

4. View 설정 ; Vue.js

1) Vue 프로젝트 생성

- Vue CLI 사용 : vue create 프로젝트명

vue create client

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Microsoft Windows [Version 10.0.19045.5737]
(c) Microsoft Corporation. All rights reserved.

D:\dev\webworkspace\library_project>vue create client

Manually select features 옵션

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Vue CLI v5.0.8

? Please pick a preset:

vue basic ([Vue 3] babel, router, vuex, eslint)

dev ([Vue 3] babel, router)

Default ([Vue 3] babel, eslint)

Default ([Vue 2] babel, eslint)

> Manually select features

- Babel : ES6+나 TypeScript로 개발 시 범용적인 ES5로 자동 전환해주는 기능

- Router : 라우터를 위한 Vue Router

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Vue CLI v5.0.8

? Please pick a preset: Manually select features

? Check the features needed for your project: (Press <space> to select, <a> to toggle all, <i> to invert selection, and <enter> to proceed)

☒ Babel

☐ TypeScript

☐ Progressive Web App (PWA) Support

> ☒ Router

☐ Vuex

☐ CSS Pre-processors

☐ Linter / Formatter

☐ Unit Testing

☐ E2E Testing

Vue.js 버전

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Vue CLI v5.0.8

? Please pick a preset: Manually select features

? Check the features needed for your project: Babel, Router

? Choose a version of Vue.js that you want to start the project with (Use arrow keys)

> 3.x

2.x

Router에 history 모드 적용

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Vue CLI v5.0.8

? Please pick a preset: Manually select features

? Check the features needed for your project: Babel, Router

? Choose a version of Vue.js that you want to start the project with 3.x

? Use history mode for router? (Requires proper server setup for index fallback in production) (Y/n) y

Babel, ESLint 등 설정을 package.json에 포함

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS node + v

Vue CLI v5.0.8
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, Router
? Choose a version of Vue.js that you want to start the project with 3.x
? Use history mode for router? (Requires proper server setup for index fallback in production) Yes
? Where do you prefer placing config for Babel, ESLint, etc.?
  In dedicated config files
> In package.json
```

현재 설정을 별도로 'devDefault'라는 이름으로 저장

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS node + v

Vue CLI v5.0.8
? Please pick a preset: Manually select features
? Check the features needed for your project: Babel, Router
? Choose a version of Vue.js that you want to start the project with 3.x
? Use history mode for router? (Requires proper server setup for index fallback in production) Yes
? Where do you prefer placing config for Babel, ESLint, etc.? In package.json
? Save this as a preset for future projects? Yes
? Save preset as: devDefault
```

위에 선택한 설정을 기준으로 새로운 프로젝트 생성

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Vue CLI v5.0.8
🌟 Creating project in D:\dev\webworkspace\library_project\client.
📁 Initializing git repository...
⚙️ Installing CLI plugins. This might take a while...

█
```

정상적으로 생성된 경우

```
🎉 Successfully created project client.
👉 Get started with the following commands:

$ cd client
$ npm run serve

D:\dev\webworkspace\library_project>
```

프로젝트 실행 : npm run serve

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

D:\dev\webworkspace\library_project> cd client

D:\dev\webworkspace\library_project\client> npm run serve

> client@0.1.0 serve
> vue-cli-service serve

[INFO] Starting development server...
```

정상적으로 실행된 경우

```
[DONE] Compiled successfully in 3116ms

App running at:
- Local: http://localhost:8080/
- Network: http://192.168.0.2:8080/

Note that the development build is not optimized.
To create a production build, run npm run build.
```

2) 모듈 설치

- axios

- 서버와 데이터를 송수신할 수 있는 HTTP 비동기 통신 라이브러리
- 결과를 Promise 객체 형태로 반환
- JSON 데이터 자동 변환하므로 별도 설정이 필요없음 (=> REST API와 통신하기 좋음)

npm install axios

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

D:\dev\webworkspace\library_project\client>npm install axios

added 8 packages, and audited 829 packages in 2s

106 packages are looking for funding
  run `npm fund` for details

4 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run `npm audit` for details.

D:\dev\webworkspace\library_project\client>

```

3) 프로젝트 구조

3.1. Vue 프로젝트 기본 구조

<div>EXPLORER</div> <div>LIBRARY...</div> <div>client</div>	public/index.html	가장 처음 로드되는 기본 html
<div>node_modules</div> <div>public</div>	src/assets/	정적 파일을 넣어놓는 폴더
<div> <div>★ favicon.ico</div> <div><> index.html</div> </div> <div>src</div>	src/components/	화면을 구성하는 컴포넌트를 넣어놓는 폴더
<div>assets</div> <div>components</div> <div>router</div> <div>views</div>	src/router/	Vue-Router에 대한 설정 및 라우팅 정보를 넣어놓는 폴더
<div>App.vue</div> <div>main.js</div>	src/views/	Vue-Router를 추가한 경우 생성되는 폴더 주로 한 페이지를 구성하는 컴포넌트를 넣어놓는 폴더
<div>.gitignore</div> <div>babel.config.js</div> <div>jsconfig.json</div> <div>package-lock.json</div> <div>package.json</div> <div>README.md</div> <div>vue.config.js</div>	App.vue :	최상위 컴포넌트, Root Vue라고도 함
<div>server</div>	main.js	Application의 진입점
	vue.config.js	vue에 대한 설정(웹팩 등)

3.2. 파일 로딩 순서 : public/index.html -> main.js -> App.vue -> 다른 컴포넌트 들

3.3 vue.config.js 변경

- AJAX를 기반으로 Server와 통신할 경우 Origin이 다른 부분으로 인해 CORS 발생 => proxy 설정

- AJAX(Asynchronous JavaScript And XML)
 - ① javascript를 기반으로 Server와 비동기적 데이터 교환하는 통신
 - ② 서버에 페이지를 요청하는 대신 데이터만 전달받아 Javascript로 페이지를 변경
- Origin : URL 중 프로토콜:host(IP):포트번호로 출처를 가리킴
- CORS(Cross Origin Resource Sharing)
 - ① 서로 다른 Origin 사이에 일어나는 자원 요청(통신)에 대한 설정
 - ② SOP를 위반하는 경우 Server의 CORS 정보에 따라 통신을 허용
- SOP (Same Origin Policy) : 웹 보안 정책, 서로 다른 Origin 사이에 일어나는 자원 요청(통신)을 금지

vue.config.js

```
const defineConfig = require('@vue/cli-service')

// Server의 Origin
const server = 'http://localhost:3000';

module.exports = defineConfig({
  transpileDependencies: true,

  // 개발용 임시 서버
  devServer: {
    // Vue.js 실행 시 적용 PORT 변경
    port: 8099,
    // CORS(Cross Origin Resource Sharing) => proxy setting
    proxy: {
      // 해당 문자열로 시작하는 통신에 적용하는 설정
      '^/api': {
        // 변경할 Origin
        target: server,
        // Origin 변경 : http://localhost:8099 -> http://localhost:3000
        changeOrigin: true,
        // URL 중 일부분을 다시 작성 : /api/books -> /books
        pathRewrite: {'^/api': '/'},
        // websocket 설정 비활성화
        ws: false
      }
    }
  }
})
```

3.4. bootstrap 추가 : CDN 방식

public/index.html

```
<!DOCTYPE html>
<html lang="">
  <head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width,initial-scale=1.0">
    <link rel="icon" href="%= BASE_URL %>favicon.ico">
    <title><%= htmlWebpackPlugin.options.title %></title>
    <!-- bootstrap 5 -->
```

```

<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"rel="stylesheet"integrity="sha384-QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH"crossorigin="anonymous">
<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0xhy9GkcIdslK1eN7N6jIeHz"crossorigin="anonymous"></script>
<!-- -->
</head>
<body>
  <noscript>
    <strong>We're sorry but <%= htmlWebpackPlugin.options.title %> doesn't work properly without JavaScript enabled. Please enable it to continue.</strong>
  </noscript>
  <div id="app"></div>
  <!-- built files will be auto injected -->
</body>
</html>

```

4) VueRouter 초기 설정

main.js

```

import { createApp } from 'vue'
// Root Vue 객체를 구성할 컴포넌트 가져옴
import App from './App.vue'

// Vue Router
// 1) Vue.js에서 SPA을 구현할 때 사용하는 공식 라우터
// 2) 라우팅을 통해 path와 component를 연결
// 3) Vue Router에 등록된 path를 호출할 경우 해당 component를 가져와 출력

// Vue Router 정보를 가져옴
import router from './router'

createApp(App).use(router).mount('#app')

```

src/router/index.js

```

import { createRouter, createWebHistory } from 'vue-router'
import HomeView from '../views/HomeView.vue'

// Vue Router에 등록될 라우팅 정보들
// 기본적으로 path와 name, component를 한쌍으로 등록
// <router-link/> : Vue Router에 등록된 정보를 호출
// <router-view/> : 호출한 라우팅 정보에 따라 component를 출력할 위치 지정
const routes = [
  {
    path: '/',
    name: 'home',
    component: HomeView
  },
  {
    path: '/about',
    name: 'about',
    // route level code-splitting
    // this generates a separate chunk (about.[hash].js) for this route
    // which is lazy-loaded when the route is visited.
    component: () => import(/* webpackChunkName: "about" */ '../views/AboutView.vue')
  }
]

```

```
// 실제 라우터 생성
const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
  routes
})

export default router
```

5) Layout 설정

App.vue

```
<template>
  <div>
    <!-- 공통적으로 사용할 Header -->
    <HeaderComponent/>
    <!-- 사용자의 요청에 따라 Vue Router에 등록된 컴포넌트 출력할 위치 설정 -->
    <!-- key 속성은 컴포넌트의 고유키로 key를 기준으로 각 컴포넌트를 식별 -->
    <!-- 동일한 컴포넌트를 재사용하는 경우 호출정보에 따라 객체를 다시 생성하도록'
key 속성에 this.$route.fullPath를 사용 -->
    <router-view:key="$route.fullPath"/>
    <!-- 공통적으로 사용할 footer -->
    <FooterComponent/>
  </div>
</template>
<script>
  // Component = View(HTML) + Data + Code(Javascript),
  //          화면을 구성하는 기본 단위로 Vue 객체를 의미하기도 함
  import HeaderComponent from './layouts/HeaderComponent.vue';
  import FooterComponent from './layouts/FooterComponent.vue';

  export default{
    // 자식 컴포넌트 등록 -> 등록하지 않은 컴포넌트는 <template>에서 사용할 수 없음
    components : {
      HeaderComponent,
      FooterComponent
    }
  }
</script>
```

src/layouts/HeaderComponent.vue

```
<template>
  <div class="container">
    <svg xmlns="http://www.w3.org/2000/svg" class="d-none">
      <symbol id="bootstrap" viewBox="0 0 118 94">
        <title>Bootstrap</title>
        <path fill-rule="evenodd" clip-rule="evenodd" d="M24.509 0c-6.733 0-11.715
5.893-11.492 12.284.214 6.14-.064 14.092-2.066 20.577C8.943 39.365 5.547 43.485 0
44.014v5.972c5.547.529 8.943 4.649 10.951 11.153 2.002 6.485 2.28 14.437 2.066
20.577C12.794 88.106 17.776 94 24.51 94H93.5c6.733 0 11.714-5.893
11.491-12.284-.214-6.14-.064-14.092 2.066-20.577 2.009-6.504 5.396-10.624
10.943-11.153v-5.972c-5.547-.529-8.934-4.649-10.943-11.153-2.002-6.484-2.28-14.437-2.066-2
0.577C105.214 5.894 100.233 0 93.5 0H24.508zM80 57.863C80 66.663 73.436 72 62.543 72H44a2 2
0 0 1-2-2V24a2 2 0 0 12-2h18.437c9.083 0 15.044 4.92 15.044 12.474 0 5.302-4.01 10.049-9.119
10.88v.277C75.317 46.394 80 51.21 80 57.863zM60.521 28.34H49.948v14.934h8.905c6.884 0
10.68-2.772 10.68-7.727 0-4.643-3.264-7.207-9.012-7.207zM49.948 49.2v16.458H60.91c7.167 0
10.964-2.876 10.964-8.281 0-5.406-3.903-8.178-11.425-8.178H49.948z"></path>
      </symbol>
    </svg>

    <header class="d-flex flex-wrap justify-content-center py-3 mb-4 border-bottom">
```



```

        <router-link class="d-flex align-items-center mb-3 mb-md-0 me-md-auto
link-body-emphasis text-decoration-none"to="/">
        <svg class="bi me-2"width="40"height="32"><use xlink:href="#bootstrap"/></svg>
        <span class="fs-4">게시판</span>
    </router-link>

    <!-- 메뉴들을 미리 정의 -->
    <ul class="nav nav-pills">
        <li class="nav-item">
            <router-link to="#"class="nav-link active"aria-current="page">Home</router-link>
        </li>
        <li class="nav-item">
            <router-link to="#"class="nav-link">전체조회</router-link>
        </li>
        <li class="nav-item">
            <router-link to="#"class="nav-link">글 등록</router-link>
        </li>
    </ul>
</header>
</div>
</template>

```

src/layouts/FooterComponent.vue

```

<template>
    <div class="container">
        <svg xmlns="http://www.w3.org/2000/svg"class="d-none">
            <symbol id="bootstrap"viewBox="0 0 118 94">
                <title>Bootstrap</title>
                <path fill-rule="evenodd"clip-rule="evenodd"
                    d="M24.509 0c-6.733 0-11.715 5.893-11.492 12.284.214 6.14-.064
14.092-2.066 20.577C8.943 39.365 5.547 43.485 0 44.014v5.972c5.547.529 8.943 4.649 10.951
11.153 2.002 6.485 2.28 14.437 2.066 20.577C12.794 88.106 17.776 94 24.51 94H93.5c6.733 0
11.714-5.893 11.491-12.284-.214-6.14-.064-14.092 2.066-20.577 2.009-6.504 5.396-10.624
10.943-11.153v-5.972c-5.547-.529-8.934-4.649-10.943-11.153-2.002-6.484-2.28-14.437-2.066-20.577
C105.214 5.894 100.233 0 93.5 0H24.508zM80 57.863C80 66.663 73.436 72 62.543 72H44a2 2
0 0 1-2-2V24a2 2 0 0 12-2h18.437c9.083 0 15.044 4.92 15.044 12.474 0 5.302-4.01 10.049-9.119
10.88v.277C75.317 46.394 80 51.21 80 57.863zM60.521 28.34H49.948v14.934h8.905c6.884 0
10.68-2.772 10.68-7.727 0-4.643-3.264-7.207-9.012-7.207zM49.948 49.2v16.458H60.91c7.167 0
10.964-2.876 10.964-8.281 0-5.406-3.903-8.178-11.425-8.178H49.948z">
                </path>
            </symbol>
        </svg>
        <footer class="d-flex flex-wrap justify-content-between align-items-center py-3
py-4 border-top">
            <p class="col-md-4 mb-0 text-body-secondary">&copy;2024 Company, Inc</p>

            <a href="#"
                class="col-md-4 d-flex align-items-center justify-content-center mb-3
mb-md-0 me-md-auto link-body-emphasis text-decoration-none">
                <svg class="bi me-2"width="40"height="32">
                    <use xlink:href="#bootstrap"/>
                </svg>
            </a>

        </footer>
    </div>
</template>
<script>
export default{
    name : 'FooterComponent'
}
</script>

```

6) 기능구현

6.1. Component

- Component 란?

Vue.js의 구조	화면(View) = 작은 단위의 View 의 집합
정의	한 화면을 이루는 작은 요소 혹은 한 화면 전체 작은 단위의 View를 재사용할 수 있는 구조로 개발
구성요소	HTML + CSS + JS

- Component 구조

1) Option API : 기존 방식으로 객체를 기반으로 각 옵션을 중심으로 코드를 배치

```
<template>
  <!-- HTML -->
  <div></div>
</template>

<script setup>
export default{
  components:{}, // 다른 컴포넌트 사용 시 컴포넌트를 import하고, 배열로 저장

  data() {
    // HTML과 JS 코드에서 사용할 데이터 변수 선언 ( CRUD )
    return{
      sampleData:'',
    };
  },
  computed() {}, // HTML과 JS 코드에서 사용할 데이터 변수 선언 (계산 값, Read Only )
  created() {}, // 컴포넌트가 생성되면 실행
  mounted() {}, // template에 정의된 HTML 코드가 렌더링된 후 실행
  methods:{}, // 컴포넌트 내에서 사용할 메소드 정의
  watch:{}, // 변수 변경 시 자동으로 진행할 작업을 메소드로 정의
};
</script>

<style>
/* CSS */
</style>
```

2) Composition API : 옵션을 선언하지 않고 import한 함수를 사용해 각 옵션의 역할을 대신함.

