

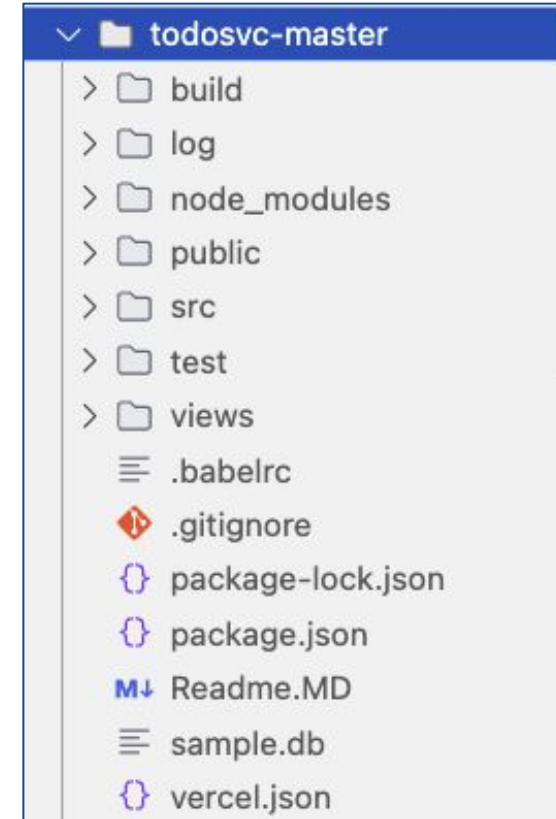
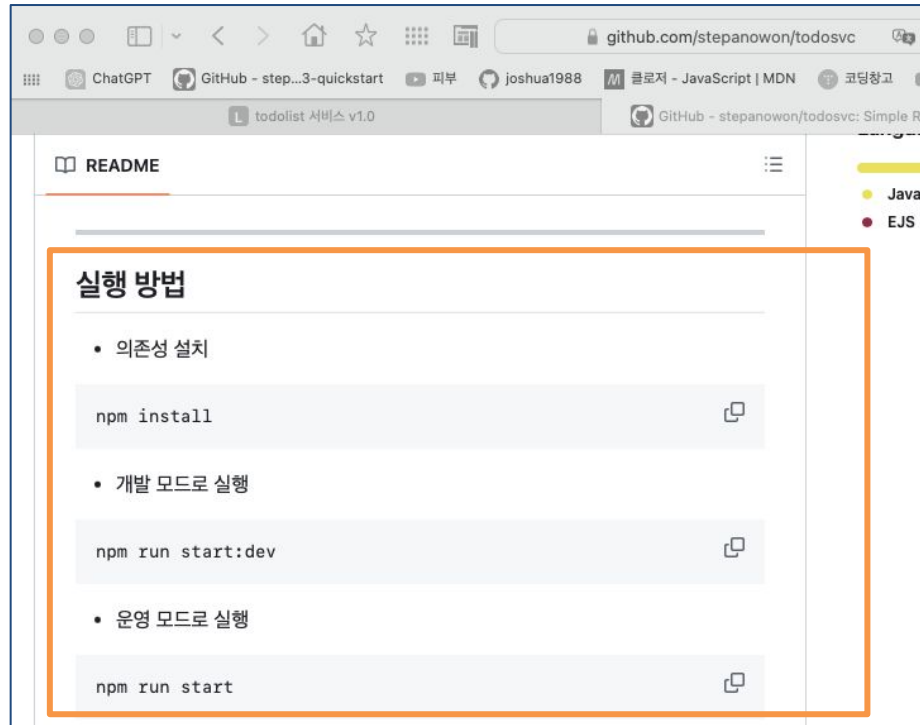
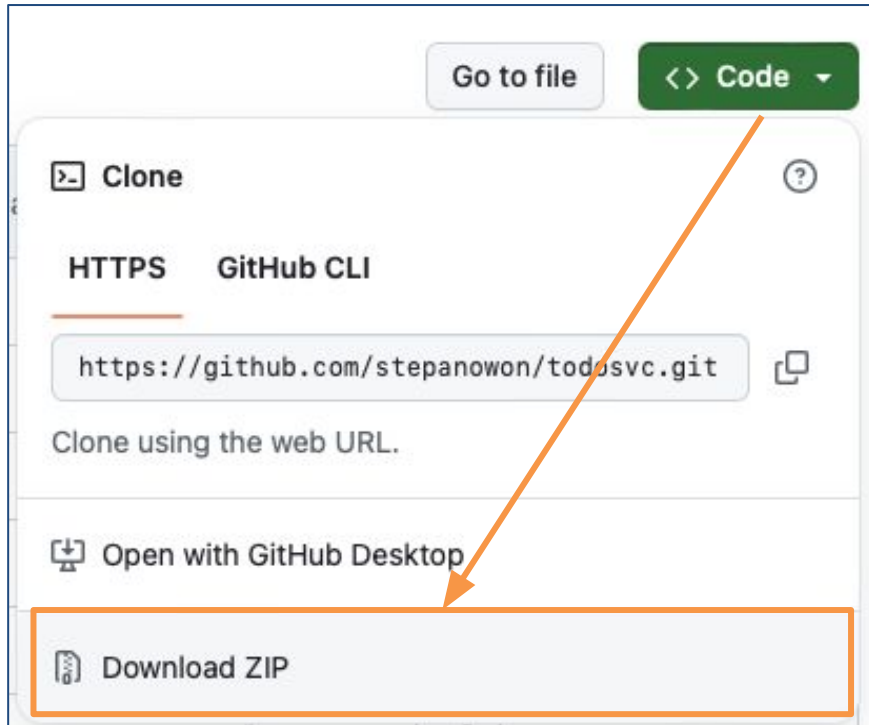
2024년 상반기 K-디지털 트레이닝

