axios와 todo 앱

**학습목표**

axios로 todo 앱 구현

**목차**

[1. todo2 프로젝트 2](#_Toc151556507)

[1) axios 요약 2](#_Toc151556508)

[2) 프로젝트 생성 2](#_Toc151556509)

[3) src/todos-db.json 2](#_Toc151556510)

[4) MyTodo.vue 생성 3](#_Toc151556511)

[5) MyTodo.vue - 추가 기능 5](#_Toc151556512)

[6) MyTodo.vue - 삭제 기능 7](#_Toc151556513)

[7) MyTodo.vue 수정 - 완료 표시 9](#_Toc151556514)

[2. 연습문제 11](#_Toc151556515)

# todo2 프로젝트

## axios 요약

등록 axios.post("/todos", todo)

삭제 axios.delete("/todos/3")

수정 axios.put("/todos/3", todo)

목록조회 axios.get("/todos")

한개조회 axios.get("/todos/3")

## 프로젝트 생성

프로젝트를 생성할 적당한 폴더로 이동하자.

|  |
| --- |
| cd /c/PJ/frontend |

아래 명령을 실행해서 프로젝트를 생성하자.

|  |
| --- |
| vue create todo2 |

### 프로젝트 폴더 열기

Visual Studio Code에서 프로젝트 폴더를 열자

### axios 설치

|  |
| --- |
| npm install axios |

## src/todos-db.json

|  |
| --- |
| {    "todos": [      { "id": 1, "title": "과제", "done": false },      { "id": 2, "title": "시험공부", "done": false },      { "id": 3, "title":"코딩훈련", "done": false }    ]  } |

### json-server 실행

|  |
| --- |
| json-server --watch src/todos-db.json |

## MyTodo.vue 생성

|  |
| --- |
| <template>    <div id="MyTodo">      <h1>할 일</h1>      <table>        <tr><td>ID</td><td>할일</td></tr>        <tr v-for="todo in todos" v-bind:key="todo.id">          <td>{{ todo.id }}</td>          <td>{{ todo.title }}</td>        </tr>      </table>    </div>  </template>  <script>  import axios from 'axios';  axios.defaults.baseURL = "http://localhost:3000";      // 백엔드 API URL 앞 부분을 미리 설정한다  export default {    name: "MyTodo",    data() {      return {        todos: [ ]      }    },    mounted() { // 컴포넌트가 처음 화면에 보이게 될 때, 자동으로 호출되는 메소드      this.reloadTodos(); // 데이터를 로드한다    },    methods: {      async reloadTodos() { // 백엔드 API를 호출하여 데이터를 로드한다        try {          const response = await axios.get("/todos");          this.todos = response.data;        } catch (error) {          alert('조회 에러: ' + (error instanceof Error ? error.message : error));        }      }    }  }  </script>  <style scoped>  div#MyTodo { padding: 30px; margin: 30px auto; width: 400px;    border: 1px solid #ccc; box-shadow: 3px 3px 3px #aaa; }  h1 { border-bottom: 1px solid gray; }  table { border-collapse: collapse; margin: 20px 0; width: 100%; }  tr:nth-child(1) { background-color: #eee; text-align: center; }  td { border: 1px solid gray; padding: 6px; }  td:nth-child(1) { text-align: center; width: 30px; }  </style> |

### App.vue

|  |
| --- |
| <template>    <div id="app">      <MyTodo />    </div>  </template>  <script>  import MyTodo from './MyTodo.vue'  export default {    name: "App",    components: { MyTodo }  }  </script>  <style scoped>  </style> |

