



| 학습주제

- ✓ Axios
- ✓ Vue-router

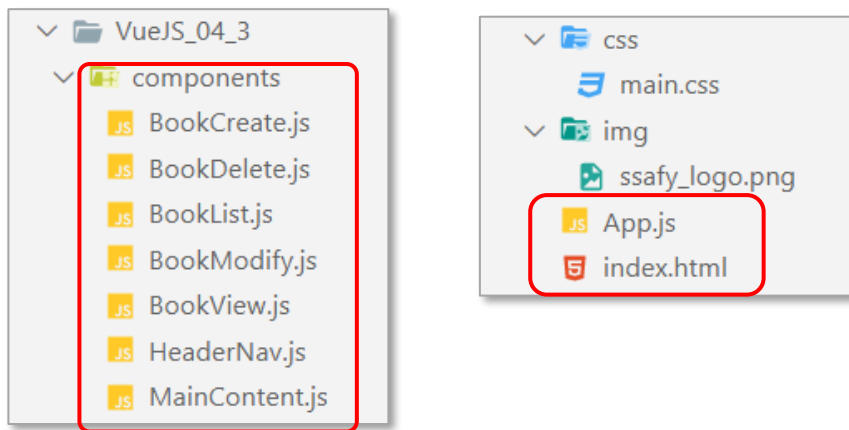
| 학습목표

- ✓ 도서정보를 localStorage 가 아닌 비동기 통신 방식인 Axios 를 활용하여 서버에서 관리한다.
- ✓ vue-router 를 이용하여 페이지를 전환한다.

| 지문

1. 도서 정보를 관리하기 위한 애플리케이션을 VueJS(Axios & vue-router) 기반으로 만들고자 한다. 아래 내용을 만족하도록 처리하시오.

A. 아래의 그림과 같이 프로젝트 구조를 작성한다.



- B. components 폴더 하위의 js 파일에서 도서 정보를 관리하는 모든 기능은 localStorage 가 아닌 Axios 를 이용한 서버와의 비동기 통신을 이용한다.
- C. App.js 에서는 VueRouter 의 객체를 생성하고 각 기능별 routes 를 설정한다.

```
import BookList from "../components/BookList.js";
```

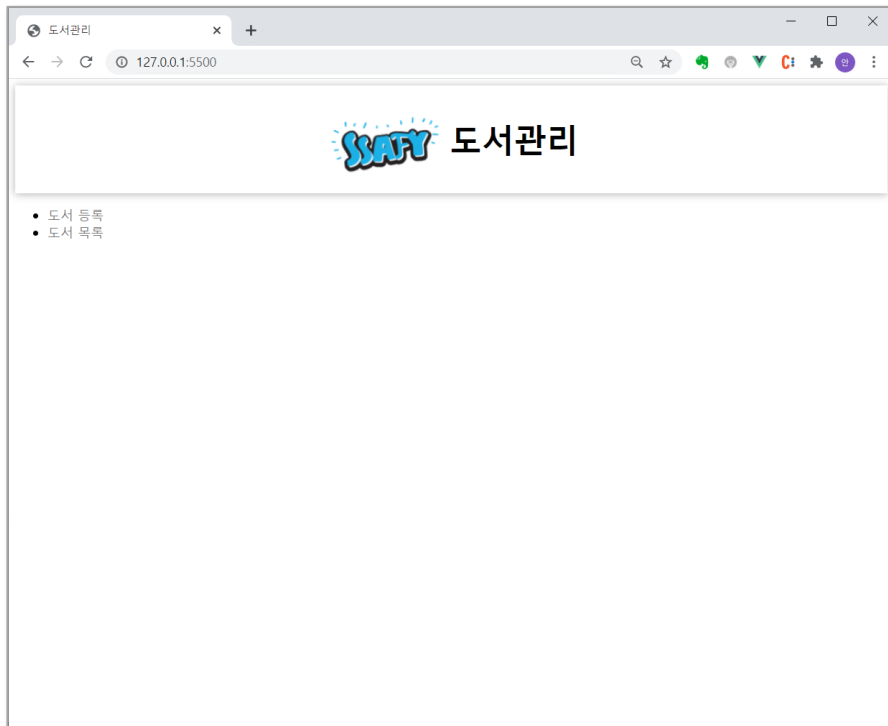
```
path: "/list",
name: "list",
component: BookList
```

