KB금융그룹 | 국민의 평생 금융파트너

2025년 상반기 K-디지털 트레이닝

pinia를 이용한 상태 관리

[KB] IT's Your Life

KB 국민은행

pinia를 이용한 상태 관리

- pinia-todo-app 프로젝트를 생성하세요.
  - Pinia 추가

pinia를 이용한 상태 관리

프로젝트 만들기

npm init vue pinia-test-app

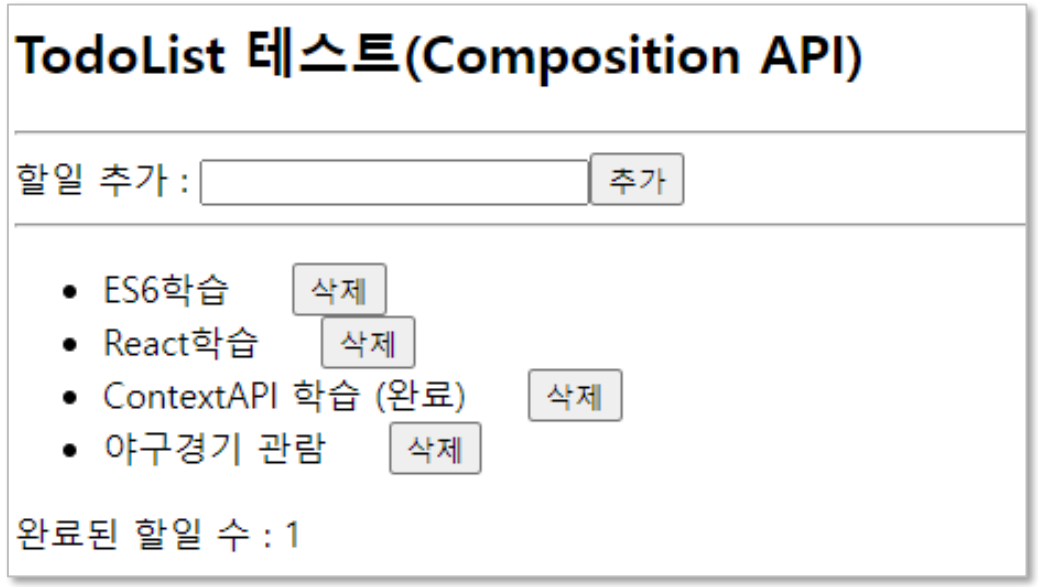
Vue.js - The Progressive JavaScript Framework

✓ Add TypeScript? … No / Yes
✓ Add JSX Support? … No / Yes
✓ Add Vue Router for Single Page Application development? … No / Yes
✓ Add Pinia for state management? … No / Yes
✓ Add Vitest for Unit Testing? … No / Yes
✓ Add an End-to-End Testing Solution? » No
✓ Add ESLint for code quality? … No / Yes

pinia를 이용한 상태 관리

- **다음 화면 처럼 Todo 앱을 Pinia로 작성하세요.**
  - Todo 목록 초기값

```
[
   { id: 1, todo: "ES6학습",  done: false },
   { id: 2, todo: "React학습",  done: false },
   { id: 3, todo: "ContextAPI 학습",  done: true },
   { id: 4, todo: "야구경기 관람", done: false },
]
```



pinia를 이용한 상태 관리

src/stores/todoList.js

```
export const useTodoListStore = defineStore("todoList", ()=> {
    // 방응형 상태
    const state = reactive({
        todoList : [
            { id: 1, todo: "ES6학습",  done: false },
            { id: 2, todo: "React학습",  done: false },
            { id: 3, todo: "ContextAPI 학습",  done: true },
            { id: 4, todo: "야구경기 관람", done: false },
        ]
    })

    // action
    const addTodo = (todo) => {
        state.todoList.push({ id: new Date().getTime(), todo, done: false })
    }

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

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

src/stores/todoList.js

```
    // 계산된 속성
    const doneCount = computed(()=> {
        return state.todoList.filter((todoItem)=>todoItem.done === true).length;
    })

    const todoList = computed(()=> state.todoList);

    return { todoList, doneCount, addTodo, deleteTodo, toggleDone };
})
```

pinia를 이용한 상태 관리

src/App.vue

```
<template>
  <div>
    <h2>TodoList 테스트(Composition API)</h2>
    <hr />
    할일 추가 :
    <input type="text" v-model="todo" />
    <button @click="addTodoHandler">추가</button>
    <hr />
    <ul>
      <li v-for="todoItem in todoList">
        <span style="cursor: pointer" @click="toggleDone(todoItem.id)">
          {{ todoItem.todo }} {{ todoItem.done ? '(완료)' : '' }}
        </span>
           
        <button @click="deleteTodo(todoItem.id)">삭제</button>
      </li>
    </ul>
    <div>완료된 할일 수 : {{ doneCount }}</div>
  </div>
</template>
```

pinia를 이용한 상태 관리

src/App.vue

```
<script setup>
import { useTodoListStore } from '@/stores/todoList.js';
import { ref, computed } from 'vue';

const todo = ref('');

const todoListStore = useTodoListStore();
const { todoList, addTodo, deleteTodo, toggleDone } = todoListStore;

const doneCount = computed(() => todoListStore.doneCount);// 기본 타입에 대해서는 계산된 속성을 다시 작성

const addTodoHandler = () => {
  addTodo(todo.value);
  todo.value = '';
};
</script>
```