```
<template>
  <!-- HTML -->
  <div></div>
</template>

<script setup>
// <script>에 setup을 선언해 composiont API 방식으로 작성

// vue 모듈에서 목적에 따라 필요한 함수를 호출
import { ref, reactive, computed, watch, onBeforeMount, ... } from 'vue';

// Option API와 달리 객체를 요구하지 않음
// <script>에 선언된 모든 변수와 함수를 그대로 Vue에 등록

</script>
```



```
<style>
  /* CSS */
</style>
```

3) Option API vs Composition API

Option API	Composition API
<template>	동일하게 사용
components	컴포넌트를 import할 경우 그대로 자식 컴포넌트로 등록
data	각 데이터의 타입과 목적에 따라 나눠짐. - ref() : 원시타입, 참조타입 모두 가능하며 새로운 값으로 재할당된 경우도 추적 - reactive() : 원시타입은 불가하며 참조타입만 가능. 재할당 시 반응이 끊어짐
computed	computed() 로 대체
created	Composition API를 사용할 경우 객체가 생성되는 created 혹은 존재하지 않음. 대신 beforeMount 혹은 해당하는 onBeforeMount()를 사용
mounted	onMounted() 로 대체
methods	<script> 안에 선언되는 모든 함수를 그대로 Vue에 등록
watch	watch() 로 대체
<style> <style scoped>	동일하게 사용

4) Composition API란?

- Vue3에서 새로 추가된 함수 기반의 API
- 기존 방식인 Option API는 프로젝트 규모가 커질수록 컴포넌트의 계층 구조가 복잡
-> 추적 및 관리가 어려움
=> 이런 단점을 보완하기 위해 추가된 방식
- 옵션을 선언하는 대신 import한 함수를 사용
- 객체 구조 없이 함수로 코드를 작성해 컴포넌트 로직을 유연하게 구성
-> 코드의 재사용성과 가독성을 높임

5) Composition API 방식

- ① ref() : 하나의 인수를 초기값으로 받아 반응형 객체를 반환하는 함수
 - 데이터 타입 : 기본값(원시타입), 참조타입(배열 객체) 전부 가능
 - 중첩 프로퍼티까지 전부 반응형으로 설정하므로 깊은 구조를 가진 데이터인 경우 성능 저하가 발생할 수 있음.
-> 필요에 따라 shallowRef()를 통해 내부 프로퍼티의 감지를 끄는 방식으로 선언
 - 사용방법

```
<template>
  <button@click="increment">
    {{ count }}
  </button>
</template>

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

// ref() 함수는 value 프로퍼티에 인수 값을 가지는 반응형 객체를 반환
// value 프로퍼티로 값을 변경하거나 가져오는 작업이 반드시 필요

const count = ref(0);
```

```
function increment() {
  count.value++
}
</script>
```

② reactive() : ref()와 같이 반응형 객체를 반환하는 함수.

- 데이터 타입 : 참조타입(배열, 객체)만 가능
- 중첩 프로퍼티까지 전부 반응형으로 설정하므로 깊은 구조를 가진 데이터인 경우 성능 저하가 발생할 수 있음.
-> 필요에 따라 shallowReactive()를 통해 내부 프로퍼티의 감지를 끄는 방식으로 선언
- 사용방법

```
<template>
  <button@click="increment">
    {{state.count}}
  </button>
</template>

<script setup>
import { reactive } from 'vue'

const state = reactive({ count:0});

function increment() {
  // reactive() 함수는 ref()와 달리 value 프로퍼티를 사용하지 않고 값에 접근가능
  state.count++
}
</script>
```

- ref()와 reactive() 차이점

	ref()	reactive()
데이터 타입	제한없음	참조타입(객체, 배열)만 가능
재할당 가능여부	값을 변경해도 추적이 가능	값을 변경할 경우 추적이 끊어짐
사용	<ul style="list-style-type: none"> - 원시 데이터 타입을 다루는 상황 - 자신과 프로퍼티를 모두 재할당할 경우 - shallowRef() : 프로퍼티를 관찰하지 않는 경우 	<ul style="list-style-type: none"> - 객체 타입의 데이터의 프로퍼티와 중첩 프로퍼티를 모두 관찰 - shallowReactive() : 중첩 프로퍼티를 관찰하지 않는 경우

③ computed : Option API의 computed 옵션에 대응하는 함수

- 특정 반응형 데이터를 바탕으로 새로운(계산된) 반응형 데이터를 생성
- 읽기 전용
- 사용방법

```
<template>
  <p>책을 가지고 있다:</p>
  <span>{{publishedBooksMessage}}</span>
</template>

<script setup>

import { reactive, computed } from 'vue'
```

```

const author = reactive({
  name: 'John Doe',
  books: [
    'Vue 2 - Advanced Guide',
    'Vue 3 - Basic Guide',
    'Vue 4 - The Mystery'
  ]
})

// 계산된 ref : 함수를 기반으로 계산식을 정의
const publishedBooksMessage = computed(() => {
  return author.books.length > 0 ? 'Yes' : 'No'
})

// 값을 불러오는 경우 value 프로퍼티를 사용
console.log(publishedBooksMessage.value); // Yes
</script>

```

④ watch : Option API의 watch 옵션에 대응하는 함수

- 반응형 데이터의 변화를 관찰하고 지정된 작업을 자동으로 실행
- 사용방법

```

<template>
  <p>
    예/아니오 질문을 하세요:
    <input v-model="question":disabled="loading"/>
  </p>
  <p>{{answer}}</p>
</template>

<script setup>
import { ref, watch } from 'vue'

const question = ref('')
const answer = ref('질문에는 보통 물음표가 포함됩니다. ;-)')
const loading = ref(false);

// watch는 ref에 직접 작동합니다
// 첫번째 매개변수 : 반응형 객체
// 두번째 매개변수 : 콜백함수로, 변화가 감지된 경우 진행할 작업
watch(question, async(newQuestion, oldQuestion) => {

  if(newQuestion.includes('?')) {
    loading.value = true
    answer.value = '생각 중...'