# VueJS

---

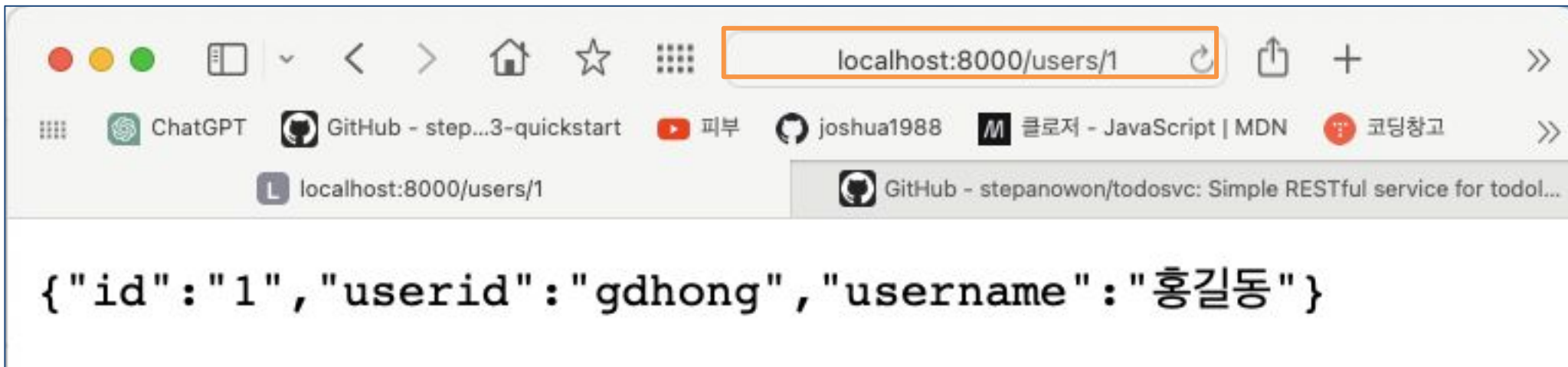
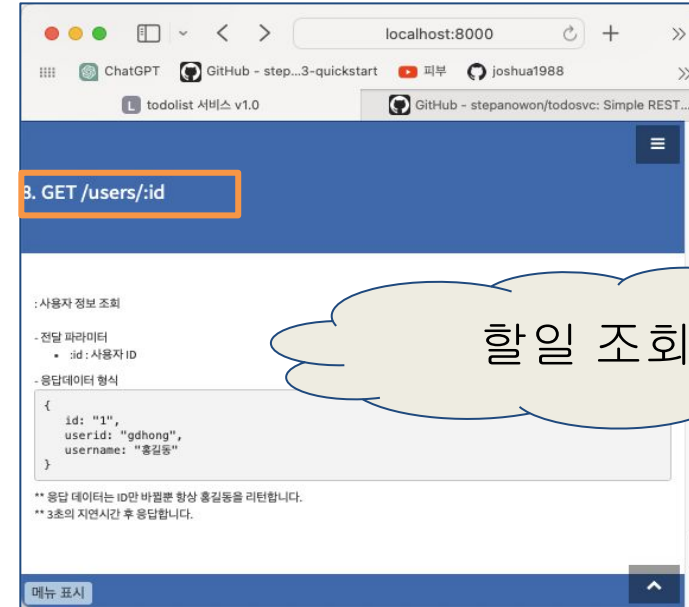
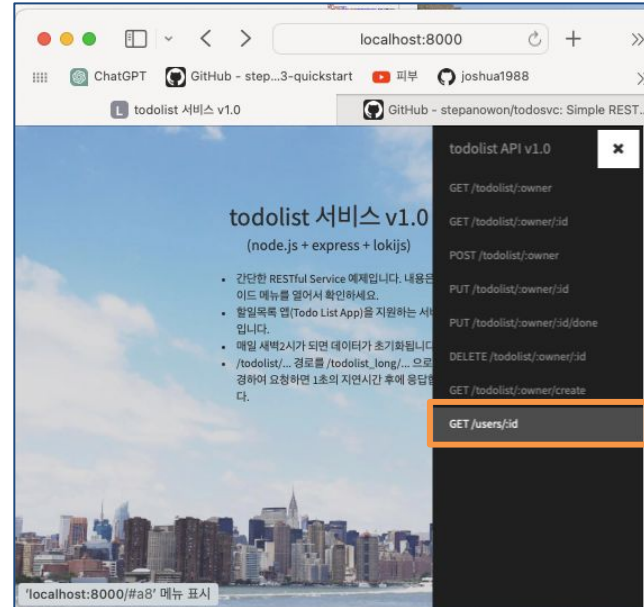
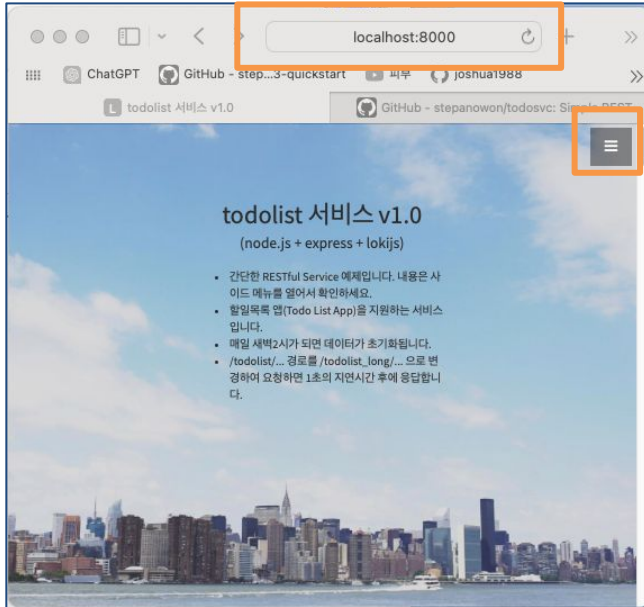
[KB] IT's Your Life

- mock server만들기([github에서 다운로드](#))



```
● administrator@MacBook-Pro todosvc-master % npm install