### 실행

|  |
| --- |
| npm run serve |

A screenshot of a computer

Description automatically generated with low confidence

## MyTodo.vue - 추가 기능

|  |
| --- |
| <template>    <div>      <h1>할 일</h1>      <table>        <tr><td>ID</td><td>할일</td></tr>        <tr v-for="todo in todos" v-bind:key="todo.id">          <td>{{ todo.id }}</td>          <td>{{ todo.title }}</td>        </tr>      </table>      <input type="text" v-model="title" />      <button type="button" v-on:click="addTodo">추가</button>    </div>  </template>  <script>  import axios from 'axios';  axios.defaults.baseURL = "http://localhost:3000";  export default {    name: "MyTodo",    data() {      return {        todos: [ ],        title: ""      }    },    mounted() {      this.reloadTodos();    },    methods: {      async reloadTodos() {        try {          const response = await axios.get("/todos");          this.todos = response.data;        } catch (error) {          alert('조회 에러: ' + (error instanceof Error ? error.message : error));        }      },      async addTodo() {        try {          const todo = {id: 0, title: this.title};          await axios.post("/todos", todo); // 새 todo를 백엔드에 등록한다          this.reloadTodos(); // todo 목록 다시 로드          this.title = "";        } catch (error) {          alert('저장 에러: ' + (error instanceof Error ? error.message : error));        }      }    }  }  </script>  <style scoped>  h1 { border-bottom: 1px solid gray; }  div { padding: 30px; margin: 30px auto; width: 400px;    border: 1px solid #ccc; box-shadow: 3px 3px 3px #aaa; }  table { border-collapse: collapse; margin: 20px 0; width: 100%; }  tr:nth-child(1) { background-color: #eee; text-align: center; }  td { border: 1px solid gray; padding: 6px; }  td:nth-child(1) { text-align: center; width: 30px; }  input[type=text] { padding: 5px; margin-right: 5px; width: 300px; }  button { padding: 0.3em 1.5em; }  </style> |

A screenshot of a computer

Description automatically generated with medium confidence

## MyTodo.vue - 삭제 기능

|  |
| --- |
| <template>    <div>      <h1>할 일</h1>      <table>        <tr><td>ID</td><td>할일</td></tr>        <tr v-for="todo in todos" v-bind:key="todo.id">          <td>{{ todo.id }}</td>          <td>            {{ todo.title }}            <span v-on:click="deleteTodo(todo.id)">x</span>          </td>        </tr>      </table>      <input type="text" v-model="title" />      <button type="button" v-on:click="addTodo">추가</button>    </div>  </template>  <script>  import axios from 'axios';  axios.defaults.baseURL = "http://localhost:3000";  export default {    name: "MyTodo",    data() {      return {        todos: [ ],        title: ""      }    },    mounted() {      this.reloadTodos();    },    methods: {      async reloadTodos() {        try {          const response = await axios.get("/todos");          this.todos = response.data;        } catch (error) {          alert('조회 에러: ' + (error instanceof Error ? error.message : error));        }      },      async addTodo() {        try {          const todo = {id: 0, title: this.title };          await axios.post("/todos", todo);          this.reloadTodos();          this.title = "";        } catch (error) {          alert('저장 에러: ' + (error instanceof Error ? error.message : error));        }      },      async deleteTodo(id) {        try {          if (confirm("삭제하시겠습니까?") == false) return;          await axios.delete("/todos/" + id);          this.reloadTodos();        } catch (error) {          alert('삭제 에러: ' + (error instanceof Error ? error.message : error));        }      },    }  }  </script>  <style scoped>  h1 { border-bottom: 1px solid gray; }  div { padding: 30px; margin: 30px auto; width: 400px;    border: 1px solid #ccc; box-shadow: 3px 3px 3px #aaa; }  table { border-collapse: collapse; margin: 20px 0; width: 100%; }  tr:nth-child(1) { background-color: #eee; text-align: center; }  td { border: 1px solid gray; padding: 6px; }  td:nth-child(1) { text-align: center; width: 30px; }  input[type=text] { padding: 5px; margin-right: 5px; width: 300px; }  button { padding: 0.3em 1.5em; }  span { font-weight: bold; margin-left: 10px; cursor: pointer; float: right; }  </style> |

