

2025년 상반기 K-디지털 트레이닝

axios를 이용한 HTTP 통신(심화1)

[KB] IT's Your Life

- ✅ axios-test-app 프로젝트에서 json-server를 프록시 서버로 설정하세요.

vite.config.js 변경

```
import { fileURLToPath, URL } from 'node:url'
import { defineConfig } from 'vite'
import vue from '@vitejs/plugin-vue'

export default defineConfig({
  plugins: [vue()],
  resolve: {
    alias: {
      '@': fileURLToPath(new URL('./src', import.meta.url))
    }
  },
  server: {
    proxy: {
      '/api': {
        target: 'http://localhost:3000',
        changeOrigin: true,
        rewrite: (path) => path.replace(/^\/api/, ''),
      },
    },
  },
})
```

- 최초 요청 경로: /api/todolist/1
- 타겟: http://localhost:3000
- 최종 전달 경로: http://localhost:3000/todolist/1

- ✓ App.vue를 axios를 이용해 /api/todos/1를 요청하고, 그 결과를 콘솔에 출력하세요.

src/App.vue

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

<script setup>
import axios from 'axios'

const requestAPI = () => {
  // const url = "http://localhost:3000/todos/1";
  const url = "/api/todos/1";
  axios.get(url).then((response) => {
    console.log("# 응답객체 : ", response);
  });
};

requestAPI();
</script>
```

```
#응답 객체 : App.vue:6
{data: {...}, status: 200, statusText: 'OK', headers: AxiosHeader
s, config: {...}, ...}
  ▼ config:
    ▶ adapter: (2) ['xhr', 'http']
    ▶ data: undefined
    ▶ env: {FormData: f, Blob: f}
    ▶ headers: AxiosHeaders {Accept: 'application/json, text/plai
maxBodyLength: -1
maxContentLength: -1
method: "get"
timeout: 0
▶ transformRequest: [f]
▶ transformResponse: [f]
▶ transitional: {silentJSONParsing: true, forcedJSONParsing:
url: "/api/todos/1"
▶ validateStatus: f validateStatus(status,
xsrCookieName: "XSRF-TOKEN"
xsrHeaderName: "X-XSRF-TOKEN"
▶ [[Prototype]]: Object
▶ data: {id: '1', todo: '야구장', desc: '프로야구 경기도 봐야합
▶ headers: AxiosHeaders {access-control-allow-headers: 'content
▶ request: XMLHttpRequest {onreadystatechange: null, readyState
status: 200
statusText: "OK"
▶ [[Prototype]]: Object
```

- ✓ axios를 이용해서 todo 목록을 얻어 콘솔에 출력하고, 이어서 id가 1인 todo를 얻어 그 결과를 콘솔에 출력하도록 App.vue를 수정하세요.
 - Promise의 then/catch를 이용해서 작성함

src/App.vue

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

<script setup>
import axios from 'axios';

const listUrl = "/api/todos";
const todoUrlPrefix = "/api/todos/";

//4건의 목록을 조회한 후 첫번째, 두번째 할일을 순차적으로 조회합니다.
const requestAPI = () => {
  let todoList = [];
  axios
    .get(listUrl)
```

h2>src/App.vue

```

.then((response) => {
  todoList = response.data;
  console.log("# TodoList : ", todoList);
  return todoList[0].id;
})
.then((id) => {
  return axios.get(todoUrlPrefix + id);
})
.then((response) => {
  console.log("## 첫번째 Todo : ", response.data);
  return todoList[1].id;
})
.then((id) => {
  axios.get(todoUrlPrefix + id).then((response) => {
    console.log("## 두번째 Todo : ", response.data);
  });
});
});

requestAPI();
</script>