    try{
      const res = await fetch('https://yesno.wtf/api')
      answer.value = (await res.json()).answer
    } catch(error) {
      answer.value = '오류! API에 도달할 수 없습니다. '+error
    } finally{
      loading.value = false
    }
  }
});
</script>

```

- LifeCycle Hooks : 라이프사이클(LifeCycle) 각 단계에서 훅(Hook)을 할 수 있도록 API를 제공
- 라이프사이클(LifeCycle)
 - ① Vue 인스턴스나 컴포넌트가 생성되며 거치는, 미리 사전에 정의된 몇 단계의 과정
 - ② Vue 인스턴스가 생성된 후 우리 눈에 보여지고, 사라지기까지의 단계

훅(Hook)	Composition API	설명
beforeCreate	setup	Composition API에서 새로 추가된 훅으로 초기화 진행 - beforeCreate와 created 훅은 Composition API에 없음
created		
beforeMount	onBeforeMount	DOM에 부착하기 직전에 호출 - 템플릿을 렌더링 한 상태로 가상 DOM이 생성 - 실제 DOM에 부착되지는 않은 상태
mounted	onMounted	가상 DOM의 내용이 실제 DOM에 부착되고 난 이후에 실행
beforeUpdate	onBeforeUpdate	data의 값이 변해 DOM에도 그 변화를 적용시키기 직전에 호출
updated	onUpdated	가상 DOM을 렌더링 하고 실제 DOM이 변경된 이후에 호출 - 변경된 data가 DOM에도 적용된 상태
beforeDestroy	onBeforeUnmount	컴포넌트 인스턴스가 해체되기 직전에 호출 - 아직 인스턴스가 작동하므로 모든 속성에 접근 가능
destroyed	onUnmounted	컴포넌트가 마운트 해제된 상태

- 실제 DOM vs 가상 DOM

실제 DOM (Real DOM)	가상 DOM (Virtual DOM)
<ul style="list-style-type: none"> - Document Object Model - 정적 파일인 HTML을 메모리에 올린 형태로 트리구조 - 트리를 구성하는 각 Node의 일부가 변해도 전체 변경 - 브라우저를 통해 화면에 실제로 출력되는 부분을 담당 	<ul style="list-style-type: none"> - Vue가 제어하는 대상 - 실제 DOM 문서 추상화 - 화면에 변화가 발생할 경우 <ol style="list-style-type: none"> 1. 가상의 DOM을 기반으로 수정 사항을 반영 2. 실제 DOM과 비교하여 변경된 사항을 확인 3. 실제 DOM에 확인된 부분을 반영

- 기본 문법 : Directive

- 데이터 바인딩 : View와 Model을 하나로 묶어 연결하는 것, 이 때 디렉티브(Directive)를 사용

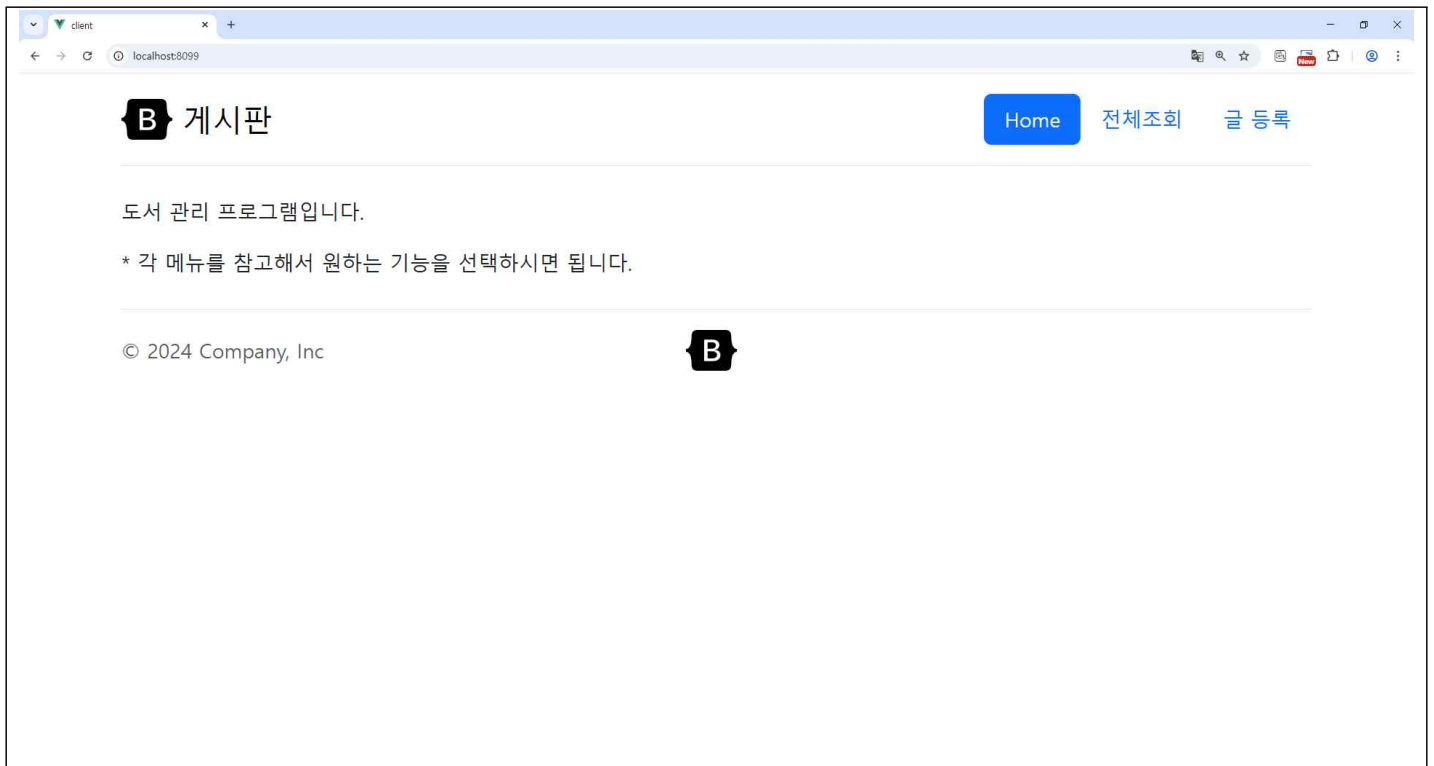
종류	바인딩 대상	바인딩 방향	설명
v-bind	태그의 속성	단방향	태그의 각 속성에 변수(데이터)를 연결
v-model	value 속성	양방향	입력태그의 value 속성에 변수(데이터)를 연결 - 단, <input>의 type이 checkbox와 radio인 경우 checked에 연결
v-for	태그	단방향	변수(데이터)가 배열인 경우 사용, 배열의 개수에 따라 태그 생성
v-if	태그	단방향	변수(데이터)을 기반으로 한 조건식에 따라 태그를 생성
v-show	태그	단방향	변수(데이터)을 기반으로 한 조건식에 따라 태그를 보여줌
v-on	on이벤트 속성	단방향	이벤트가 발생할 경우 동작할 함수(이벤트 핸들러)를 연결

- 바인딩 방향

- ① 단방향인 경우 변수(데이터)를 기준으로 일방적으로 속성과 태그를 변경
- ② 양방향인 경우 변수(데이터)를 기준으로 속성을 변경할 수 있으며 동시에 변경된 속성 값을 변수에 전달

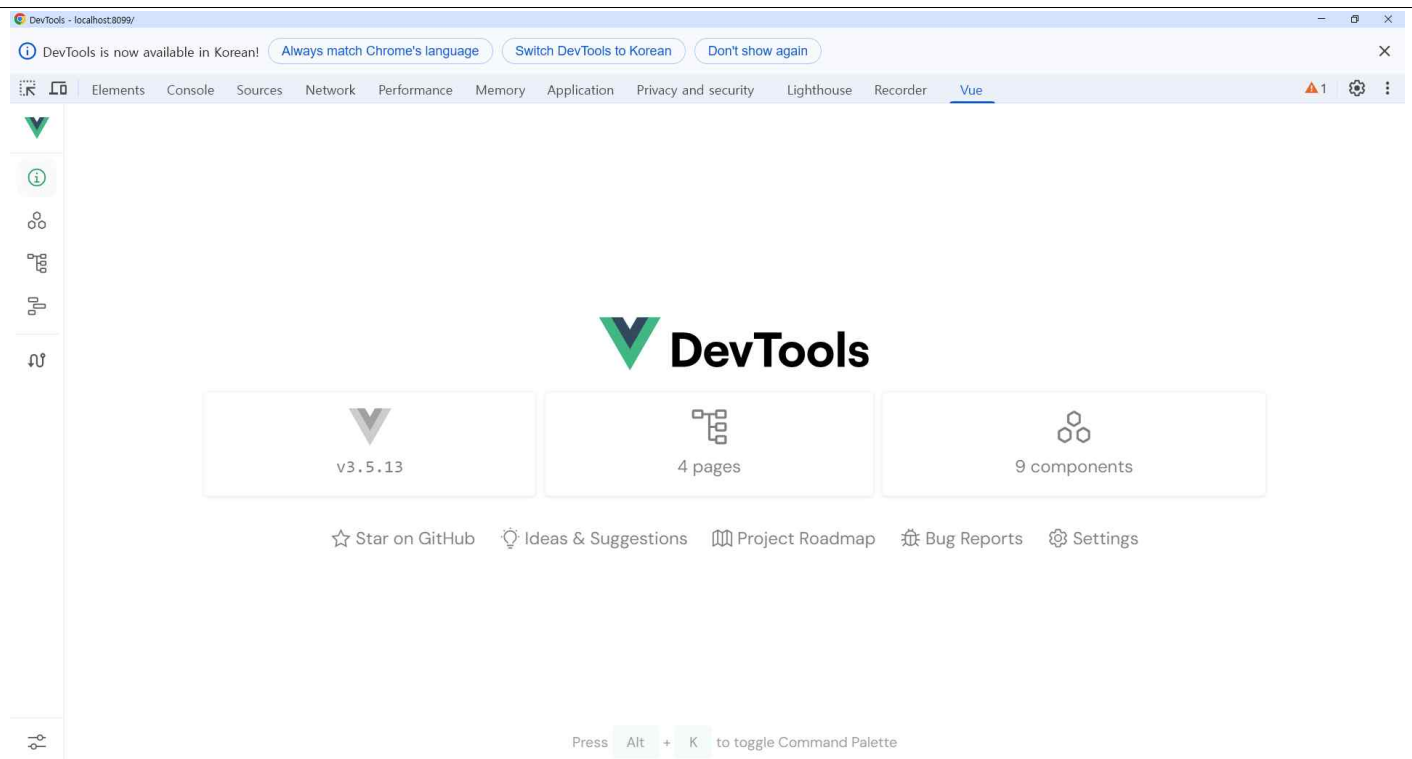
8) 구현결과(개발자 도구의 Network 탭과 Vue devtools 포함)

8.1. 메인페이지



Vue devtools

- 사이트에 적용된 Vue 객체 및 관련 정보를 확인할 수 있는 도구



현재 화면을 구성하고 있는 컴포넌트 구조 및 정보 확인

DevTools - localhost:8099/

DevTools is now available in Korean! [Always match Chrome's language](#) [Switch DevTools to Korean](#) [Don't show again](#)

Elements Console Sources Network Performance Memory Application Privacy and security Lighthouse Recorder **Vue**

Find components...

<Root>

- <HeaderComponent>
 - <RouterLink> /
 - <RouterLink> /
 - <RouterLink> /list
 - <RouterLink> /form
- <RouterView key="/"> main: /
 - <MainContent>
 - <FooterComponent>

<MainContent> Filter State...

provided

- Symbol(router view depth) : undefined(Computed)
- Symbol(router view location matched) : Object(Computed)
- Symbol(router view location) : Object(Computed)

event listeners

- vnode-unmounted : "⚠ Not declared"

Routing

- \$route : "/"
 - fullPath : /
 - path : /
 - query : Object (empty)
 - hash :
 - name : main
 - params : Object (empty)
- matched : Array[1]
- meta : Object (empty)
- redirectedFrom : undefined
- href : /

Router에 등록된 라우팅 정보 확인, 각 path를 선택할 경우 해당 컴포넌트 요청

DevTools - localhost:8099/

DevTools is now available in Korean! [Always match Chrome's language](#) [Switch DevTools to Korean](#) [Don't show again](#)

Elements Console Sources Network Performance Memory Application Privacy and security Lighthouse Recorder **Vue**

Current route

/

Edit path above to navigate

All Routes

4 routes registered in your application

Route Path	Name
/	main
/form	bookForm
/info/ bno	bookInfo
/list	bookList

Router에 등록된 라우팅 정보 확인. 실제 라우팅을 구성하는 여러 정보를 확인할 수 있음.

The screenshot shows the Chrome DevTools interface with the Vue Router component inspector open. The left sidebar contains icons for various DevTools panels, with the Router icon (a green circle with a white 'R') highlighted by a red square. The main panel is divided into two sections: 'Routes' and 'options'.

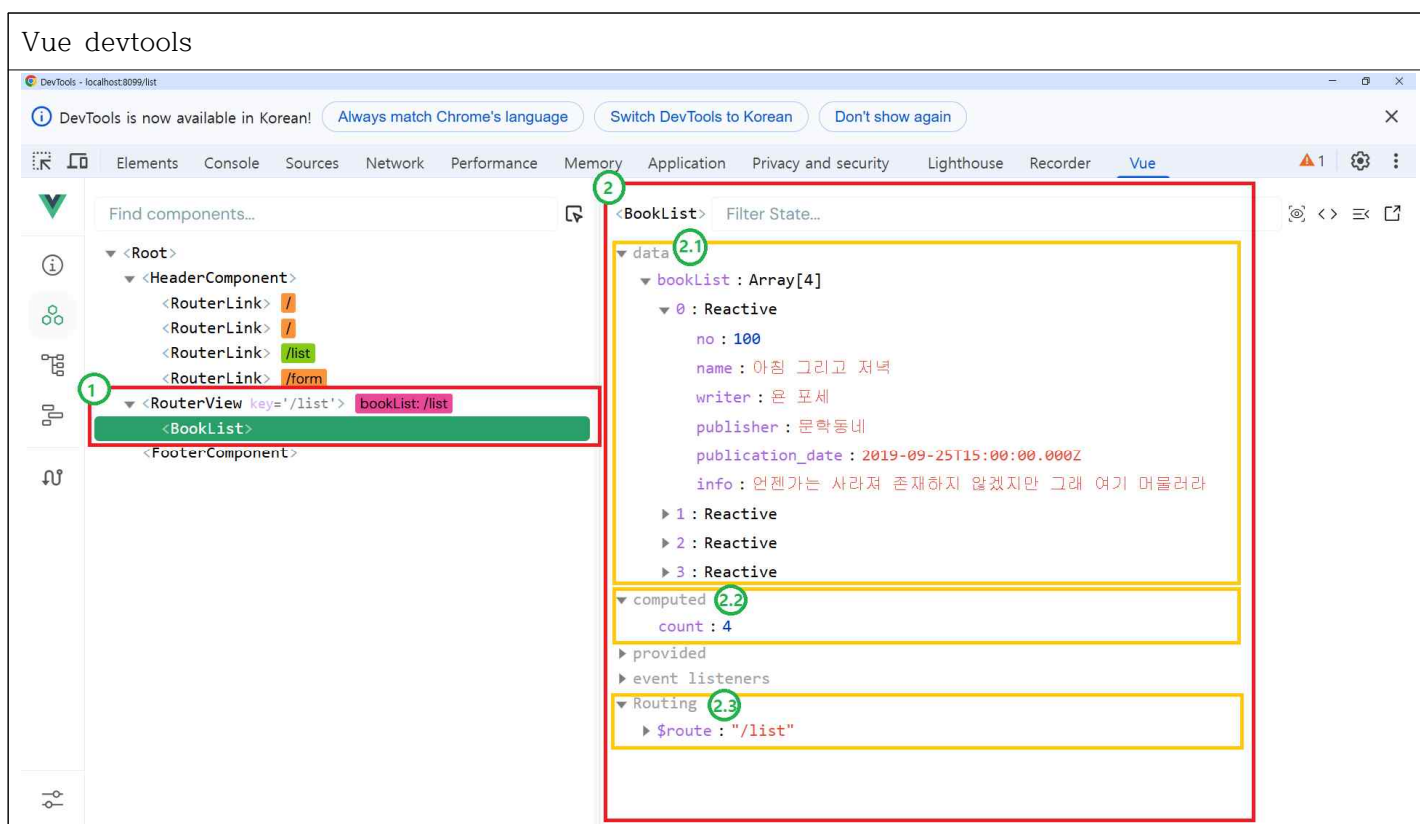
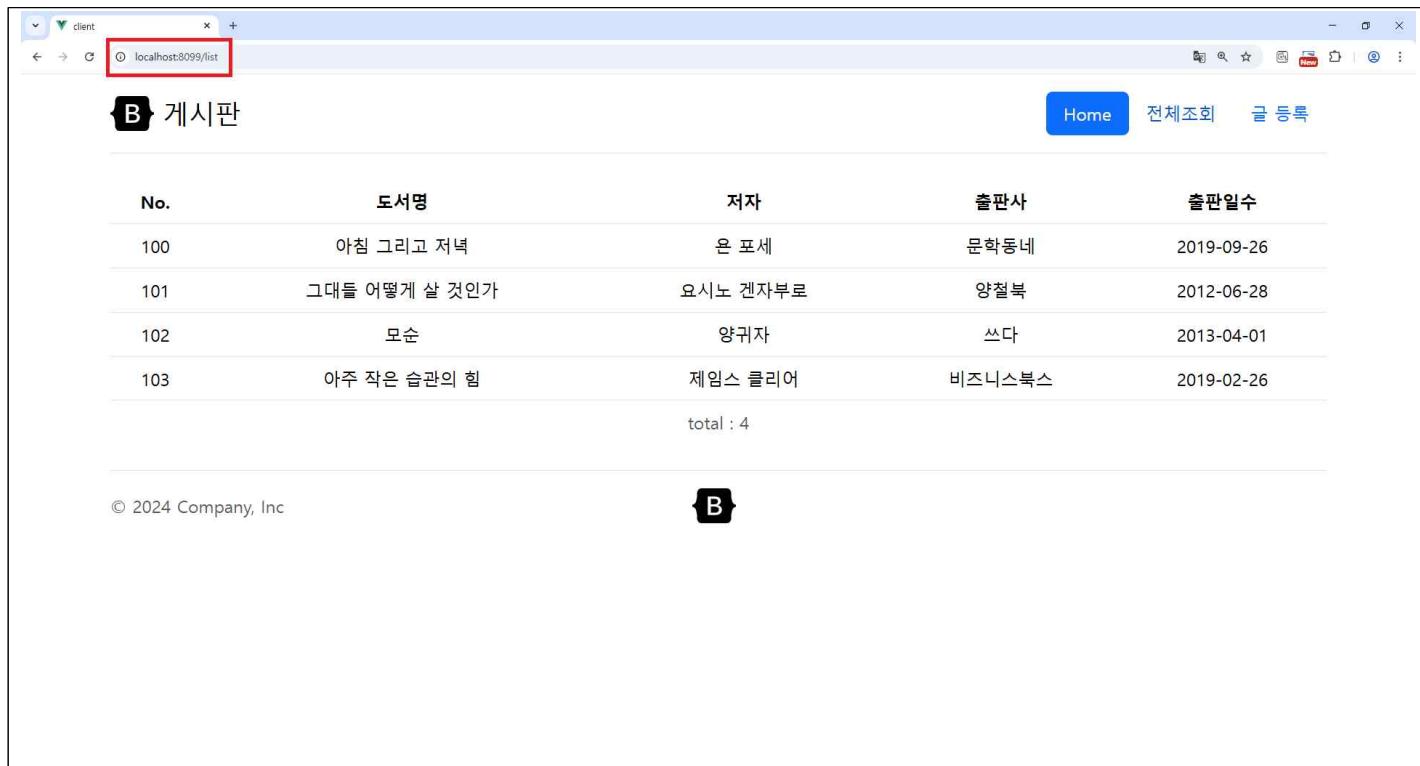
Routes: A search bar is at the top. Below it, a list of routes is displayed:

- `/info/:bno` with component `bookInfo`
- `/` with component `main`, flags `exact` and `active`
- `/list` with component `bookList`
- `/form` with component `bookForm`

options: A dropdown menu is expanded, showing the following configuration:

```
path : /
name : main
regexp : /^\/\/?$/i
score : "80"
```

8.2. 전체조회



- 1) Router에 등록된 라우팅을 요청한 결과
- 2) 선택한 컴포넌트가 가지고 있는 정보
 - 2.1. data 옵션에 등록된 프로퍼티, 실시간으로 현재 가지고 있는 값이 출력됨
 - 2.2. computed 옵션에 정의된 프로퍼티, 실시간으로 계산된 결과값이 출력됨
 - 2.3 Router를 통해 요청된 경우 라우팅 정보를 가짐

AJAX(axios 모듈 실행결과)를 통해 실제 Server에 통신한 결과 확인 : 개발자도구(F12)의 Network 탭

1) Request(요청)

The screenshot shows the Chrome DevTools Network tab. The 'Network' tab is selected at the top. A list of requests is on the left, with 'books' selected. The 'Headers' sub-tab is active, showing the 'General' section. The request details are as follows:

Request URL:	http://localhost:8099/api/books
Request Method:	GET
Status Code:	304 Not Modified
Remote Address:	127.0.0.1:8099
Referrer Policy:	strict-origin-when-cross-origin

The 'Response Headers' section shows the following details:

Request Headers	Raw
Accept:	application/json, text/plain, */*
Accept-Encoding:	gzip, deflate, br, zstd
Accept-Language:	ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7
Connection:	keep-alive
Host:	localhost:8099
If-None-Match:	W/"3c2-rGY3h806CID0DaNSEMxQljgDeE"
Referer:	http://localhost:8099/list
Sec-Ch-UA:	"Google Chrome";v="135", "Not-A.Brand";v="8", "Chromium";v="135"

2) Response(응답)

The screenshot shows the Chrome DevTools Network tab. The 'Network' tab is selected at the top. A list of requests is on the left, with 'books' selected. The 'Preview' sub-tab is active, showing the response data. The response is a JSON array of book objects:

```
[{"no": 100, "name": "아침 그리고 저녁", "writer": "윤 포세", "publisher": "문학동네", "..."}, {"no": 101, "name": "그대들 어떻게 살 것인가", "writer": "요시노 겐자부로", "publisher": "양철북", "..."}, {"no": 102, "name": "모순", "writer": "양귀자", "publisher": "쓰다", "publication_date": "2013-03-31T15:00:00.000Z", "..."}, {"no": 103, "name": "아주 작은 습관의 힘", "writer": "제임스 클리어", "publisher": "비즈니스북스", "..."}]
```

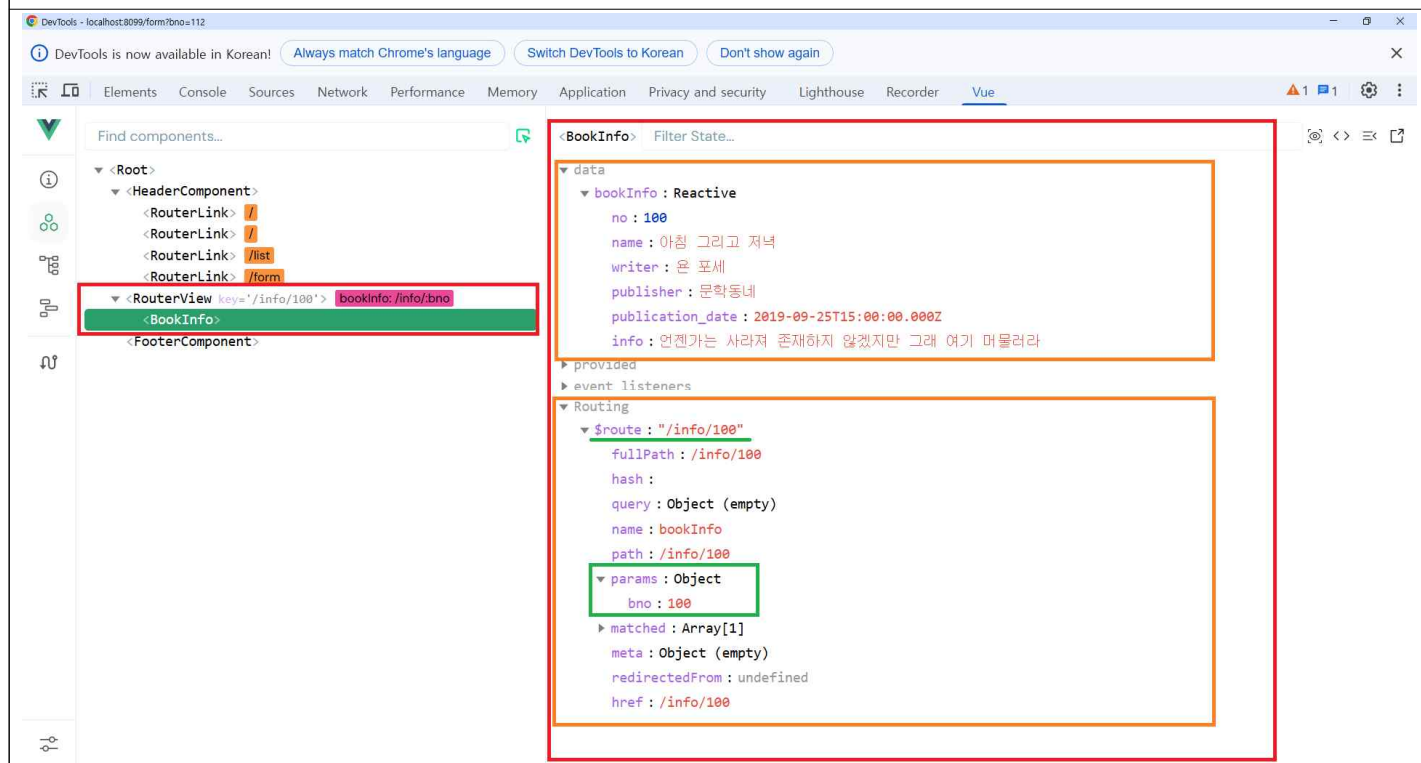
8.3. 단건조회



Vue devtools

- BookList.vue에서 실제 요청 코드 o

```
this.$router.push({ name: 'bookInfo', params: { bno : bookNo } });
```



AJAX(axios 모듈 실행결과)를 통해 실제 Server에 통신한 결과 확인

1) Request(요청)

The screenshot shows the Chrome DevTools Network tab. The 'Headers' sub-tab is selected. The request is a GET to `http://localhost:8099/api/books/100`. The status is `304 Not Modified`. The 'Request Headers' section shows various headers including `Accept`, `Accept-Encoding`, `Accept-Language`, `Connection`, `Host`, `If-None-Match`, `Referer`, and `Sec-Ch-Ua`.

Header	Value
Request URL	<code>http://localhost:8099/api/books/100</code>
Request Method	<code>GET</code>
Status Code	<code>304 Not Modified</code>
Remote Address	<code>127.0.0.1:8099</code>
Referrer Policy	<code>strict-origin-when-cross-origin</code>
Accept	<code>application/json, text/plain, */*</code>
Accept-Encoding	<code>gzip, deflate, br, zstd</code>
Accept-Language	<code>ko-KR,ko;q=0.9,en-US;q=0.8,en;q=0.7</code>
Connection	<code>keep-alive</code>
Host	<code>localhost:8099</code>
If-None-Match	<code>W/"df-OZQW8pHT6XTYFLn7q0rZQVqhuOI"</code>
Referer	<code>http://localhost:8099/info/100</code>
Sec-Ch-Ua	<code>"Google Chrome";v="135", "Not-A.Brand";v="8", "Chromium";v="135"</code>

2) Response(응답)

The screenshot shows the Chrome DevTools Network tab. The 'Preview' sub-tab is selected. The response is a JSON object representing a book entry.