- D. index.html 에서는 <router-view> tag 를 설정하여 현재 route 에 맞는 component 가 렌더링 되도록 한다.

```
<!-- 현재 route에 맞는 component가 렌더링된다. -->
<router-view></router-view>
```

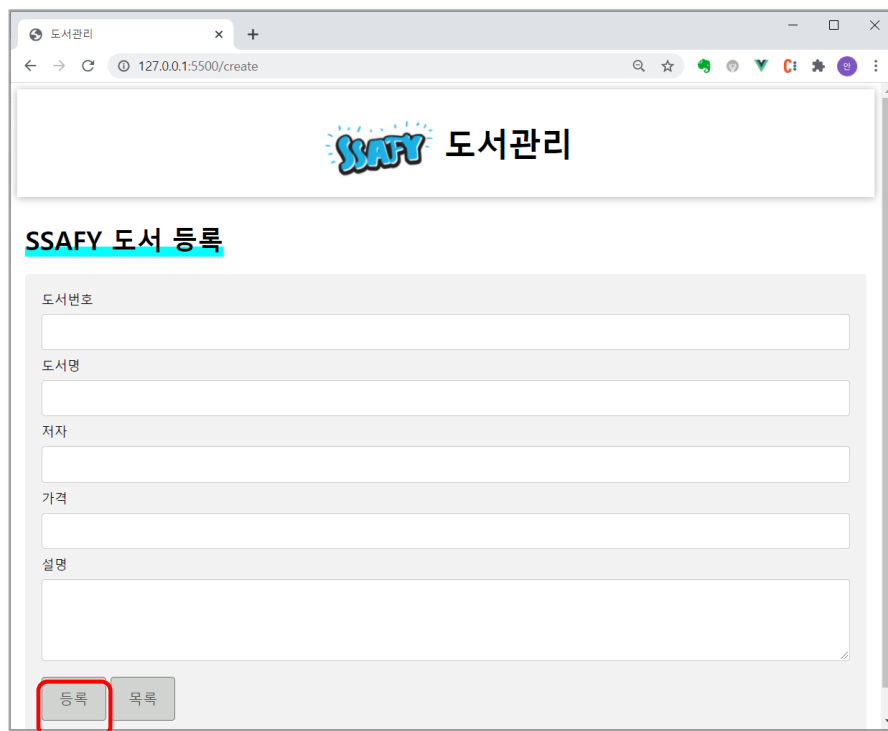


E. index.html 은 프로젝트 실행 시 첫번째 화면이다.