```

```
○ administrator@MacBook-Pro todosvc-master % npm run start
```

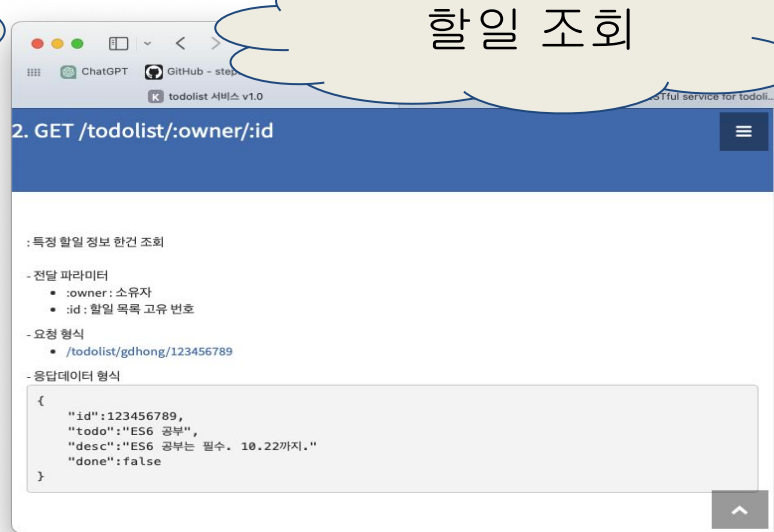


## ● http://localhost:8000의 주요 API

### 할일목록 조회



### 할일 조회



### 할일 추가



### 할일 수정



### 할일 완료 수정



### 할일 삭제



콘솔을 확인합니다.

