

2025년 상반기 K-디지털 트레이닝

vue-router와 axios를 사용한 예제

[KB] IT's Your Life

vue-router와 axios를 사용한 예제

- ✓ **todolist-app-router 프로젝트를 생성하세요.**
 - 라우트 기능 추가
 - axios, bootstrap@패키지 설치

✓ 프로젝트 생성

```
npm init vue todolist-app-router
```

→ 라우트 기능 설정

```
cd todolist-app-router
```

```
npm install
```

```
npm install bootstrap@5 axios
```

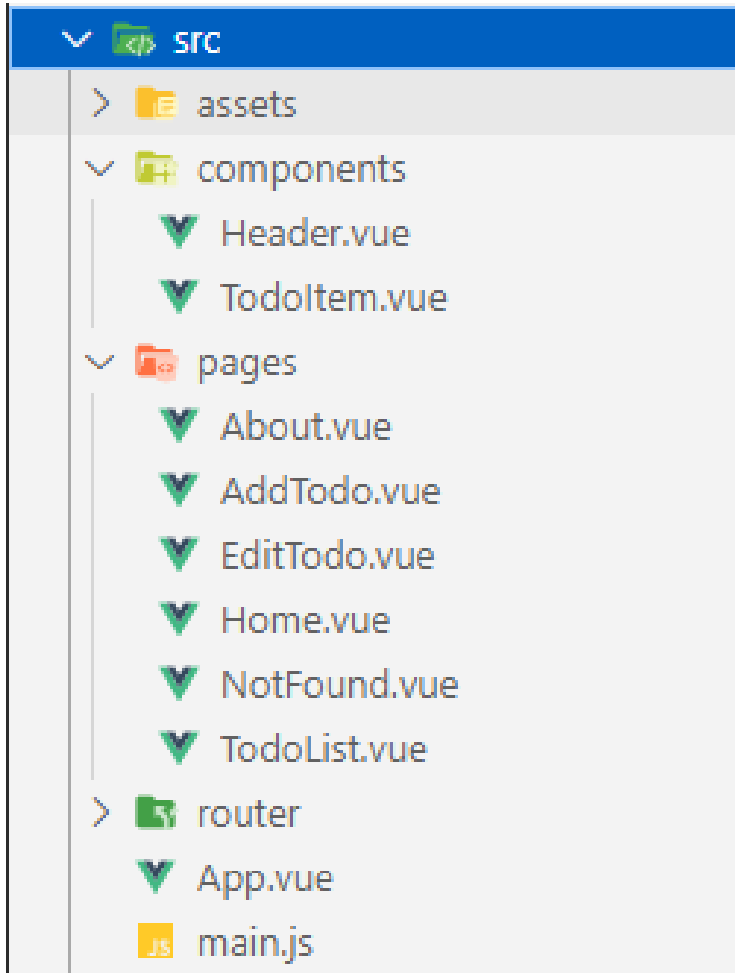
- ✔ 프로젝트 루트 폴더에 db.json 파일을 작성하세요.

```
{
  "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 json-serve로 운영하세요.
- ✔ 브라우저로 todos 엔드 포인트를 확인하세요.

✓ 다음 그림처럼 페이지 컴포넌트와 일반 컴포넌트를 추가하세요.

○ 각 컴포넌트의 템플릿에서는 컴포넌트 이름을 출력함



src/pages/Home.vue

```
<template>
  <h3>Home</h3>
</template>

<script setup></script>
```

- 나머지 페이지 컴포넌트도 같은 방식으로 작성

src/pages/NotFound.vue

```
<template>
  <div class="m-3">
    <h3>존재하지 않는 경로</h3>
    <p>요청 경로 : {{currentRoute.path}}</p>
  </div>
</template>
```

```
<script setup>
import { useRoute } from 'vue-router'
const currentRoute = useRoute();
</script>
```

- ✔ 라우터를 다음과 같이 설정하세요.

경로	컴포넌트
/	Home
/about	About
/todos	TodoList
/todos/add	AddTodo
/todos/edit/:id	EditTodo
그 외	NotFound

src/router/index.js

```
import { createRouter, createWebHistory } from 'vue-router'
import Home from '@pages/Home.vue';
import About from '@pages/About.vue';
import TodoList from '@pages/TodoList.vue';
import AddTodo from '@pages/AddTodo.vue';
import EditTodo from '@pages/EditTodo.vue';
import NotFound from '@pages/NotFound.vue';

const router = createRouter({
  history: createWebHistory(),
  routes: [
    { path: '/', component: Home },
    { path: '/about', component: About },
    { path: '/todos', component: TodoList },
    { path: '/todos/add', component: AddTodo },
    { path: '/todos/edit/:id', component: EditTodo },
    { path: '/*', component: NotFound },
  ]
})

export default router;
```

✓ **main.css**를 다음과 같이 수정하세요.