```
{
  "no": 100,
  "name": "\uc54c\uc77c \uc77c\uc77c",
  "writer": "\uc6b4 \uc6b4",
  "publisher": "\uc6b4\uc6b4",
  "info": "\uc6b4\uc6b4\uc6b4\uc6b4\uc6b4\uc6b4\uc6b4\uc6b4\uc6b4\uc6b4\uc6b4",
  "publication_date": "2019-09-25T15:00:00.000Z",
  "publisher": "\uc6b4\uc6b4",
  "writer": "\uc6b4 \uc6b4"
}
```

8.4. 등록

- 1) 메뉴를 선택

The image shows a web browser window displaying a book registration form titled "게시판" (Board). The form includes input fields for "No.", "도서명" (Book Title), "저자" (Author), "출판사" (Publisher), "출판일자" (Publication Date) (pre-filled with 2025-04-22), and "소개" (Introduction). A green "저장" (Save) button is at the bottom.

Below the browser window, the Chrome DevTools component inspector is open for the "BookForm" component. The component tree on the left shows the hierarchy: <Root> > <HeaderComponent> > <RouterView key='/form'> > <BookForm> (highlighted). The component details on the right show the following structure:

```
<BookForm> Filter State...
  data
    bookInfo : Reactive
      no :
      name :
      writer :
      publisher :
      publication_date : 2025-04-22
      info :
      isUpdated : false
  provided
    Symbol(router view depth) : undefined(Computed)
    Symbol(router view location matched) : Object(Computed)
    Symbol(router view location) : Object(Computed)
  event listeners
    vnode-unmounted : ⚠ Not declared
  Routing
    $route : "/form"
```


-2) 사용자가 등록한 데이터를 입력

The top part of the image shows a web browser window at localhost:8099/form. The page has a header with a logo and navigation links (Home, 전체조회, 글 등록). The main content area contains a form with the following fields: No. (text input), 도서명 (text input), 출판의 자리 (text input), 저자 (text input), 정해연 (text input), 출판사 (text input), 엘릭시르 (text input), 출판일자 (text input, value: 2025-04-21), and 소개 (text area). A '저장' (Save) button is at the bottom left of the form. The bottom part of the image shows the Chrome DevTools component inspector for the 'BookForm' component. The 'data' property is expanded, showing the following reactive data: bookInfo: { no: , name: 출판의 자리, writer: 정해연, publisher: 엘릭시르, publication_date: 2025-04-21, info: 이 행복이 영원할 거라고 생각한 적은 없었다. 그러나 이런 끝을 상상한 적도 없었다., isUpdated: false }. The 'provided' property shows router view depth: undefined, router view location matched: Object, and router view location: Object. The 'event listeners' property shows a warning for 'vnode-unmounted: Not declared'. The 'Routing' property shows \$route: '/form'.

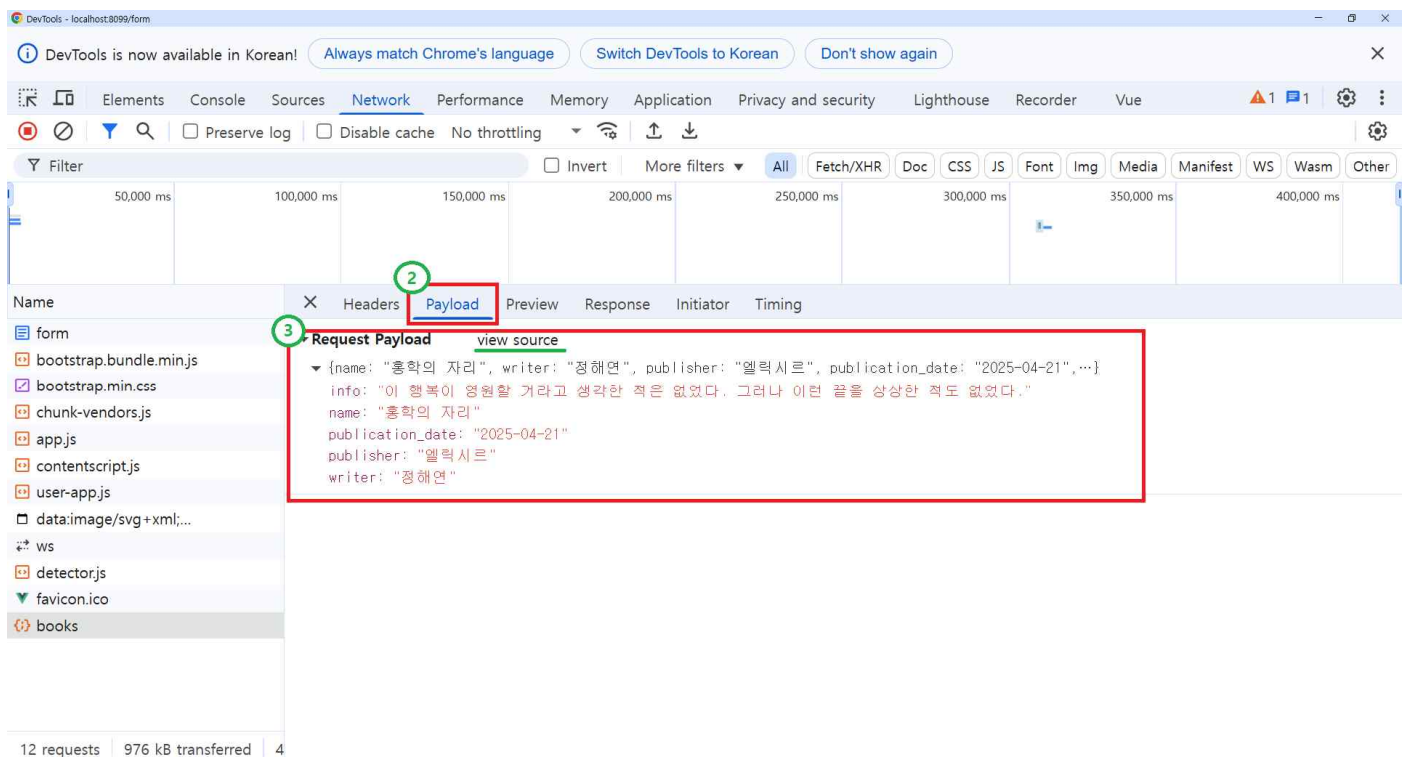
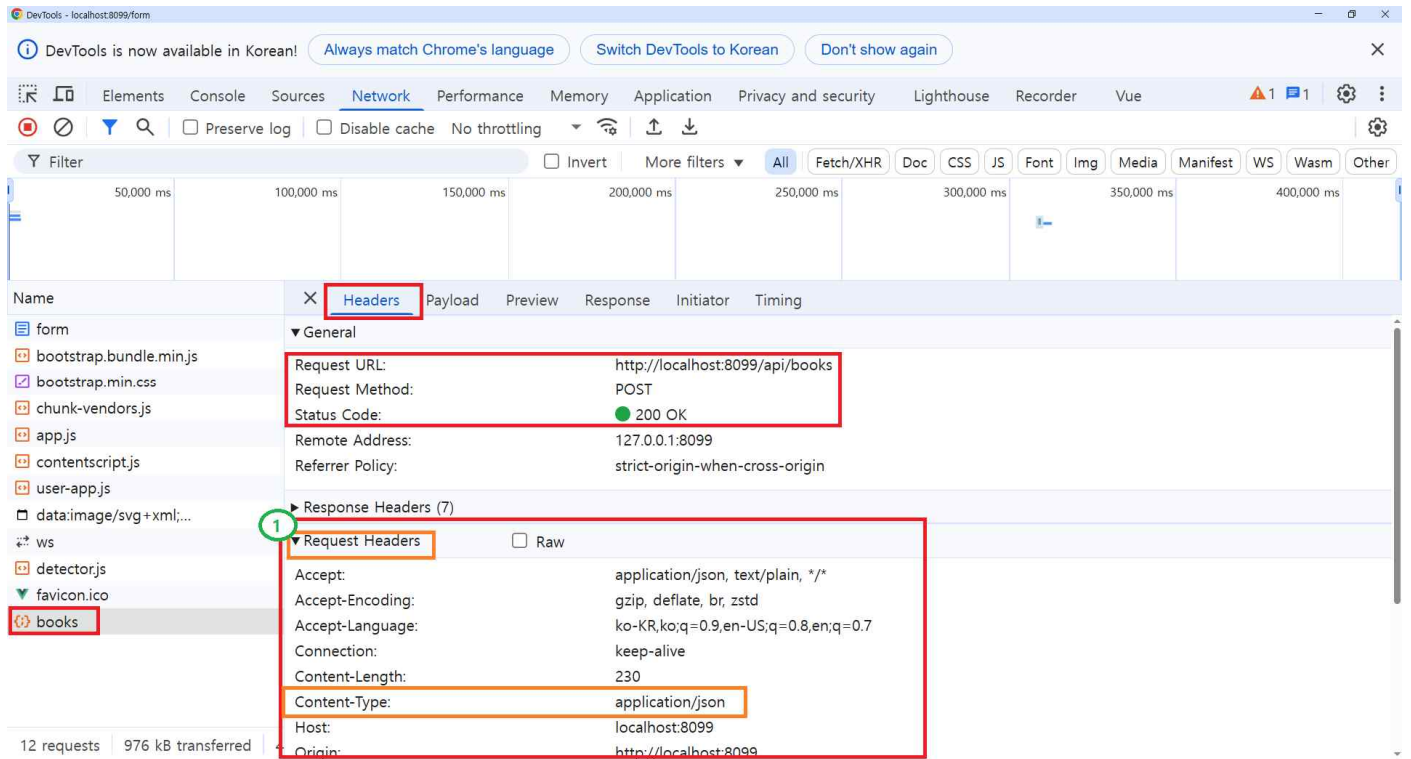
-3) 사용자가 저장 버튼을 눌러 서버에 요청 후 결과를 받음

AJAX(axios 모듈 실행결과)를 통해 실제 Server에 통신한 결과 확인

1) Request(요청)

1. Body를 통해서 데이터를 전달하는 경우 Header에 Content-type(데이터 포맷)이 존재
(application/json : JSON 포맷을 의미)
2. Payload 탭
 - Method 및 Content-type 상관없이 보내는 데이터가 있는 경우에만 생김
 - 보내는 데이터가 있는 데 해당 탭이 없는 경우 Javascript 코드를 확인
3. 통신을 통해 실제로 보내는 데이터를 의미하며 보통 parse된 데이터를 보여줌.
view source를 누를 경우 parse 되기 전 실제 데이터를 보여줌. => 4번 이미지

4. Body에 실제로 작성된 형태로 JSON 포맷으로 작성된 것을 확인 할 수 있음.
view parsed를 누를 경우 parse된 데이터를 보여줌. => 3번 이미지



DevTools - localhost:8099/form

DevTools is now available in Korean! [Always match Chrome's language](#) [Switch DevTools to Korean](#) [Don't show again](#)

Elements Console Sources **Network** Performance Memory Application Privacy and security Lighthouse Recorder Vue

Preserve log Disable cache No throttling

Filter 50,000 ms 100,000 ms 150,000 ms 200,000 ms 250,000 ms 300,000 ms 350,000 ms 400,000 ms

Name X Headers **Payload** Preview Response Initiator Timing

form

bootstrap.bundle.min.js

bootstrap.min.css

chunk-vendors.js

app.js

contentscript.js

user-app.js

data:image/svg+xml;...

ws

detector.js

favicon.ico

books

12 requests 976 kB transferred 4

4 Request Payload view parsed