Loaded new content contentScript.js:6

# 응답객체 : App5.vue:14

▼ {data: Array(4), status: 200, statusText: 'OK', headers: AxiosHeaders, config: {...}, ...} i

▼ config:

- ▶ adapter: (2) ['xhr', 'http']
- ▶ data: undefined
- ▶ env: {FormData: f, Blob: f}
- ▶ headers: AxiosHeaders {Accept: 'application/json, text/plain, \*/\*', Content-Type: null}
- ▶ maxBodyLength: -1
- ▶ maxContentLength: -1
- ▶ method: "get"
- ▶ timeout: 2000
- ▶ transformRequest: [f]
- ▶ transformResponse: [f]
- ▶ transitional: {silentJSONParsing: true, forcedJSONParsing: true, clarifyTimeoutError: fa
- ▶ url: "/api/todolist\_long/gdhong"
- ▶ validateStatus: f validateStatus(status)
- ▶ xsrfCookieName: "XSRF-TOKEN"
- ▶ xsrfHeaderName: "X-XSRF-TOKEN"
- ▶ [[Prototype]]: Object

▼ data: Array(4)

- ▶ 0: {id: 123456789, todo: 'ES6 공부', desc: 'ES6공부를 해야 합니다', done: true}
- ▶ 1: {id: 1716018870861, todo: 'Vue 학습', desc: 'Vue 학습을 해야 합니다', done: false}
- ▶ 2: {id: 1716018870862, todo: '놀이', desc: '노는 것도 중요합니다.', done: true}
- ▶ 3: {id: 1716018870863, todo: '야구장', desc: '프로야구 경기도 봐야합니다.', done: false}

length: 4



- 다음 프로젝트를 vite기반으로 생성하고, 컴포넌트를 설계하시오.
- 프로젝트 명 : mock-app



The screenshot shows a web browser window with the address bar displaying 'localhost:8000/todolist/gdhong'. The page content is a JSON array of three todo items. The first item is completed, while the other two are not.

```
1 // 20240416112431
2 // http://localhost:8000/todolist/gdhong
3
4 [
5   {
6     "id": 123456789,
7     "todo": "ES6 공부",
8     "desc": "ES6공부를 해야 합니다",
9     "done": true
10  },
11  {
12    "id": 1713233337974,
13    "todo": "Vue 학습",
14    "desc": "Vue 학습을 해야 합니다",
15    "done": false
16  },
17  {
18    "id": 1713233337975,
19    "todo": "놀이",
20    "desc": "노는 것도 중요합니다.",
21    "done": true
22  },
23  {
24    "id": 1713233337976,
25    "todo": "야구장",
26    "desc": "프로야구 경기도 봐야합니다.",
27    "done": false
28  }
29 ]
```

- 다음은 App.vue에서 gdhong에 대한 정보를 검색하는 화면이다. 오류를 해결하시오.

