Thunder Client



✓ GET 요청 - 목록보기

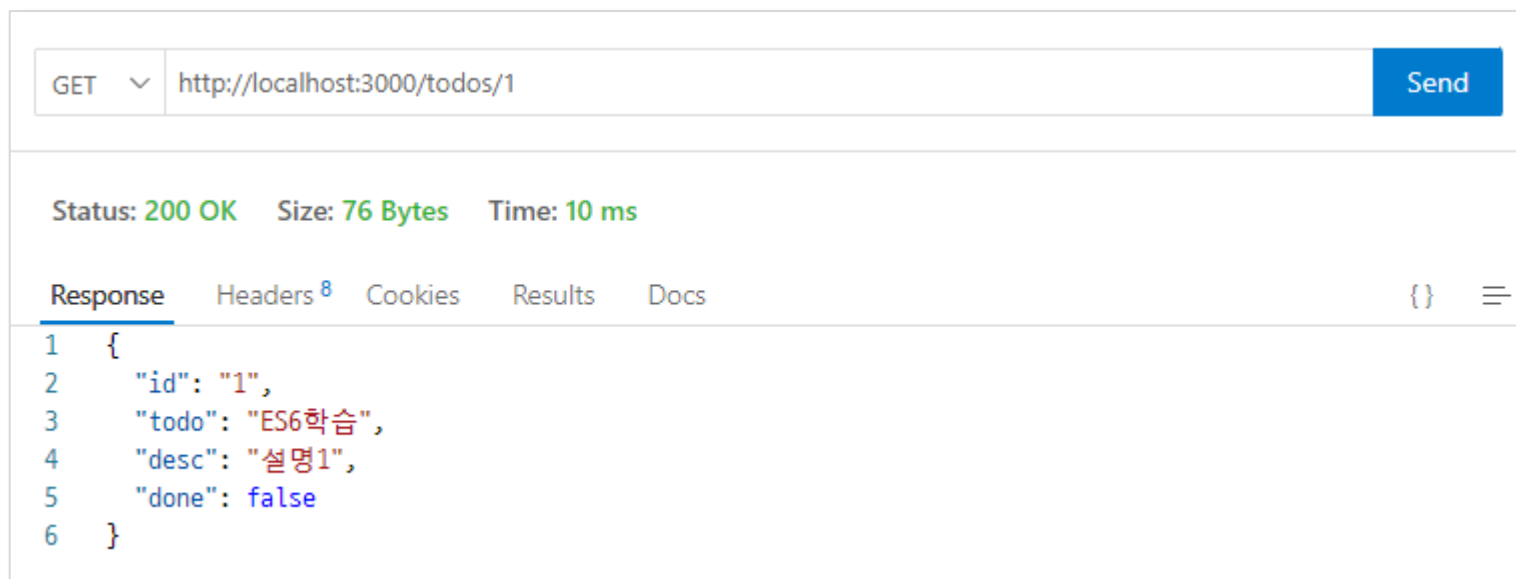
GET ⌵ http://localhost:3000/todos Send

Status: 200 OK Size: 381 Bytes Time: 13 ms

Response Headers ⁸ Cookies Results Docs

```
1  [  
2    {  
3      "id": "1",  
4      "todo": "ES6학습",  
5      "desc": "설명1",  
6      "done": false  
7    },  
8    {  
9      "id": "2",  
10     "todo": "React학습",  
11     "desc": "설명2",  
12     "done": false  
13   },  
14   {  
15     "id": "3",  
16     "todo": "ContextAPI 학습",  
17     "desc": "설명3",  
18     "done": true  
19   },  
20   {  
21     "id": "4",  
22     "todo": "야구경기 관람",  
23     "desc": "설명4",  
24     "done": false  
25   }  
26 ]
```

✓ GET 요청 - 상세보기



✓ POST 요청 - 생성

POST ▼ http://localhost:3000/todos Send

Query Headers² Auth Body¹ Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "todo": "axios 패키지",
3   "desc": "axios rest api 실습",
4   "done": false
5 }
```

Status: 201 Created Size: 99 Bytes Time: 19 ms

Response Headers⁸ Cookies Results Docs {} ≡

```
1 {
2   "id": "87a6",
3   "todo": "axios 패키지",
4   "desc": "axios rest api 실습",
5   "done": false
6 }
```

✓ PUT 요청 - 수정

PUT Send

Query Headers² Auth **Body¹** Tests Pre Run

JSON XML Text Form Form-encode GraphQL Binary

JSON Content Format

```
1 {
2   "id": "87a6",
3   "todo": "axios 패키지",
4   "desc": "axios rest api 실습",
5   "done": true
6 }
```

Status: 200 OK Size: 98 Bytes Time: 16 ms

Response Headers⁸ Cookies Results Docs {} ≡

```
1 {
2   "id": "87a6",
3   "todo": "axios 패키지",
4   "desc": "axios rest api 실습",
5   "done": true
6 }
```

✓ DELETE 요청 - 삭제

The screenshot displays a REST client interface with the following details:

- Method:** DELETE
- URL:** http://localhost:3000/todos/87a6
- Body:** JSON Content: 1
- Status:** 200 OK
- Size:** 98 Bytes
- Time:** 24 ms
- Response:**

```
1 {
2   "id": "87a6",
3   "todo": "axios 패키지",
4   "desc": "axios rest api 실습",
5   "done": true
6 }
```