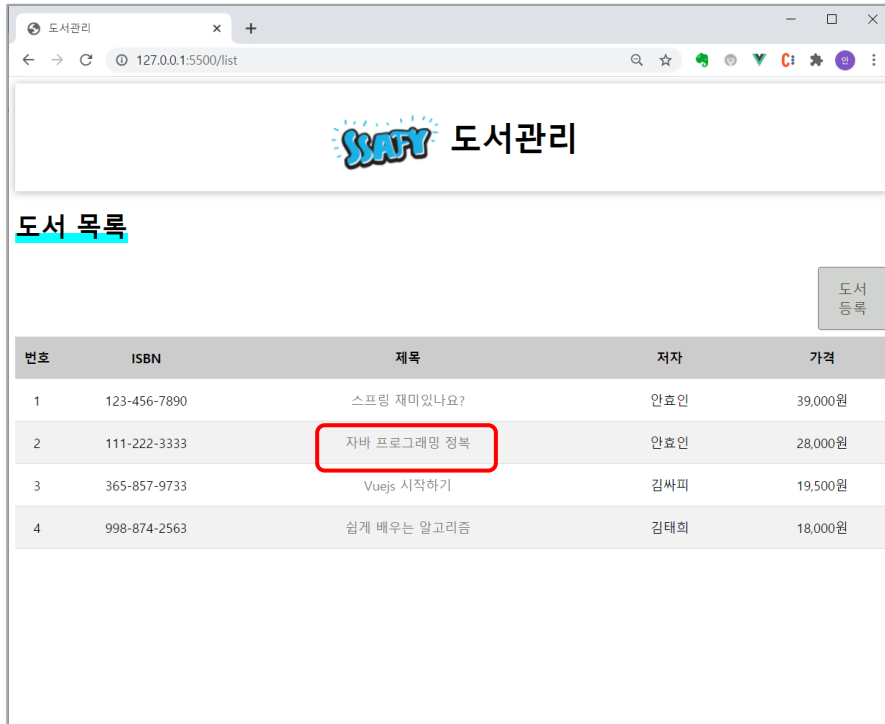
F. index.html 에서

i. 도서등록 클릭 시 화면 route : /create





ii.도서목록 클릭 시 화면 route : /list

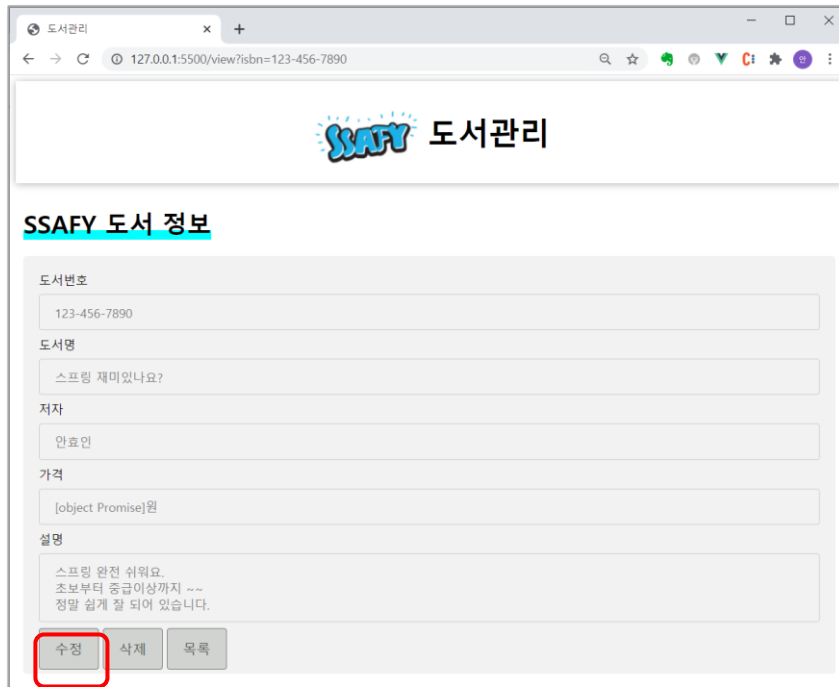


G. BookCreate.js 에서 도서 정보를 입력 후 등록 클릭 시 유효성 검사를 한 후 서버로 전송한다 (Axios 의 post 적용). 정상적으로 실행됐을 경우 /list 로 이동한다.