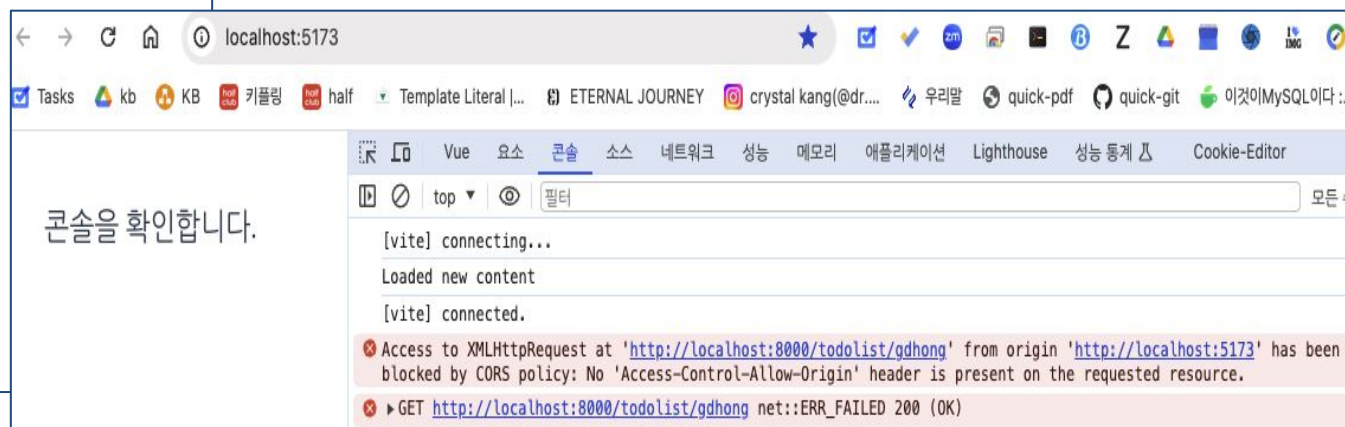
```
<template>
  <div>
    <h2>콘솔을 확인합니다.</h2>
  </div>
</template>

<script setup>
import axios from 'axios'

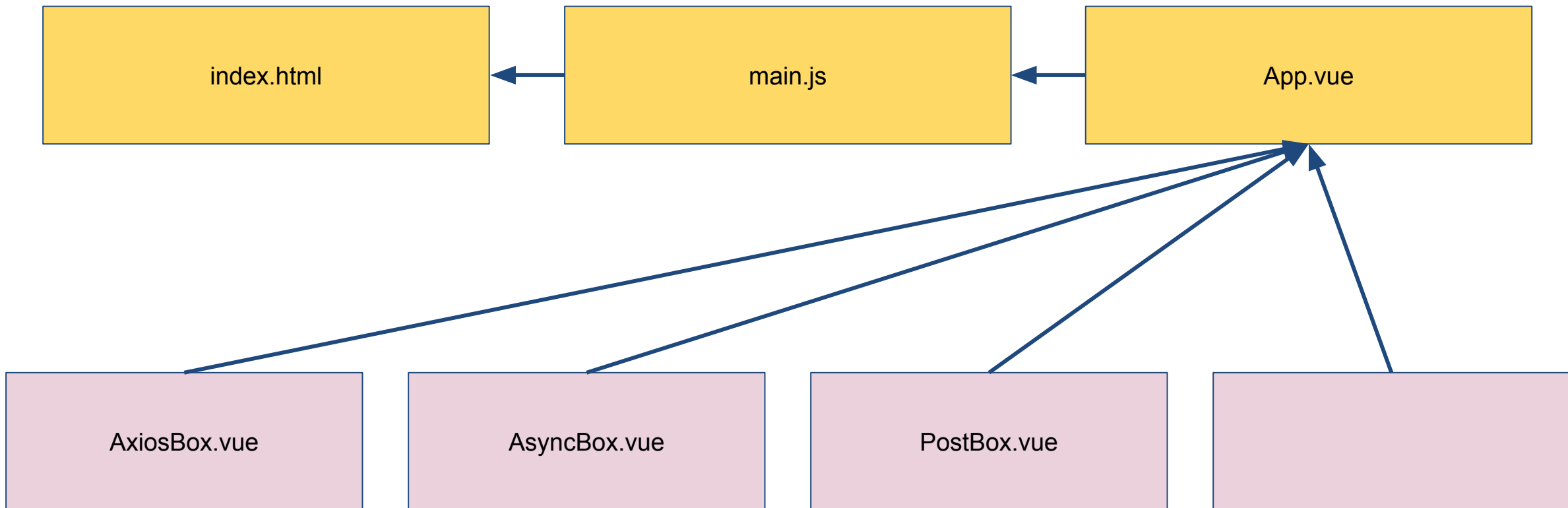
const requestAPI = () => {
  const url = "http://localhost:8000/todolist/gdhong";
  //const url = "/api/todolist/gdhong";
  axios.get(url).then((response) => {
    console.log("# 응답객체 : ", response);
  });
};

requestAPI();
</script>
```

App.vue



- 다음 컴포넌트로 각각 **axios**로 **CRUD**를 처리하시오.