```
{ "name": "홍학의 자리", "writer": "정해연", "publisher": "엘릭시르", "publication_date": "2025-04-21", "info": "이 행복이 영원할 거라고 생각한 적은 없었다. 그러나 이런 끝을 상상한 적도 없었다." }
```

2) Response(응답)

DevTools - localhost:8099/form

DevTools is now available in Korean! [Always match Chrome's language](#) [Switch DevTools to Korean](#) [Don't show again](#)

Elements Console Sources **Network** Performance Memory Application Privacy and security Lighthouse Recorder Vue

Preserve log Disable cache No throttling

Filter 50,000 ms 100,000 ms 150,000 ms 200,000 ms 250,000 ms 300,000 ms 350,000 ms 400,000 ms

Name X Headers Payload **Preview** Response Initiator Timing

form

bootstrap.bundle.min.js

bootstrap.min.css

chunk-vendors.js

app.js

contentscript.js

user-app.js

data:image/svg+xml;...

ws

detector.js

favicon.ico

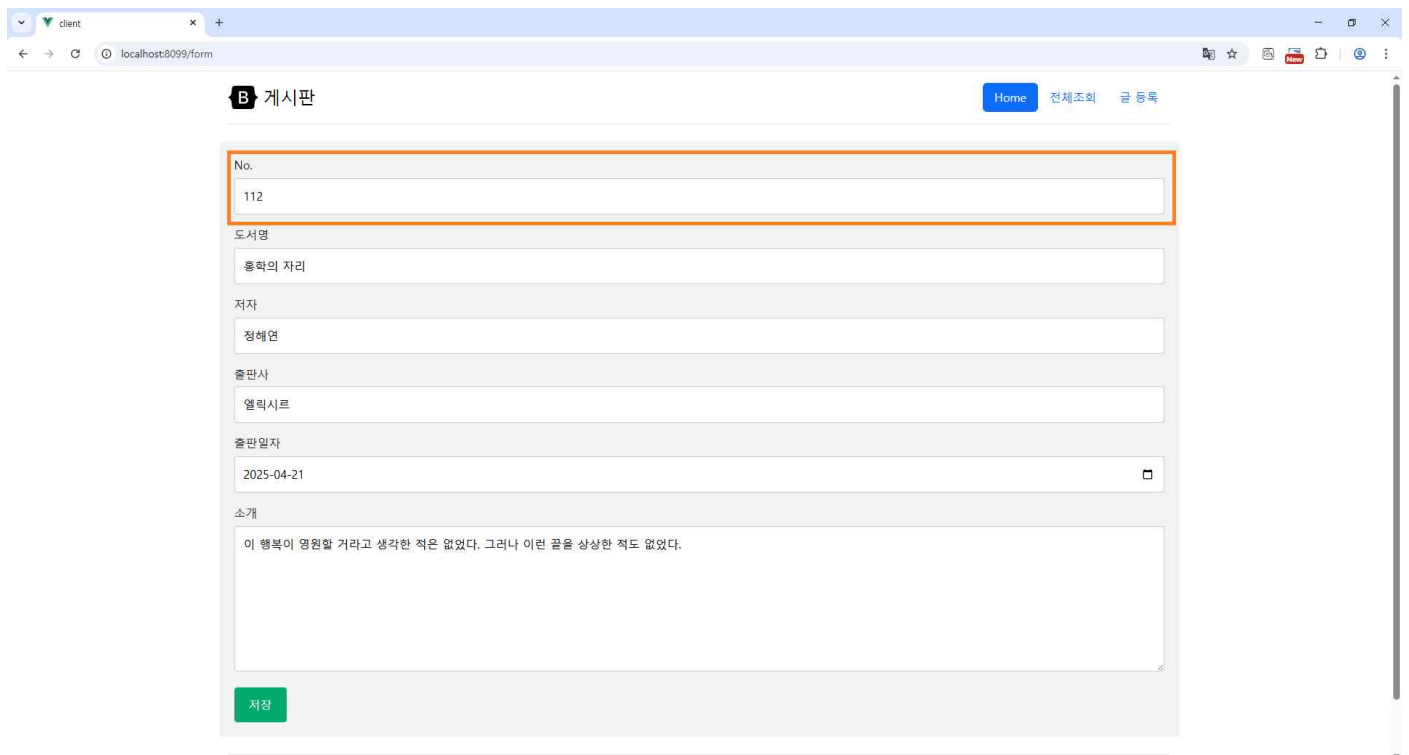
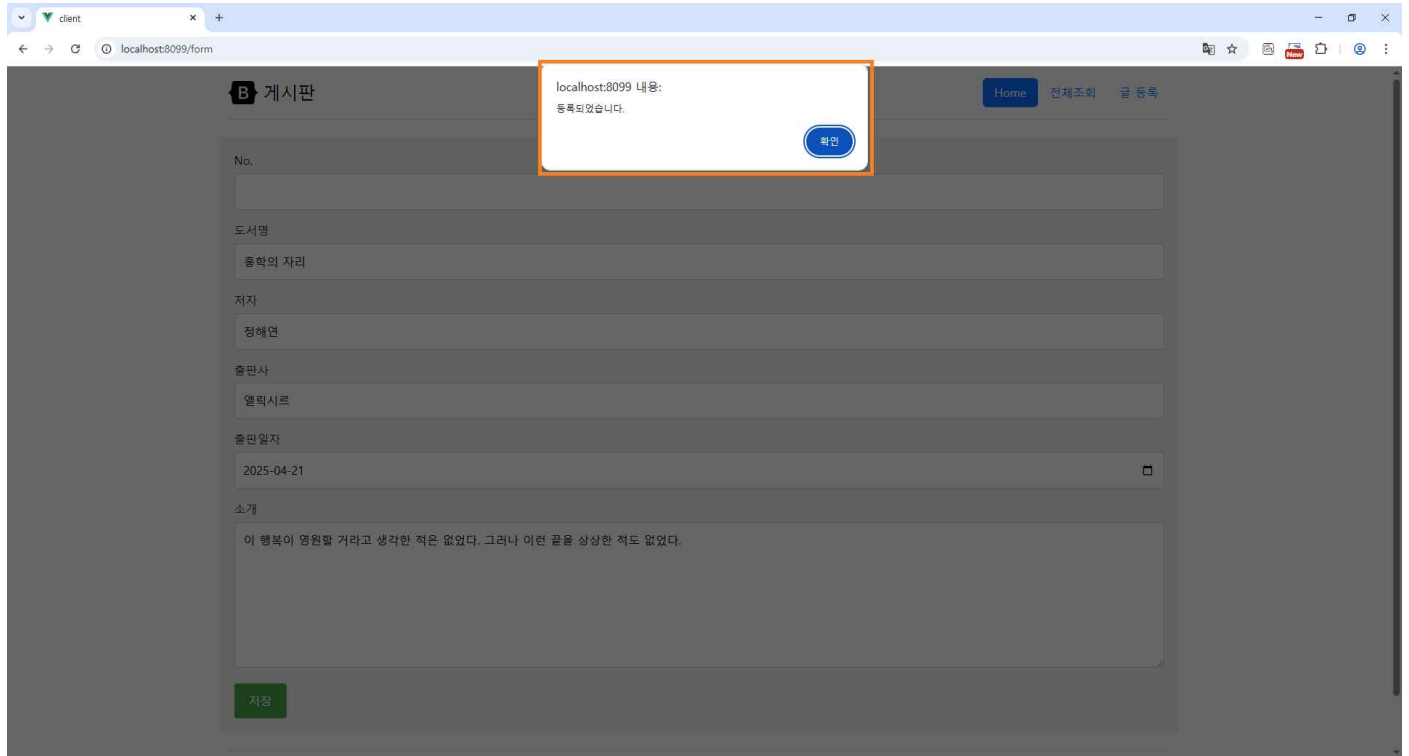
books

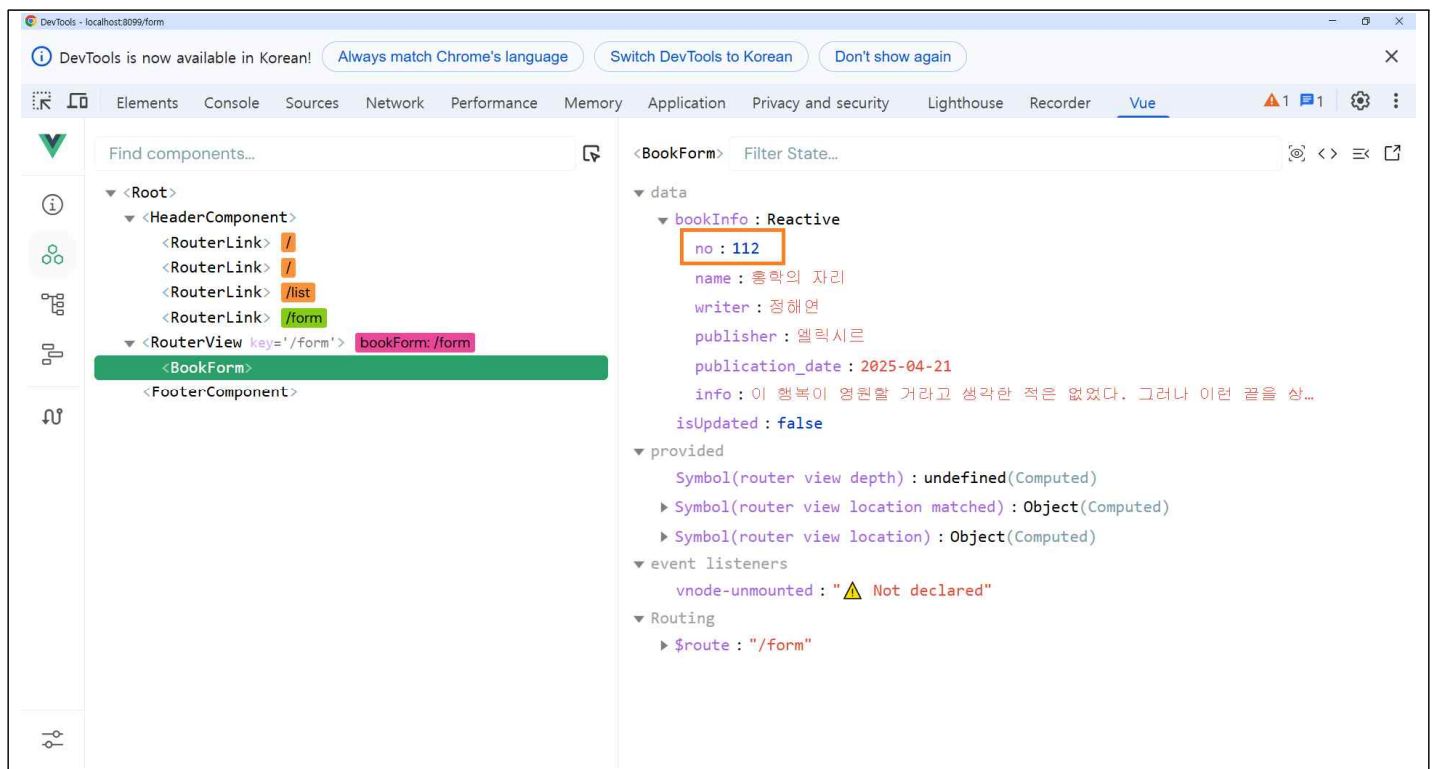
12 requests 976 kB transferred 4

▼ {isSucceeded: true, bookNo: 112}