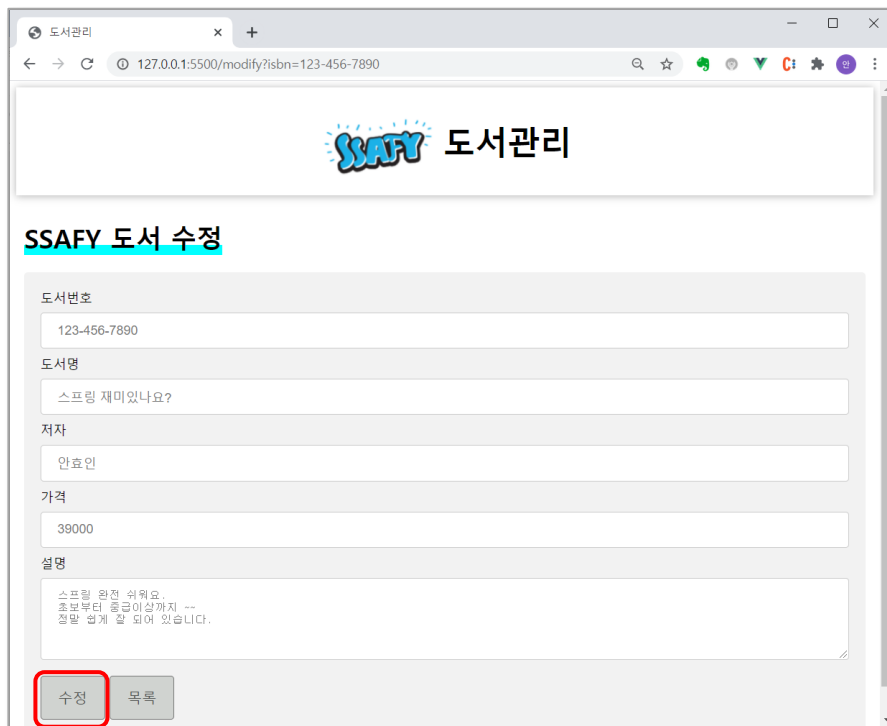
H. BookList.js 에서는 서버에서 모든 도서 목록을 얻어와 화면에 출력한다(Axios 의 get 적용).



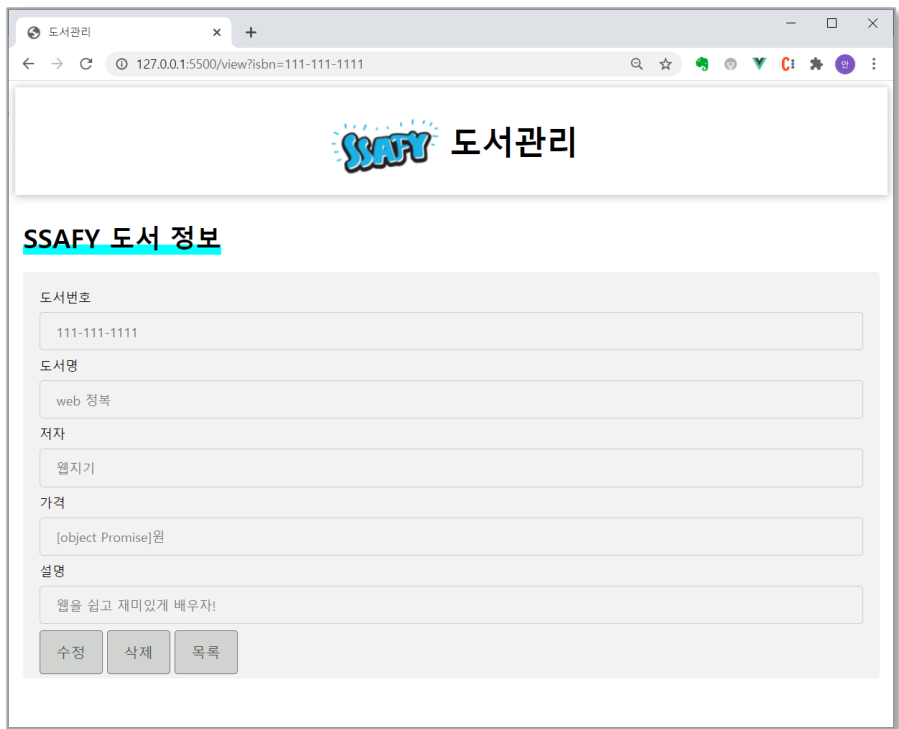
- I. 도서목록에서 각 도서의 제목을 클릭하면 해당 도서의 상세 정보 화면으로 전환된다.
(Axios의 get 적용) route : /view