- 전체검색, 첫번째 Todo, 두번째 Todo를 검색하시오. (Promise)

AxiosBox.vue

```
[vite] connecting...
[vite] connected.
Loaded new content
# TodoList : ▼ (4) [{...}, {...}, {...}, {...}] i
  ▶ 0: {id: 123456789, todo: 'ES6 공부', desc: 'ES6공부를 해야 합니다', done: true}
  ▶ 1: {id: 1713240846776, todo: 'Vue 학습', desc: 'Vue 학습을 해야 합니다', done: false}
  ▶ 2: {id: 1713240846777, todo: '놀이', desc: '노는 것도 중요합니다.', done: true}
  ▶ 3: {id: 1713240846778, todo: '야구장', desc: '프로야구 경기도 봐야합니다.', done: false}
  length: 4
  ▶ [[Prototype]]: Array(0)
▶ XHR 로드 완료됨: GET "http://localhost:5174/api/todolist long/gdhong".
## 첫번째 Todo : ▼ {id: 123456789, todo: 'ES6 공부', desc: 'ES6공부를 해야 합니다', done: true} i
  desc: "ES6공부를 해야 합니다"
  done: true
  id: 123456789
  todo: "ES6 공부"
  ▶ [[Prototype]]: Object
▶ XHR 로드 완료됨: GET "http://localhost:5174/api/todolist long/gdhong/123456789".
▶ XHR 로드 완료됨: GET "http://localhost:5174/api/todolist long/gdhong/1713240846776".
## 두번째 Todo : ▼ {id: 1713240846776, todo: 'Vue 학습', desc: 'Vue 학습을 해야 합니다', done: false} i
  desc: "Vue 학습을 해야 합니다"
  done: false
  id: 1713240846776
  todo: "Vue 학습"
  ▶ [[Prototype]]: Object
```

- gdhong01 작성한 글을 검색하시오.(async~await, try~catch처리)

AsyncBox.vue

```
# 응답객체 :  
▼ Object i  
  ▶ config: {transitional: {...}, adapter: Array(2), transformRequest: Array(1), transformResponse: Array(1), timeout: 0, headers: {Accept: 'application/json, text/plain, */*', 'Accept-Encoding': 'gzip, deflate, br', 'Cache-Control': 'no-cache', 'Content-Type': 'application/json', 'User-Agent': 'axios/0.18.0'}, method: 'get', url: 'https://api.github.com/search/issues?q=ES6+공부', withCredentials: false}  
  ▼ data: Array(6)  
    ▶ 0: {id: 123456789, todo: 'ES6 공부', desc: 'ES6공부를 해야 합니다', done: true}  
    ▶ 1: {id: 1713240846776, todo: 'Vue 학습', desc: 'Vue 학습을 해야 합니다', done: false}  
    ▶ 2: {id: 1713240846777, todo: '놀이', desc: '노는 것도 중요합니다.', done: true}  
    ▶ 3: {id: 1713240846778, todo: '야구장', desc: '프로야구 경기도 봐야합니다.', done: false}  
    ▶ 4: {id: 1713245457312, todo: '윗몸일으키기 3세트', desc: '너무 빠르지 않게...', done: false}  
    ▶ 5: {id: 1713245458593, todo: '윗몸일으키기 3세트', desc: '너무 빠르지 않게...', done: false}  
    length: 6  
    ▶ [[Prototype]]: Array(0)  
  ▶ headers: AxiosHeaders {access-control-allow-origin: '*', cache-control: 'private, no-cache', content-type: 'application/json', user-agent: 'axios/0.18.0'}  
  ▶ request: XMLHttpRequest {onreadystatechange: null, readyState: 4, timeout: 0, withCredentials: false, status: 200, statusText: "OK"}  
  ▶ [[Prototype]]: Object
```

- data키에 있는 데이터를 post방식으로 전송하시오.(async~await, try~catch처리)

PostBox.vue

```
▼ {status: 'success', message: '추가 성공', item: {...}} i
  ▼ item:
    desc: "너무 빠르지 않게..."
    id: 1713245841685
    todo: "윗몸일으키기 3세트"
    ► [[Prototype]]: Object
    message: "추가 성공"
    status: "success"
    ► [[Prototype]]: Object
  ► XHR 로드 완료됨: POST "http://localhost:5174/api/todolist\_long/gdhong".
```

```
let data = { todo: "윗몸일으키기 3세트", desc: "너무 빠르지 않게..." };
```

TodoList App Home About TodoList

Home

TodoList App Home About TodoList

할일 추가

ES6학습	삭제	편집
React학습	삭제	편집
ContextAPI 학습 (완료)	삭제	편집
야구경기 관람	삭제	편집

TodoList App Home About TodoList

할일 추가

할일 :