```
bookNo: 112
isSucceeded: true
```

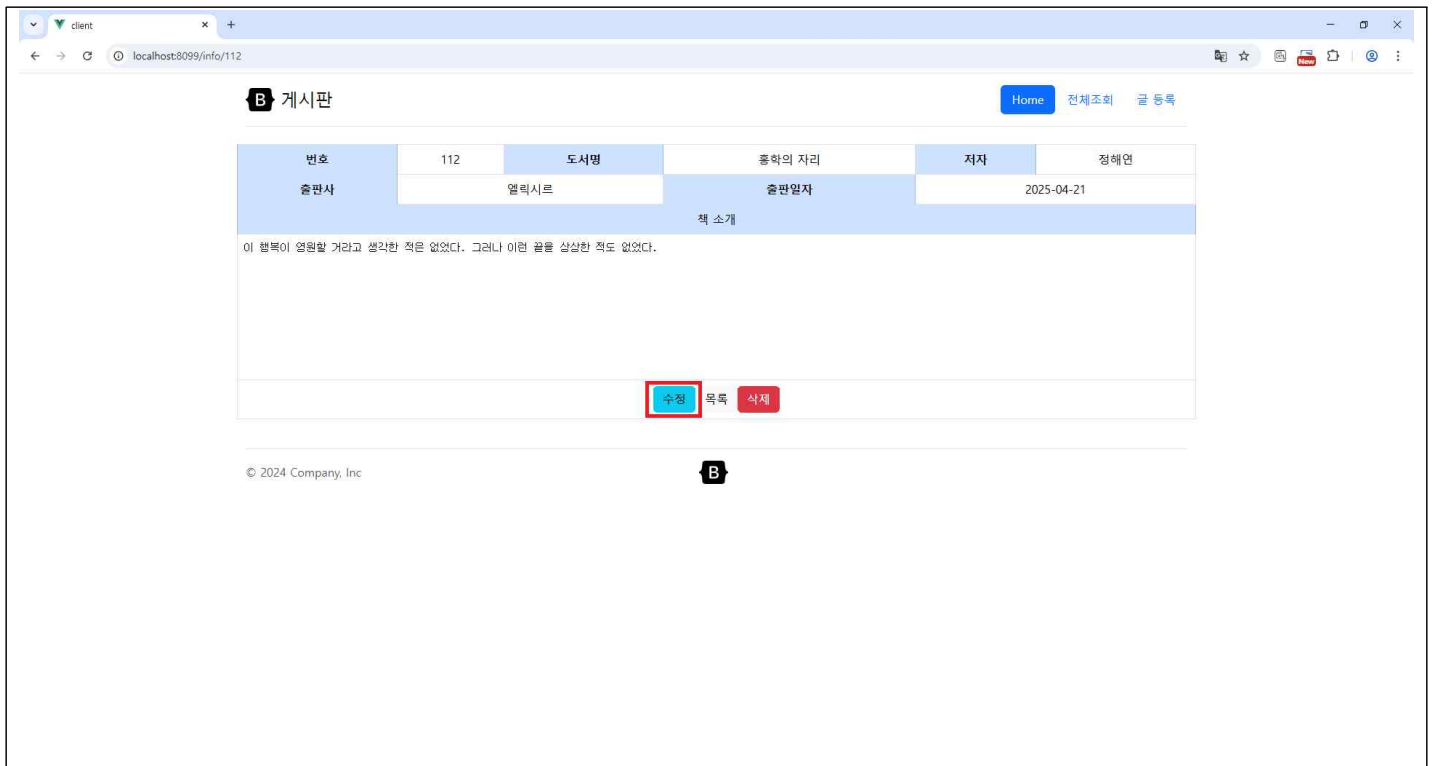
- 4) 서버가 응답한 결과를 기반으로 화면에서 처리



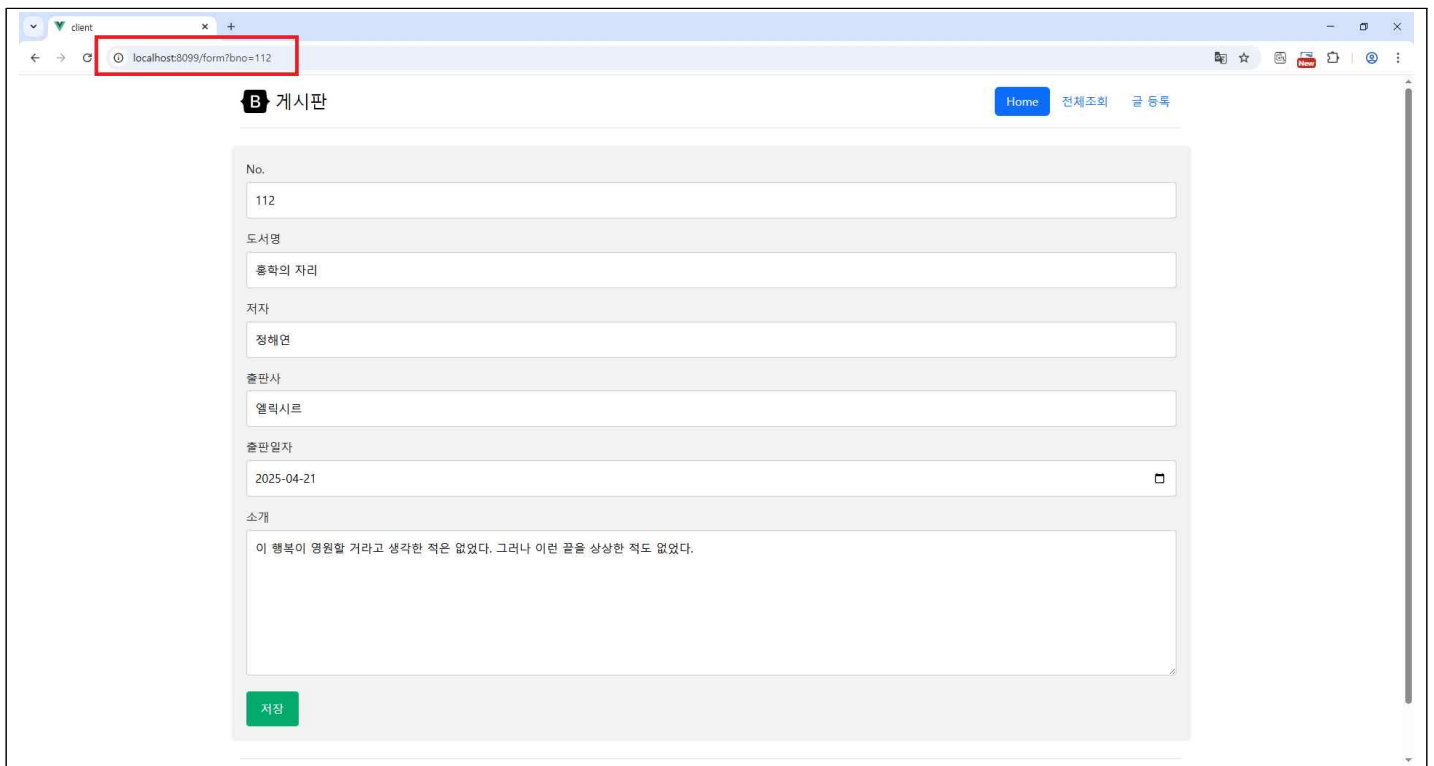


8.5. 수정

- 1) 단건조회에서 수정 버튼을 선택



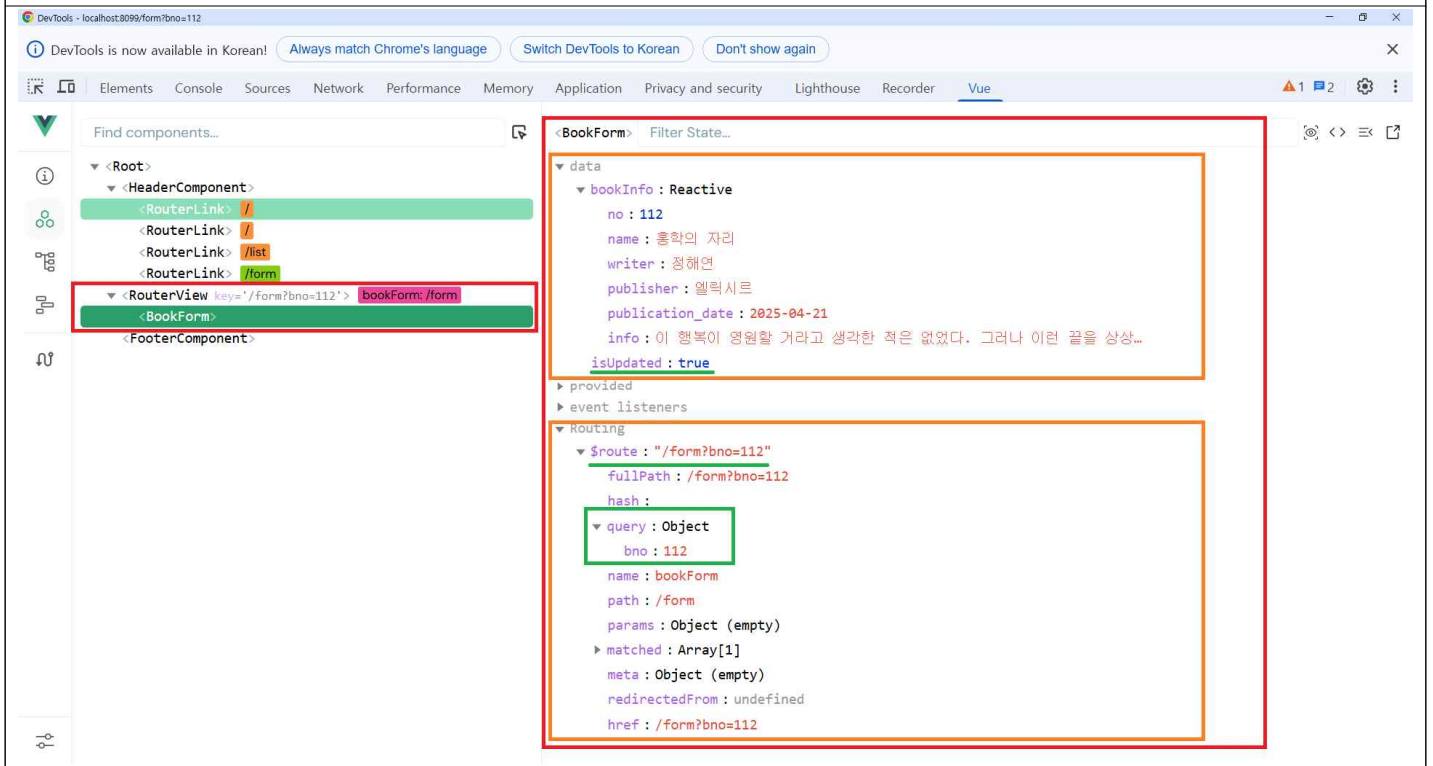
- 2) 수정 페이지를 출력



Vue devtools

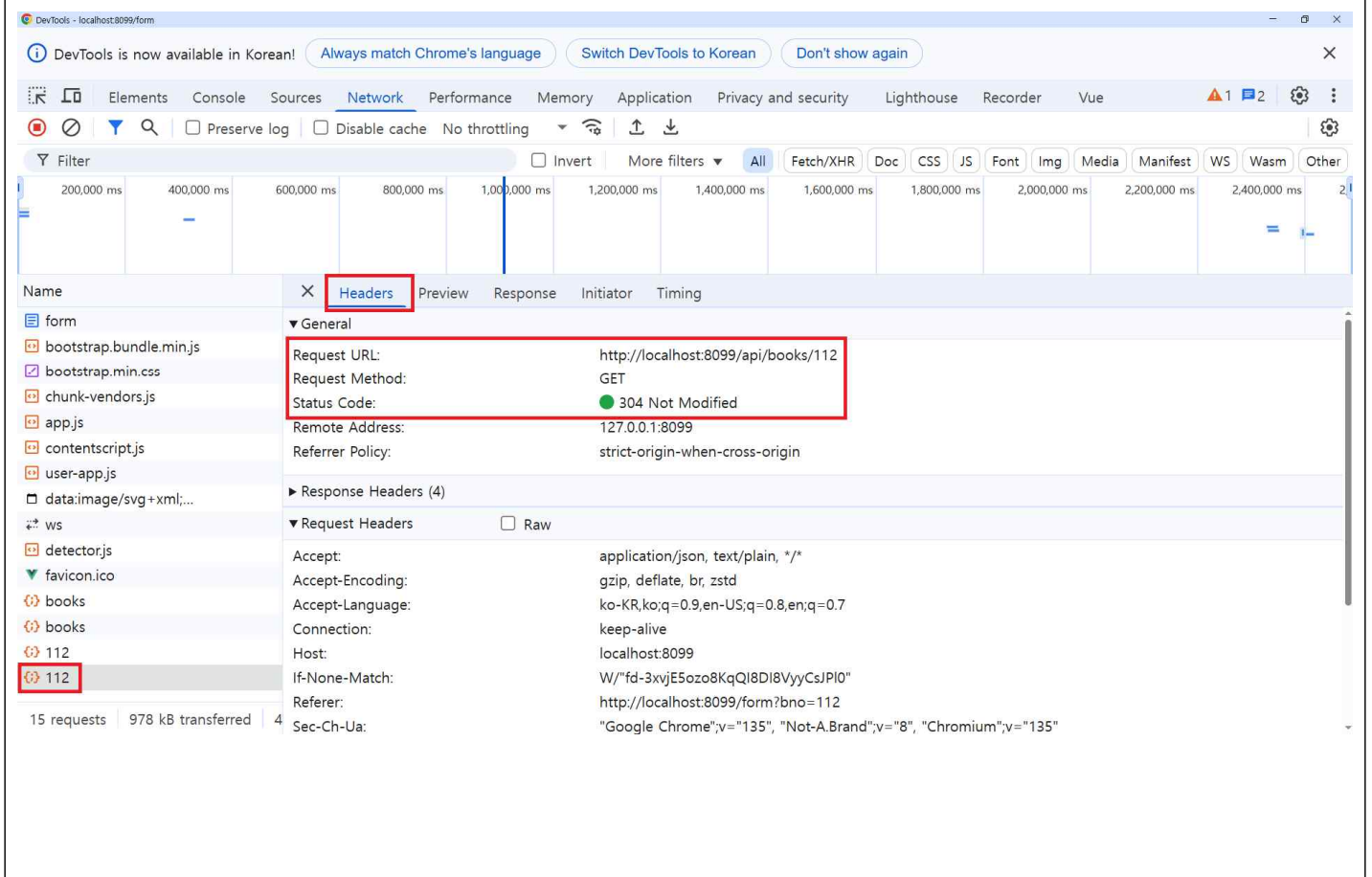
- BookInfo.vue에서 실제 요청 코드

```
this.$router.push({ name: 'bookForm', query: { bno : this.bookInfo.no } });
```

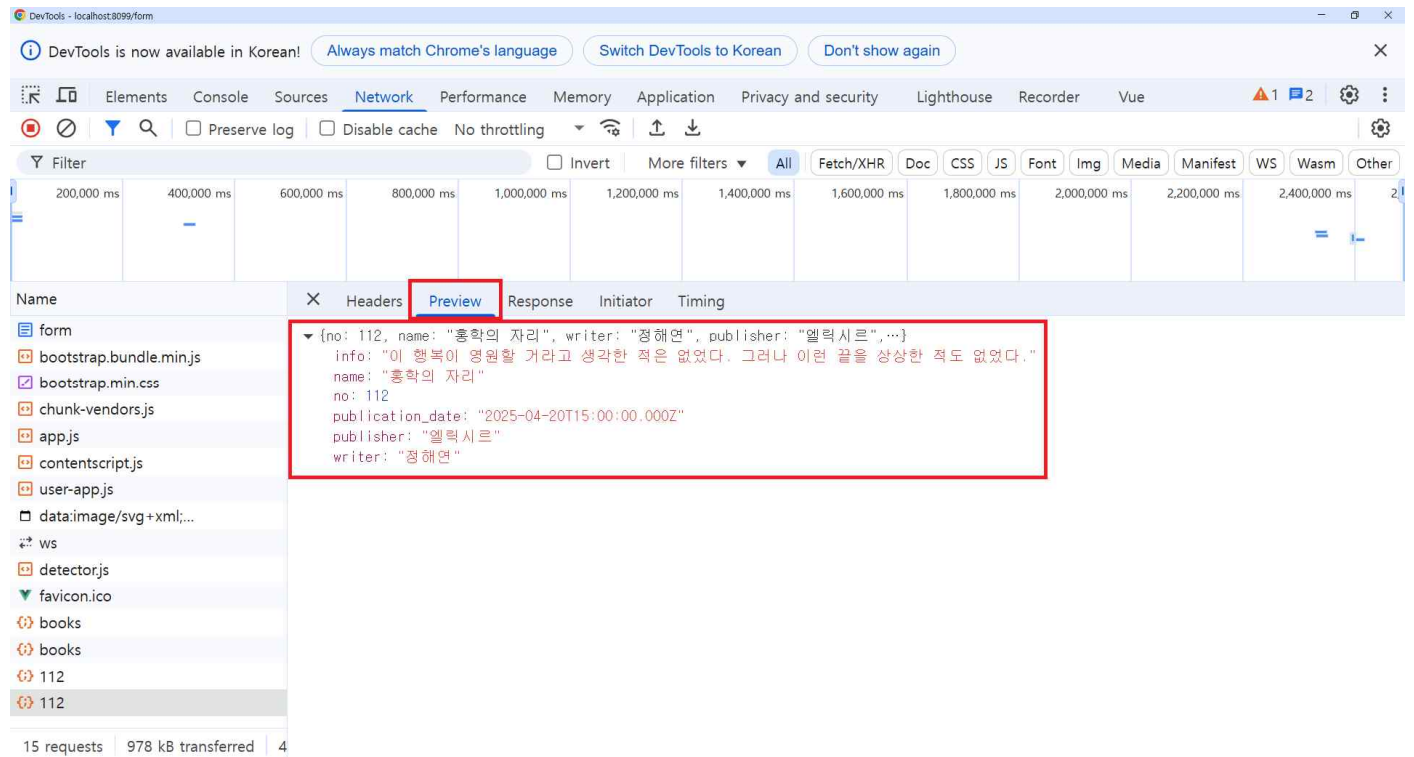


- 3) 사용자가 선택한 데이터를 서버에 요청 후 결과를 받음

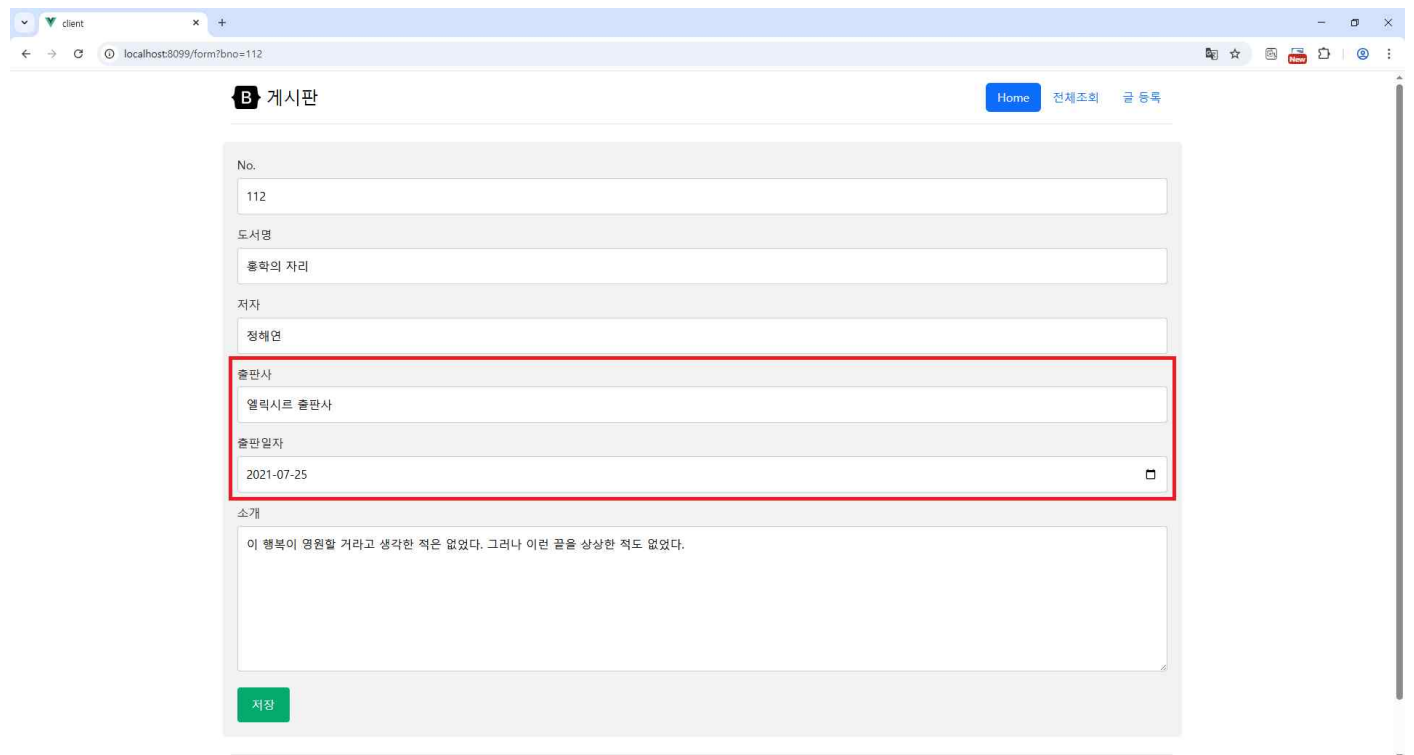
AJAX(axios 모듈 실행결과)를 통해 실제 Server에 통신한 결과 확인

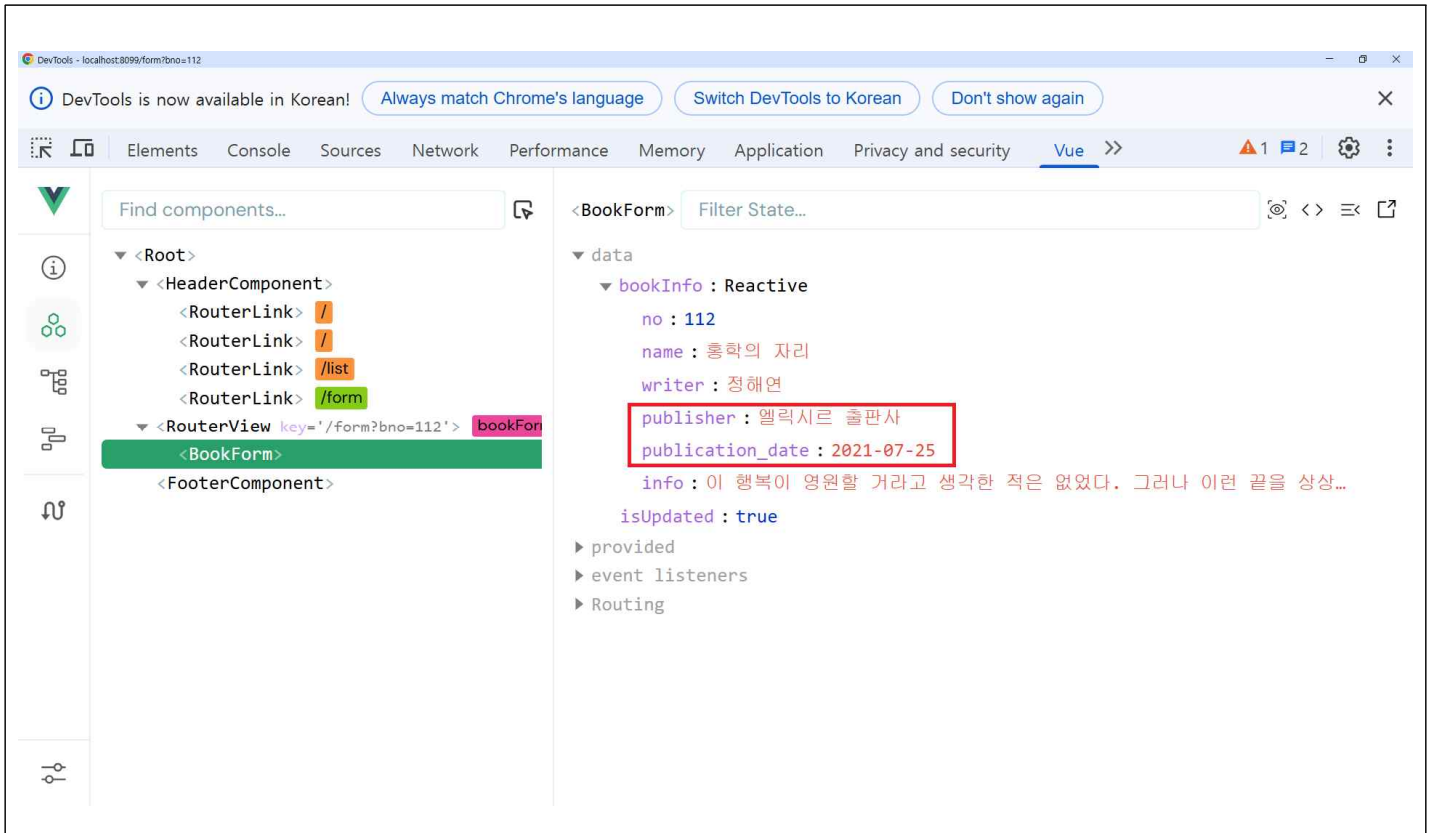


2) Response(응답)



- 4) 사용자가 수정할 데이터를 입력

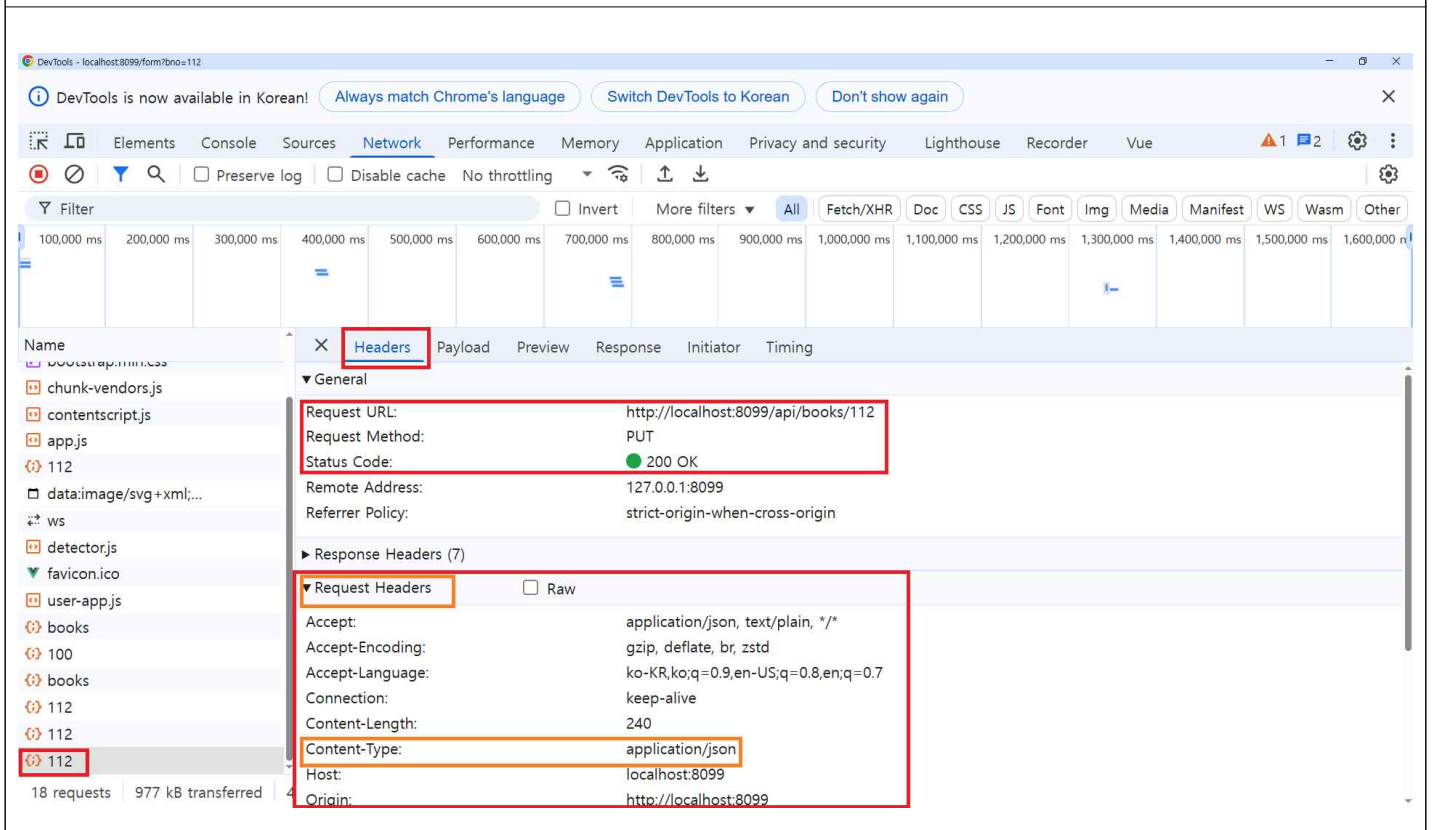




- 5) 사용자가 저장 버튼을 눌러 서버에 요청 후 결과를 받음

AJAX(axios 모듈 실행결과)를 통해 실제 Server에 통신한 결과 확인

1) Request(요청)



DevTools - localhost:8099/form?bno=112

DevTools is now available in Korean! [Always match Chrome's language](#) [Switch DevTools to Korean](#) [Don't show again](#)

Elements Console Sources **Network** Performance Memory Application Privacy and security Lighthouse Recorder Vue

Preserve log Disable cache No throttling

Filter Invert More filters All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other

100,000 ms 200,000 ms 300,000 ms 400,000 ms 500,000 ms 600,000 ms 700,000 ms 800,000 ms 900,000 ms 1,000,000 ms 1,100,000 ms 1,200,000 ms 1,300,000 ms 1,400,000 ms 1,500,000 ms 1,600,000 ms

Name

- bootstrap.min.css
- chunk-vendors.js
- contentscript.js
- app.js
- 112
- data:image/svg+xml;...
- ws
- detector.js
- favicon.ico
- user-app.js
- books
- 100
- books
- 112
- 112
- 112

18 requests 977 kB transferred 4

X Headers **Payload** Preview Response Initiator Timing

Request Payload view source