```
body { margin: 0; padding: 0; font-family: sans-serif; }
.title { text-align: center; font-weight: bold; font-size: 20pt; }
.todo-done { text-decoration: line-through; }
.container { padding: 10px 10px 10px 10px; }
.panel-borderless { border: 0; box-shadow: none; }
.pointer { cursor: pointer; }
```

- ✓ 기존 완성된 Todo 앱을 이번 프로젝트에 반영하세요.

src/components/Header.vue

```
<template>
  <nav class="navbar navbar-expand-sm bg-dark navbar-dark">
    <span class="navbar-brand ps-2">TodoList App</span>
    <button class="navbar-toggler" type="button" @click="isNavShow = !isNavShow">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div :class="isNavShow ? 'collapse navbar-collapse show' : 'collapse navbar-collapse'">
      <ul class="navbar-nav">
        <li class="nav-item">
          <router-link class="nav-link" to="/">Home</router-link>
        </li>
        <li class="nav-item">
          <router-link class="nav-link" to="/about">About</router-link>
        </li>
        <li class="nav-item">
          <router-link class="nav-link" to="/todos">TodoList</router-link>
        </li>
      </ul>
    </div>
  </nav>
</template>
```

src/components/Header.vue

```
<script setup>
import { ref } from 'vue';

const isNavShow = ref(false);
</script>
```

TodoList App Home About TodoList

Todo List

src/App.vue

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

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

const states = reactive({
  todoList : [
    { 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 },
  ]
})
```

src/App.vue

```
const addTodo = ({ todo, desc }) => {
  states.todoList.push({ id: new Date().getTime(), todo, desc, done: false })
};

const updateTodo = ({ id, todo, desc, done }) => {
  let index = states.todoList.findIndex((todo) => todo.id === id);
  states.todoList[index] = { ...states.todoList[index], todo, desc, done };
};

const deleteTodo = (id) => {
  let index = states.todoList.findIndex((todo) => todo.id === id);
  states.todoList.splice(index, 1);
}

const toggleDone = (id) => {
  let index = states.todoList.findIndex((todo) => todo.id === id);
  states.todoList[index].done = !states.todoList[index].done;
}

provide('todoList', computed(() => states.todoList))
provide('actions', { addTodo, deleteTodo, toggleDone, updateTodo })
</script>
```

src/components/TodoItem.vue

```
<template>
  <li :class="todoItem.done ? 'list-group-item list-group-item-success' : 'list-group-item'">
    <span :class="todoItem.done ? 'todo-done pointer' : 'pointer'"
      @click="toggleDone(todoItem.id)">
      {{todoItem.todo}}
      {{todoItem.done ? '(완료)' : '' }}
    </span>
    <span class="float-end badge bg-secondary pointer m-1"
      @click="router.push(`/todos/edit/${todoItem.id}`)">
      편집</span>
    <span class="float-end badge bg-secondary pointer m-1"
      @click="deleteTodo(todoItem.id)">
      삭제</span>
  </li>
</template>
```


src/components/TodoItem.vue

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

defineProps({
  todoItem: { Type: Object, required:true }
})

const router = useRouter();
const { deleteTodo, toggleDone } = inject('actions');
</script>
```

src/pages/ToDoList.vue

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

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

const todoList = inject('todoList');
</script>
```

src/pages/AddTodo.vue

```
<template>
  <div class="row">
    <div class="col p-3">
      <h2>할일 추가</h2>
    </div>
  </div>
  <div class="row">
    <div class="col">
      <div class="form-group">
        <label htmlFor="todo">할일 :</label>
        <input type="text" class="form-control" id="todo" v-model="todoItem.todo" />
      </div>
      <div class="form-group">
        <label htmlFor="desc">설명 :</label>
        <textarea class="form-control" rows="3" id="desc" v-model="todoItem.desc"></textarea>
      </div>
      <div class="form-group">
        <button type="button" class="btn btn-primary m-1" @click="addTodoHandler">추 가</button>
        <button type="button" class="btn btn-primary m-1" @click="router.push('/todos')">취 소</button>
      </div>
    </div>
  </div>
</template>
```

src/pages/AddTodo.vue

```
<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

```
<template>
  <div class="row">
    <div class="col p-3">
      <h2>할일 수정</h2>
    </div>
  </div>
  <div class="row">
    <div class="col">
      <div class="form-group">
        <label htmlFor="todo">할일:</label>
        <input type="text" class="form-control" id="todo" v-model="todoItem.todo" />
      </div>
      <div class="form-group">
        <label htmlFor="desc">설명:</label>
        <textarea class="form-control" rows="3" id="desc" v-model="todoItem.desc"></textarea>
      </div>
      <div class="form-group">
        <label htmlFor="done">완료여부 : </label>&nbsp;
        <input type="checkbox" v-model="todoItem.done" />
      </div>
    </div>
  </div>
</template>
```

src/pages/EditTodo.vue

```
<div class="form-group">
  <button type="button" class="btn btn-primary m-1" @click="updateTodoHandler">
    수 정
  </button>
  <button type="button" class="btn btn-primary m-1" @click="router.push('/todos')">
    취 소
  </button>
</div>
</div>
</div>
</template>
```

src/pages/EditTodo.vue

```
<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 === parseInt(currentRoute.params.id))
if (!matchedTodoItem) {
  router.push('/todos');
}
const todoItem = reactive({ ...matchedTodoItem })

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