```

```

# TodoList : App.vue:14
▼ (4) [{...}, {...}, {...}, {...}]
  ▶ 0: {id: '1', todo: '야구장', desc: '프로야구 경기도 봐야합니다',
  ▶ 1: {id: '2', todo: '놀이', desc: '노는 것도 중요합니다.', done:
  ▶ 2: {id: '3', todo: 'Vue 학습', desc: 'Vue 학습을 해야 합니다',
  ▶ 3: {id: '4', todo: 'ES6 공부', desc: 'ES6공부를 해야 합니다',
    length: 4
  ▶ [[Prototype]]: Array(0)

## 첫번째 Todo : App.vue:21
{id: '1', todo: '야구장', desc: '프로야구 경기도 봐야합니다.', d
  one: false}
  desc: "프로야구 경기도 봐야합니다."
  done: false
  id: "1"
  todo: "야구장"
  ▶ [[Prototype]]: Object

## 두번째 Todo : App.vue:26
{id: '2', todo: '놀이', desc: '노는 것도 중요합니다.', done: fal
  se}
  desc: "노는 것도 중요합니다."
  done: false
  id: "2"
  todo: "놀이"
  ▶ [[Prototype]]: Object

```


2025년 상반기 K-디지털 트레이닝

axios를 이용한 HTTP 통신(심화2)

[KB] IT's Your Life

- ✅ **async/await axios를 이용해서 todo 목록을 얻어 출력하도록 App2.vue를 작성하고 확인하세요.**

src/App2.vue

```
...

//전체 목록을 조회한 후 한 건씩 순차적으로 순회하며 조회하기
const requestAPI = async () => {
  let todoList;

  let response = await axios.get(listUrl);
  todoList = response.data;
  console.log("# TodoList :", todoList);
  for (let i = 0; i < todoList.length; i++) {
    response = await axios.get(todoUrlPrefix + todoList[i].id);
    console.log(`# ${i + 1}번째 Todo : `, response.data);
  }
};

requestAPI();
</script>
```

```
# TodoList : App.vue:43
▼ (4) [{...}, {...}, {...}, {...}]
  ▶ 0: {id: '1', todo: '야구장', desc: '프로야구 경기도 봐야합니다', done: false}
  ▶ 1: {id: '2', todo: '놀이', desc: '노는 것도 중요합니다.', done: false}
  ▶ 2: {id: '3', todo: 'Vue 학습', desc: 'Vue 학습을 해야 합니다', done: false}
  ▶ 3: {id: '4', todo: 'ES6 공부', desc: 'ES6공부를 해야 합니다', done: false}
  length: 4
  ▶ [[Prototype]]: Array(0)

# 1번째 Todo : App.vue:46
  ▶ {id: '1', todo: '야구장', desc: '프로야구 경기도 봐야합니다.', done: false}

# 2번째 Todo : App.vue:46
  ▶ {id: '2', todo: '놀이', desc: '노는 것도 중요합니다.', done: false}

# 3번째 Todo : App.vue:46
  ▶ {id: '3', todo: 'Vue 학습', desc: 'Vue 학습을 해야 합니다', done: false}

# 4번째 Todo : App.vue:46
  ▶ {id: '4', todo: 'ES6 공부', desc: 'ES6공부를 해야 합니다', done: false}
```

- ✓ 다음 데이터를 추가하도록 App4.vue를 작성하고, 확인하세요.

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

src/App4.vue 추가

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

```
<script setup>
import axios from "axios";

const requestAPI = async () => {
  const url = "/api/todos";
  let data = { todo: "윗몸일으키기 3세트", desc: "너무 빠르지 않게..." };
  const resp1 = await axios.post(url, data);
  console.log(resp1.data);
};
requestAPI();
</script>
```

```
App.vue:60
{id: '009d', todo: '윗몸일으키기 3세트', desc: '너무 빠르지 않
게...'}
  desc: "너무 빠르지 않게..."
  id: "009d"
  todo: "윗몸일으키기 3세트"
  ▶ [[Prototype]]: Object
```

- ✓ 다음 url을 axios로 요청 했을 때 에러가 발생함. 에러 처리 코드를 가지는 App6.vue를 작성하세요.

```
const url = "/api/todos2";
```

src/App6.vue

...

```
<script setup>
import axios from "axios";

const requestAPI = async () => {
  const url = "/api/todos2";
  axios
    .get(url, { timeout: 900 })
    .then((response) => {
      console.log("# 응답객체 : ", response);
    })
    .catch((e) => {
      console.log('에러=====');
      console.log(e);
      if (e instanceof Error) console.log(e.message);
      else console.log(e);
    });
};

requestAPI();
</script>
```