```
{name: "홍학의 자리", writer: "정해연", publisher: "엘릭시르 출판사", publication_date: "2021-07-25", ...}
info: "이 행복이 영원할 거라고 생각한 적은 없었다. 그러나 이런 끝을 상상한 적도 없었다."
name: "홍학의 자리"
publication_date: "2021-07-25"
publisher: "엘릭시르 출판사"
writer: "정해연"
```

DevTools - localhost:8099/form?bno=112

DevTools is now available in Korean! [Always match Chrome's language](#) [Switch DevTools to Korean](#) [Don't show again](#)

Elements Console Sources **Network** Performance Memory Application Privacy and security Lighthouse Recorder Vue

Preserve log Disable cache No throttling

Filter Invert More filters All Fetch/XHR Doc CSS JS Font Img Media Manifest WS Wasm Other

100,000 ms 200,000 ms 300,000 ms 400,000 ms 500,000 ms 600,000 ms 700,000 ms 800,000 ms 900,000 ms 1,000,000 ms 1,100,000 ms 1,200,000 ms 1,300,000 ms 1,400,000 ms 1,500,000 ms 1,600,000 ms

Name

- bootstrap.min.css
- chunk-vendors.js
- contentscript.js
- app.js
- 112
- data:image/svg+xml;...
- ws
- detector.js
- favicon.ico
- user-app.js
- books
- 100
- books
- 112
- 112
- 112

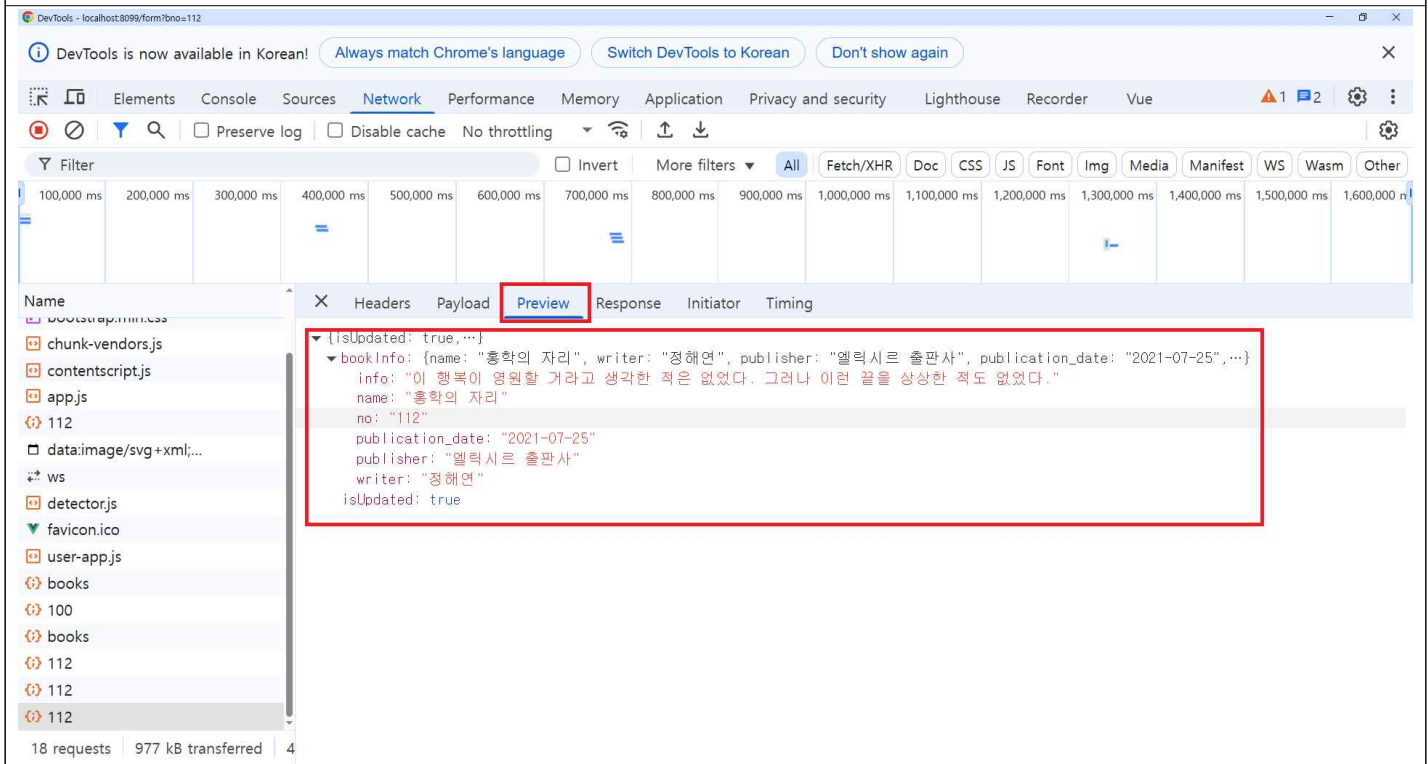
18 requests 977 kB transferred 4

X Headers **Payload** Preview Response Initiator Timing

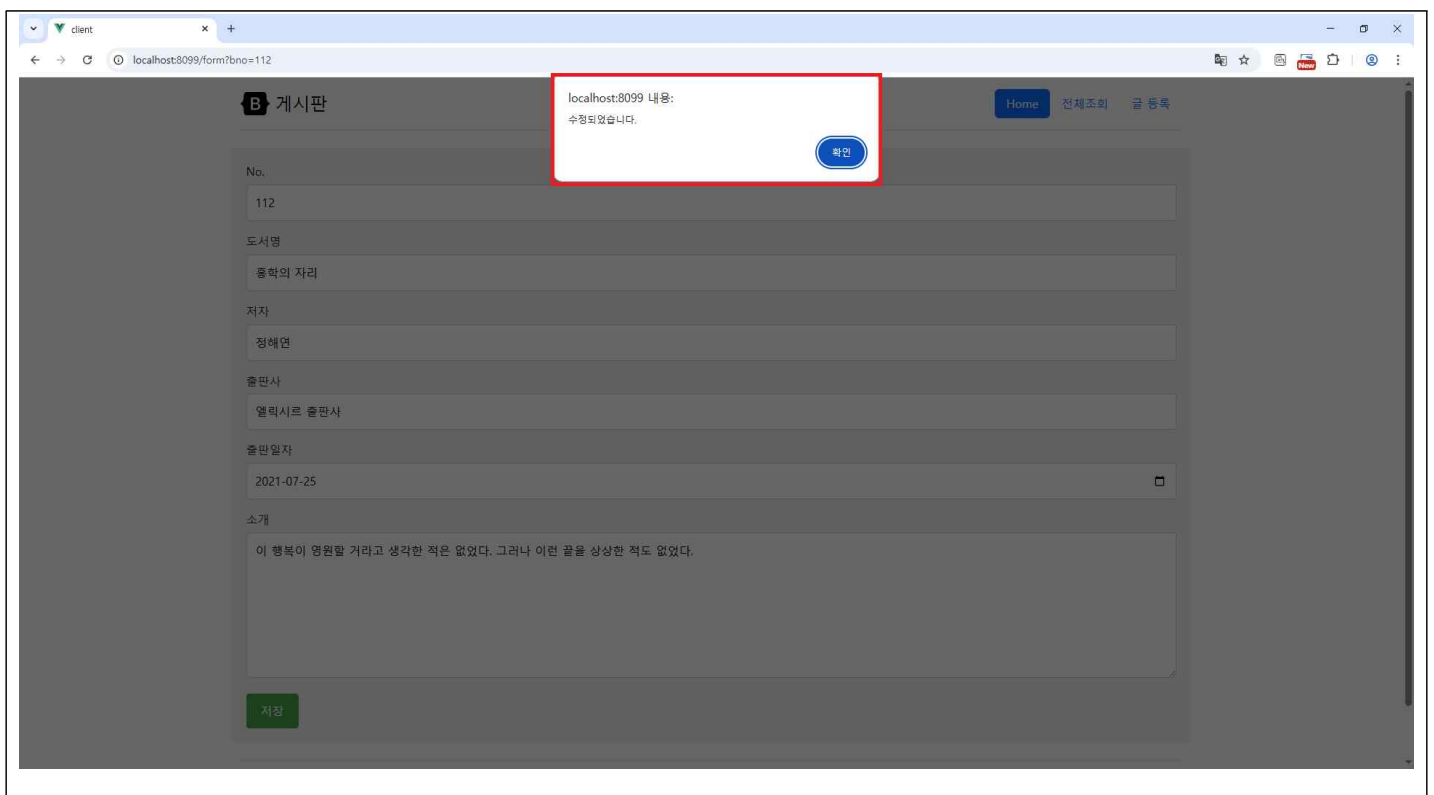
Request Payload view parsed

