

2025년 상반기 K-디지털 트레이닝

vue-router와 axios를 사용한 예제(심화 1,2)

[KB] IT's Your Life

vue-router와 axios를 사용한 예제

- ✓ 앞에서 작성한 코드를 다음 조건을 만족하도록 수정하세요.
 - 앞 챕터에서 사용한 db.json을 사용하여 json-server 운영함
 - 로컬변수에 정의된 todos 목록을 제거
 - 모든 데이터는 axios를 이용하여 json-server에서 가져와서 처리하도록 함

vite.config.js 변경

```
...
server: {
  proxy: {
    '/api': {
      target: 'http://localhost:3000',
      changeOrigin: true,
      rewrite: (path) => path.replace(/^\/api/, ''),
    },
  },
},
})
```

db.json

```
{
  "todos": [
    { "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 }
  ]
}
```

○ npx json-server db.json

✓ 데이터 서버 기동

- 새 터미널에서
npx json-server db.json

src/App.vue

```
<template>
  <div class="container">
    <Header />
    <router-view />
  </div>
</template>

<script setup>
import { reactive, provide, computed } from 'vue'
import Header from '@components/Header.vue'
import axios from 'axios';

const BASEURI = "/api/todos";
const states = reactive({ todoList:[]  })
```

src/App.vue

```
//TodoList 목록을 조회합니다.  
const fetchTodoList = async () => {  
  try {  
    const response = await axios.get(BASEURI);  
    if (response.status === 200) {  
      states.todoList = response.data;  
    } else {  
      alert('데이터 조회 실패');  
    }  
  } catch(error) {  
    alert('에러발생 :' + error);  
  }  
}
```

src/App.vue

```
// 새로운 TodoItem을 추가합니다.
const addTodo = async ({ todo, desc }, successCallback) => {
  try {
    const payload = { todo, desc };
    const response = await axios.post(BASEURI, payload);
    if (response.status === 201) {
      states.todoList.push({ ...response.data, done: false });
      successCallback();
    } else {
      alert('Todo 추가 실패');
    }
  } catch (error) {
    alert('에러발생 : ' + error);
  }
};
```


src/App.vue

```
// 기존 TodoItem을 변경합니다.
const updateTodo = async ({ id, todo, desc, done }, successCallback) => {
  try {
    const payload = { id, todo, desc, done };
    const response = await axios.put(BASEURI + `/${id}`, payload);
    if (response.status === 200) {
      let index = states.todoList.findIndex((todo) => todo.id === id);
      states.todoList[index] = payload;
      successCallback();
    } else {
      alert('Todo 변경 실패');
    }
  } catch (error) {
    alert('에러발생 :' + error);
  }
};
```

src/App.vue

```
//기존 TodoItem을 삭제합니다.
const deleteTodo = async (id) => {
  try {
    const response = await axios.delete(BASEURI + `/${id}`);
    console.log(response.status, response.data);
    if (response.status === 200) {
      let index = states.todoList.findIndex((todo) => todo.id === id);
      states.todoList.splice(index, 1);
    } else {
      alert('Todo 삭제 실패');
    }
  } catch (error) {
    alert('에러발생 :' + error);
  }
};
```

src/App.vue

//기존 TodoItem의 완료여부(done) 값을 토글합니다.

```
const toggleDone = async (id) => {
  try {
    let todo = states.todoList.find((todo) => todo.id === id);
    let payload = { ...todo, done: !todo.done };
    const response = await axios.put(BASEURI + `/${id}`, payload);
    if (response.status === 200) {
      todo.done = payload.done;
    } else {
      alert('Todo 완료 변경 실패');
    }
  } catch (error) {
    alert('에러발생 :' + error);
  }
};

provide('todoList', computed(() => states.todoList));
provide('actions', { addTodo, deleteTodo, toggleDone, updateTodo, fetchTodoList })

fetchTodoList();
</script>
```

src/pages/AddTodo.vue 변경

...

```
<script setup>
import { inject, reactive } from 'vue';
import { useRouter } from 'vue-router';

const router = useRouter();
const { addTodo } = inject('actions');
const todoItem = reactive({ todo:"", desc:"" })

const addTodoHandler = () => {
  let { todo } = todoItem;
  if (!todo || todo.trim()=== "") {
    alert('할일은 반드시 입력해야 합니다');
    return;
  }
  addTodo({ ...todoItem }, ()=>{
    router.push('/todos')
  });
}
</script>
```

src/pages/EditTodo.vue 변경

```
...
<script setup>
import { inject, reactive } from 'vue';
import { useRouter, useRoute } from 'vue-router';

const todoList = inject('todoList');
const { updateTodo } = inject('actions');
const router = useRouter();
const currentRoute = useRoute();

const matchedTodoItem = todoList.value.find((item) => item.id === currentRoute.params.id)
if (!matchedTodoItem) {
  router.push('/todos');
}
const todoItem = reactive({ ...matchedTodoItem })
```

src/pages/EditTodo.vue 변경

```
const updateTodoHandler = () => {  
  let { todo } = todoItem;  
  if (!todo || todo.trim() === "") {  
    alert('할일은 반드시 입력해야 합니다');  
    return;  
  }  
  updateTodo({ ...todoItem }, ()=>{  
    router.push('/todos');  
  });  
}  
</script>
```

src/pages/ToDoList.vue 변경

```
<template>
  <div class="row">
    <div class="col p-3">
      <router-link class="btn btn-primary" to="/todos/add">
        할일 추가
      </router-link>
      <button class="btn btn-primary ms-1" @click="fetchToDoList">
        새로 고침
      </button>
    </div>
  </div>
  <div class="row">
    <div class="col p-3">
      <ul class="list-group">
        <ToDoItem v-for="todoItem in todoList" :key="todoItem.id" :todoItem="todoItem" />
      </ul>
    </div>
  </div>
</template>
```

src/pages/ToDoList.vue 변경

```
<script setup>
import {inject} from 'vue';
import TodoItem from '@components/ToDoItem.vue'

const todoList = inject('todoList');
const { fetchTodoList } = inject('actions');
</script>
```