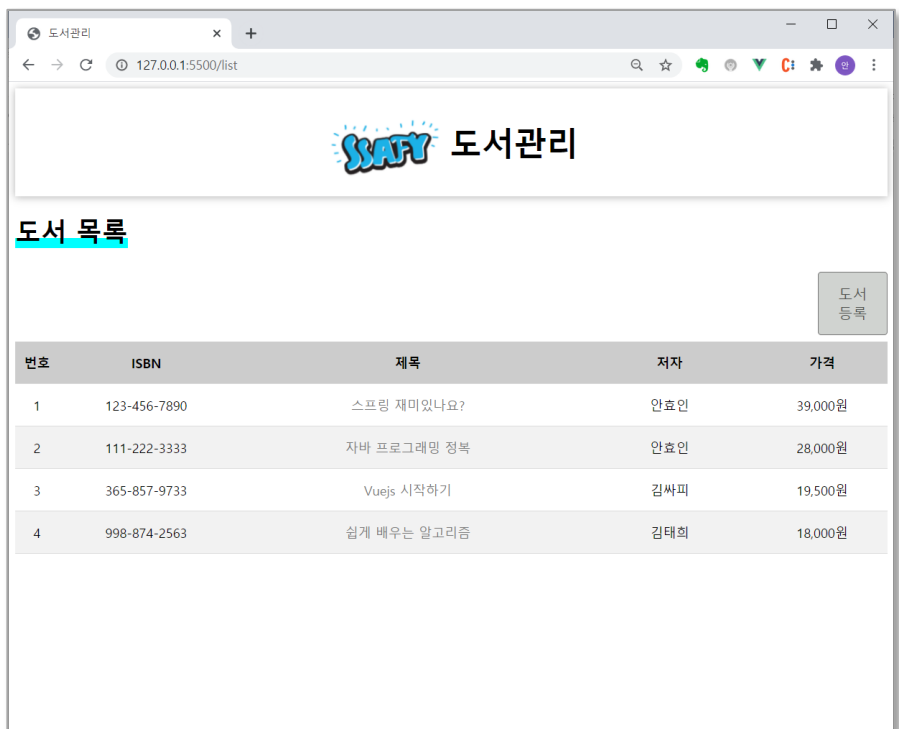
- J. 수정 버튼을 클릭하면 서버에서 ISBN에 해당하는 도서정보를 얻어와 수정화면으로 전환되고 정보가 출력된다. (Axios의 get 적용) route : /modify



- K. 수정페이지에서 도서정보를 수정한 후 수정 버튼 클릭 시 수정한 data를 서버에 전송한다.
(Axios의 put 적용) 정상적으로 실행됐을 경우 도서 목록 페이지로 이동.
- L. 삭제 버튼을 클릭 시 isbn을 서버로 보내고 정상적으로 삭제 시 도서 목록 페이지로 이동한다
(Axios의 delete 적용).



삭제 후 route : /list





| 자료 1: 소스코드 - BackEnd Rest API 제공 프로젝트

```

WS_VUE_BOOK_API [boot] [devtools]
├── src/main/java
│   ├── com.ssafy.book
│   │   ├── ServletInitializer.java
│   │   └── WsVueBookApiApplication.java
│   ├── com.ssafy.book.config
│   │   ├── DatabaseConfig.java
│   │   └── SwaggerConfig.java
│   ├── com.ssafy.book.controller
│   │   └── BookRestController.java
│   ├── com.ssafy.book.model
│   │   └── BookDto.java
│   └── com.ssafy.book.model.mapper
│       └── BookMapper.java

```

```

├── com.ssafy.book.model.service
│   ├── BookService.java
│   └── BookServiceImpl.java
├── src/main/resources
│   ├── mappers
│   │   └── book.xml
│   ├── static
│   │   └── ssafy.png
│   ├── templates
│   └── application.properties

```

Swagger REST API

<http://localhost:9999/vuews/swagger-ui.html>