A screenshot of a computer

Description automatically generated with medium confidence

A picture containing text, screenshot, font, rectangle

Description automatically generated

## MyTodo.vue 수정 - 완료 표시

|  |
| --- |
| <template>    <div>      <h1>할 일</h1>      <table>        <tr><td>ID</td><td>할일</td></tr>        <tr v-for="todo in todos" v-bind:key="todo.id" v-bind:class="{done: todo.done}">          <td>{{ todo.id }}</td>          <td>            <input type="checkbox" v-model="todo.done" v-on:click="toggleTodo(todo)"/>            {{ todo.title }}            <span v-on:click="deleteTodo(todo.id)">x</span>          </td>        </tr>      </table>      <input type="text" v-model="title" />      <button type="button" v-on:click="addTodo">추가</button>    </div>  </template>  <script>  import axios from 'axios';  axios.defaults.baseURL = "http://localhost:3000";  export default {    name: "MyTodo",    data() {      return {        todos: [ ],        title: ""      }    },    mounted() {      this.reloadTodos();    },    methods: {      async reloadTodos() {        try {          const response = await axios.get("/todos");          this.todos = response.data;        } catch (error) {          alert('조회 에러: ' + (error instanceof Error ? error.message : error));        }      },      async addTodo() {        try {          const todo = {id: 0, title: this.title };          await axios.post("/todos", todo);          this.reloadTodos();          this.title = "";        } catch (error) {          alert('저장 에러: ' + (error instanceof Error ? error.message : error));        }      },      async deleteTodo(id) {        try {          if (confirm("삭제하시겠습니까?") == false) return;          await axios.delete("/todos/" + id);          this.reloadTodos();        } catch (error) {          alert('삭제 에러: ' + (error instanceof Error ? error.message : error));        }      },      async toggleTodo(todo) {        try {          todo.done = !todo.done;          await axios.put("/todos/" + todo.id, todo);          this.reloadTodos();        } catch (error) {          alert('저장 에러: ' + (error instanceof Error ? error.message : error));        }      },    }  }  </script>  <style scoped>  h1 { border-bottom: 1px solid gray; }  div { padding: 30px; margin: 30px auto; width: 400px;    border: 1px solid #ccc; box-shadow: 3px 3px 3px #aaa; }  table { border-collapse: collapse; margin: 20px 0; width: 100%; }  tr:nth-child(1) { background-color: #eee; text-align: center; }  td { border: 1px solid gray; padding: 6px; }  td:nth-child(1) { text-align: center; width: 30px; }  input[type=text] { padding: 5px; margin-right: 5px; width: 300px; }  button { padding: 0.3em 1.5em; }  span { font-weight: bold; margin-left: 10px; cursor: pointer; float: right; }  tr.done { background-color: #f8f8f8; color: #bbb; text-decoration: line-through; }  input[type=checkbox] { accent-color: #bbb; }  </style> |

<input type="checkbox" v-model="todo.done" v-on:click="toggleTodo(todo)"/>

체크 박스를 클릭하면, 아직 체크 상태가 변경되기도 전에 클릭 이벤트가 먼저 발생하여

toggleTodo(todo) 메소드가 호출된다.

async toggleTodo(todo) {

  try {

    todo.done = !todo.done; // 아직 체크 상태가 변경되기도 전에 toggleTodo 메소드가 호출되었으므로,

// done 속성값도 아직 변경되지 않았다. 따라서 done 속성값을 변경한다.

    await axios.put("/todos/" + todo.id, todo);

// done 속성값이 변경된 todo 객체를 백엔드에 저장한다.

    this.reloadTodos();

  } catch (error) {

    alert('저장 에러: ' + (error instanceof Error ? error.message : error));

  }

},

A screenshot of a computer

Description automatically generated with medium confidence

# 연습문제

A screenshot of a computer

Description automatically generated with medium confidence

todo 객체에 dueDate 속성 추가하기