```
{"name": "홍학의 자리", "writer": "정해연", "publisher": "엘릭시르 출판사", "publication_date": "2021-07-25", "info": "이 행복이 영원할 거라고 생각한 적은 없었다. 그러나 이런 끝을 상상한 적도 없었다."}
```

2) Response(응답)



- 6) 서버가 응답한 결과를 기반으로 화면에서 처리



client

localhost8099/form?bno=112

Home전체조회글 등록

B게시판

No.

112

도서명

통학의 자리

저자

정해연

출판사

엘릭시르 출판사

출판일자

2021-07-25

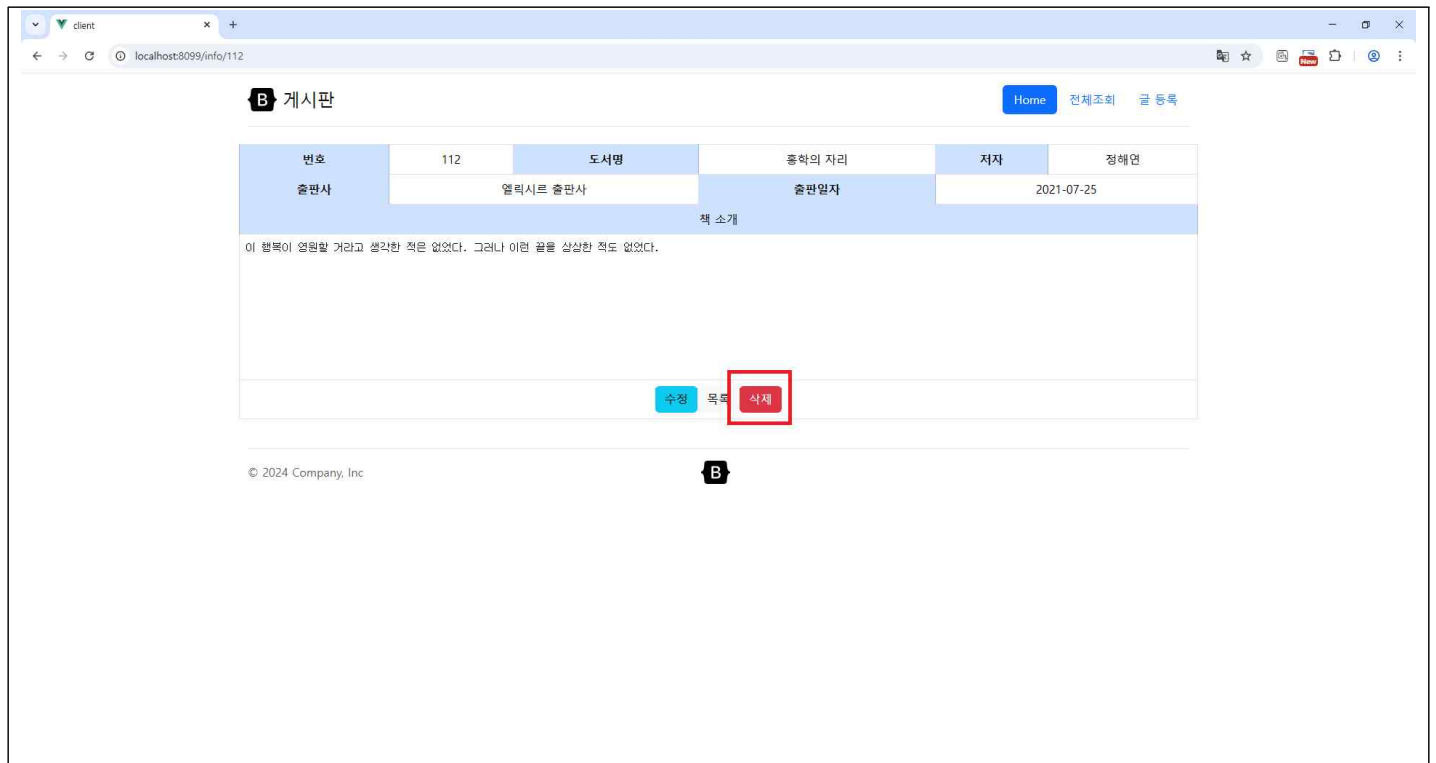
소개

이 행복이 영원할 거라고 생각한 적은 없었다. 그러나 이런 끝을 상상한 적도 없었다.

저장

8.6. 삭제

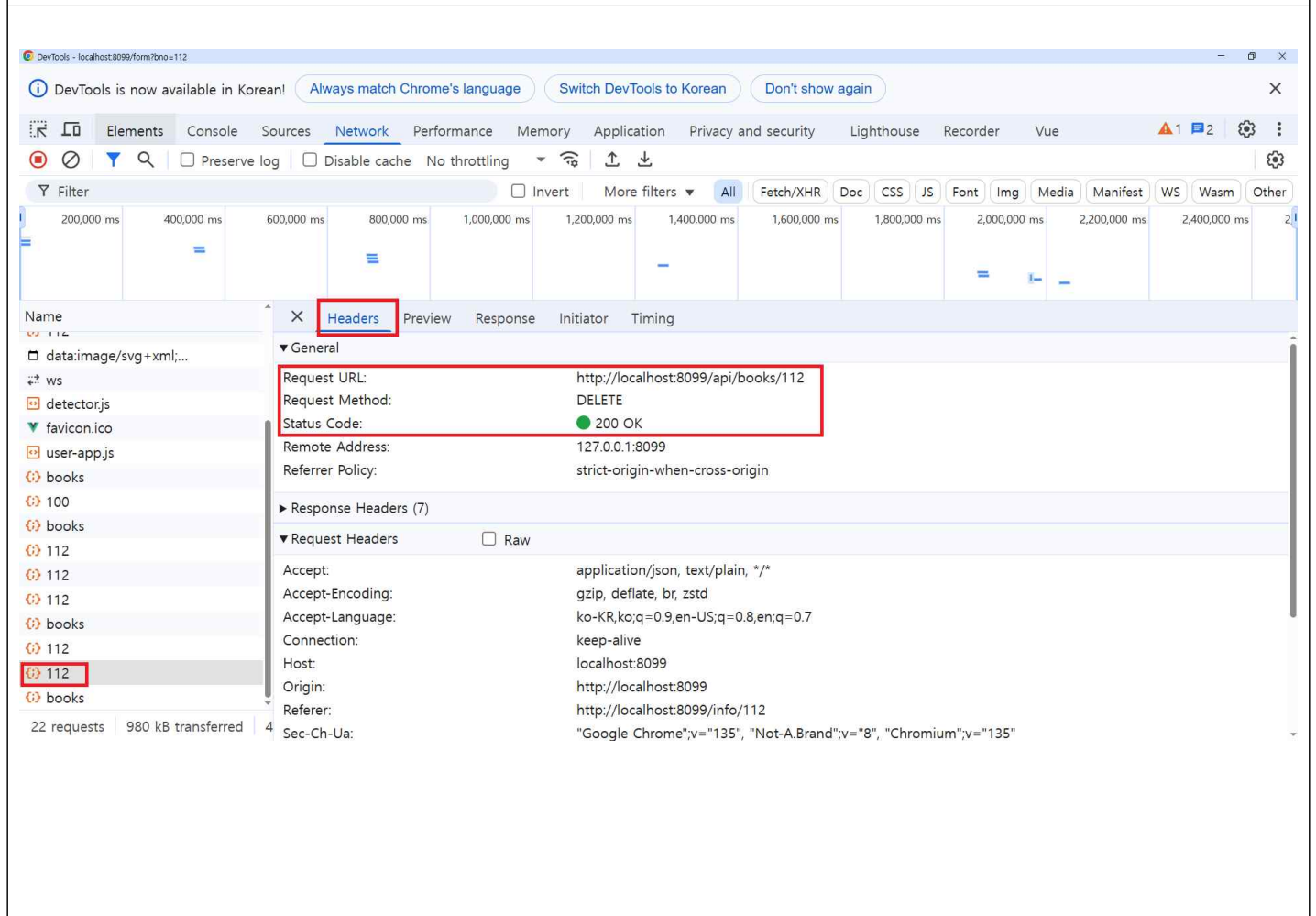
- 1) 단건조회에서 삭제 버튼을 선택



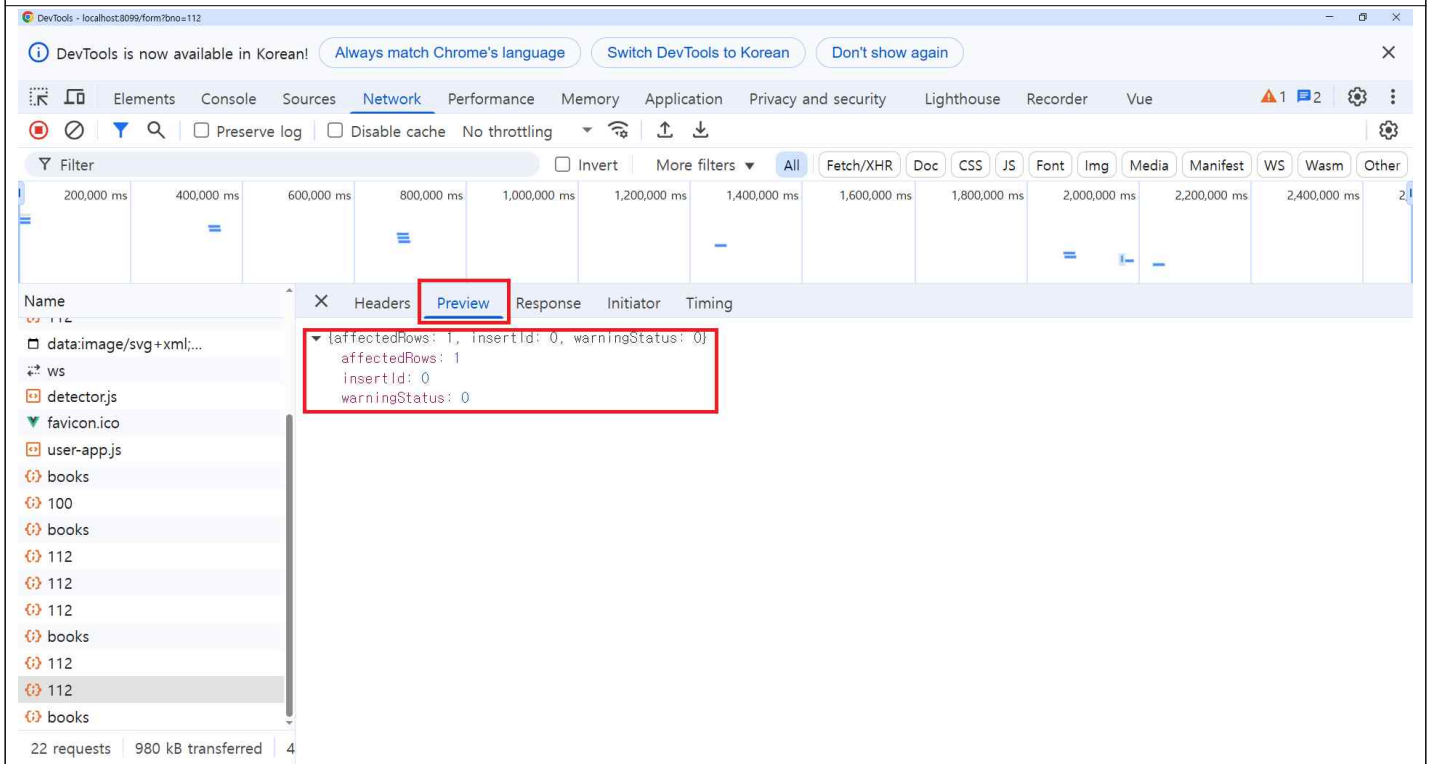
- 2) 선택한 대상에 대해 서버에 삭제를 요청

AJAX(axios 모듈 실행결과)를 통해 실제 Server에 통신한 결과 확인

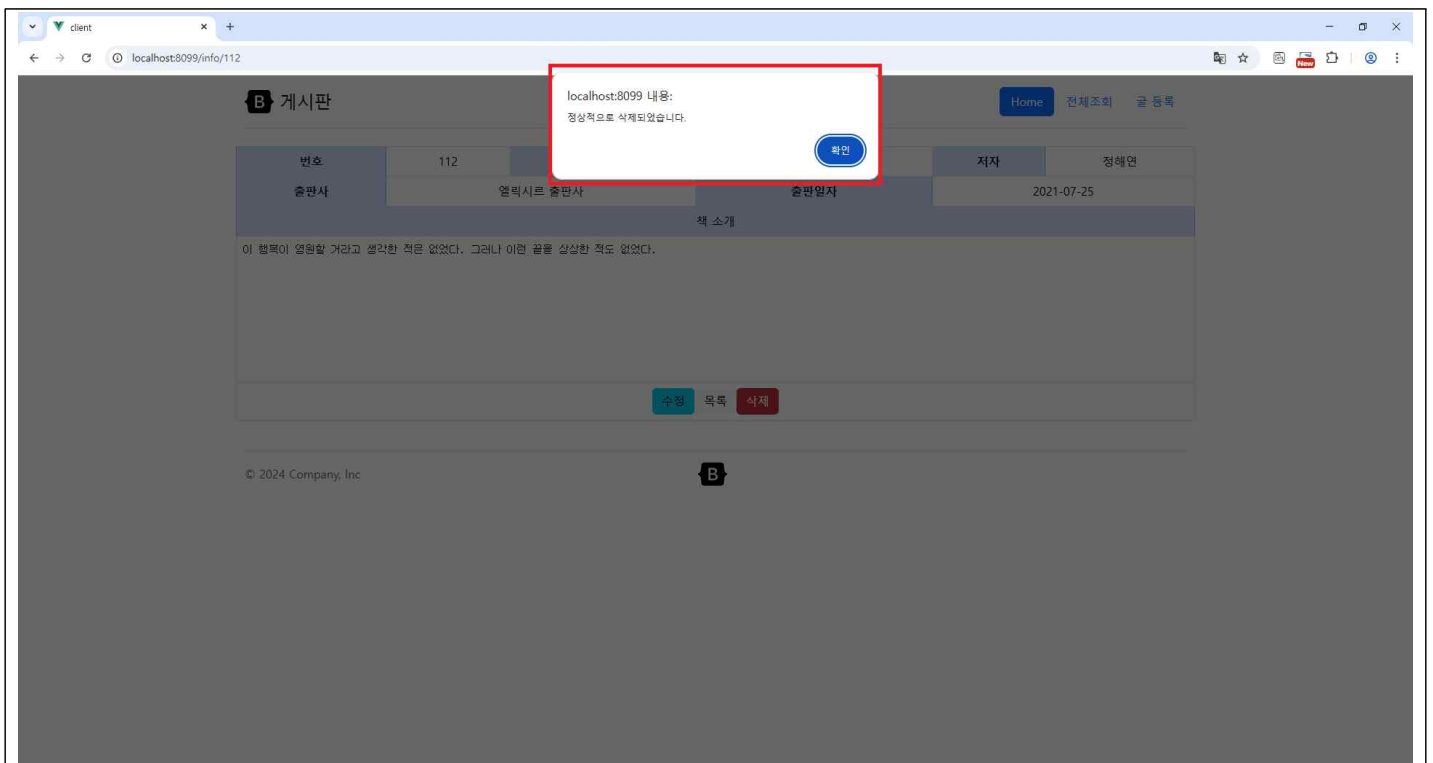
1) Request(요청)



2) Response(응답)



- 3) 서버가 응답한 결과를 기반으로 화면에서 처리



client

localhost:8099/list

🏠

☆

🔍

📄

📁

🔒

⋮

B

게시판

Home전체조회글 등록

No.	도서명	저자	출판사	출판일수
100	아침 그리고 저녁	윤 포세	문학동네	2019-09-26
101	그대를 어떻게 살 것인가	요시노 겐자부로	양철북	2012-06-28
102	모순	양귀자	쓰다	2013-04-01
103	아주 작은 습관의 힘	제임스 클리어	비즈니스북스	2019-02-26

total : 4

© 2024 Company, Inc

B