설명 :

추가 취소

TodoList App Home About TodoList

할일 수정

할일:

공부하지마

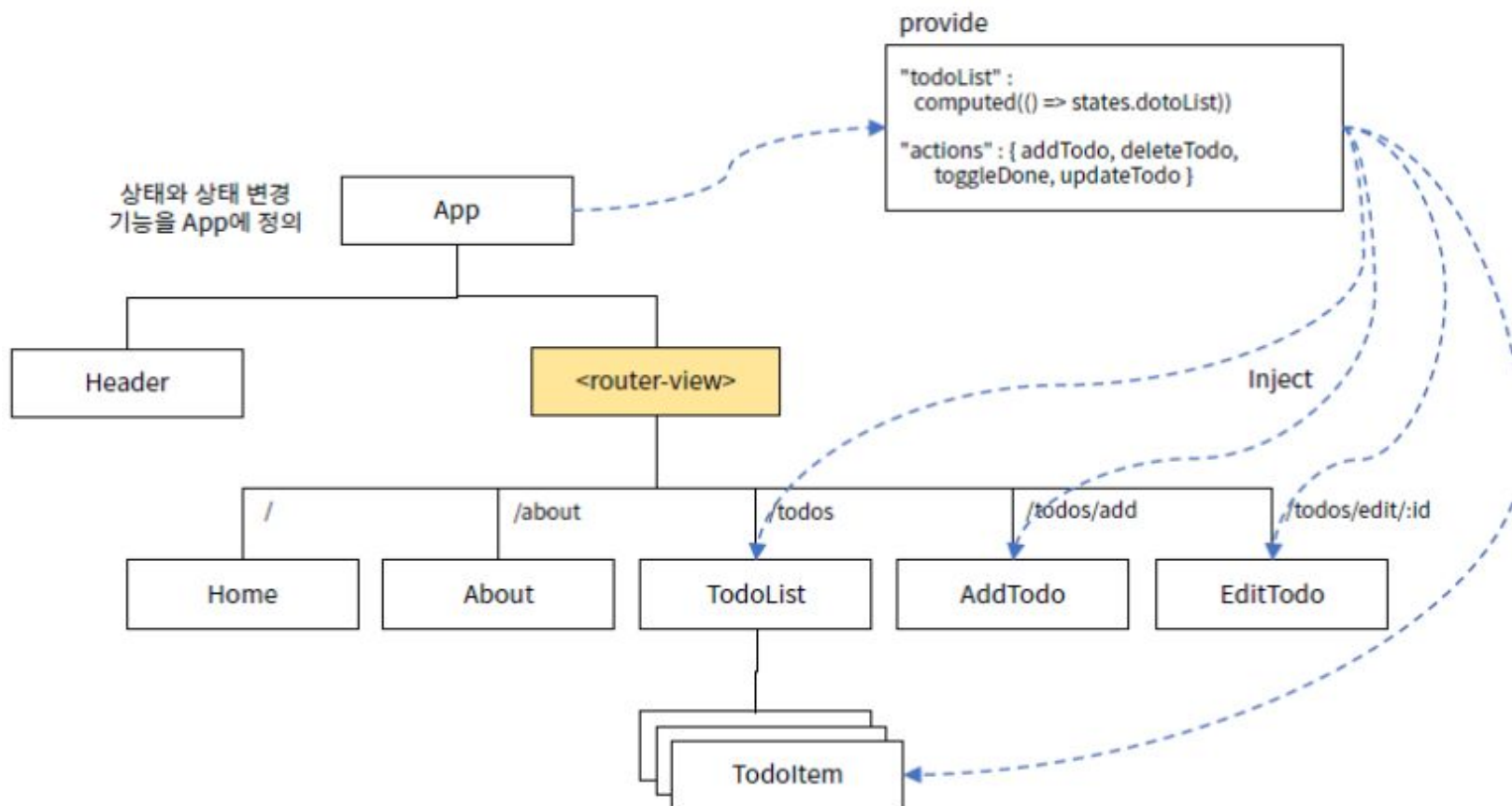
설명:

설명1

완료여부 : ☒

수정 취소

- vite project를 생성하시오.
- 프로젝트 명: **todolist-app**
- mock server를 사용하시오.



- spinnerUI를 적용하시오.

