

삼성 청년 SW 아카데미

Vue.js

<알림>

본 강의는 삼성 청년 SW아카데미의 콘텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

Vue.js

목차

1. [axios](#)
2. [axios를 활용한 아파트 매매 정보](#)
3. [vue-router](#)

axios

삼성 청년 SW 아카데미

✓ HTTP 통신 : axios.

- Vue에서 권고하는 HTTP 통신 라이브러리는 axios이다.
- **Promise 기반의** HTTP 통신 라이브러리이며 상대적으로 다른 HTTP 통신 라이브러리들에 비해 문서화가 잘되어 있고 API가 다양하다.
- `axios.get(URL) << Promise 객체를 return >> then, catch` 사용가능.
- <https://github.com/axios/axios>

Promise란 서버에 데이터를 요청하여 받아오는 동작과 같은 비동기 로직 처리에 유용한 자바스크립트 라이브러리이다. 자바스크립트는 단일 스레드로 코드를 처리하기 때문에 특정 로직의 처리가 끝날 때까지 기다려 주지 않는다. 따라서 데이터를 요청하고 받아올 때까지 기다렸다가 화면에 나타내는 로직을 실행해야 할 때 주로 Promise를 활용한다. 그리고 데이터를 받아왔을 때 Promise로 데이터를 화면에 표시하거나 연산을 수행하는 등 특정 로직을 수행한다. **test12.html, test13.html 참고**

✓ axios 설치.

- CDN 방식.
 - `<script src="https://unpkg.com/axios/dist/axios.min.js"></script>`
- NPM 방식
 - `npm install axios`

✓ axios API.

- axios 대표 API.

API 유형	처리 결과
<code>axios.get('URL 주소').then().catch()</code>	해당 URL 주소에 대해 HTTP GET 요청을 보냄. 서버에서 보낸 데이터를 정상적으로 받아오면 <code>then()</code> 안에 정의된 로직이 실행되고, 데이터를 받아올 때 오류가 발생하면 <code>catch()</code> 에 정의한 로직이 실행.
<code>axios.post('URL 주소').then().catch()</code>	해당 URL 주소에 대해 HTTP POST 요청을 보냄. <code>then()</code> 과 <code>catch()</code> 는 <code>get</code> 방식과 동일.
<code>axios({옵션 속성})</code>	HTTP 요청에 대한 자세한 속성들을 직접 정의하여 보낼 수 있다. 데이터 요청을 보낼 URL, HTTP 요청방식, 보내는 데이터 유형 등..

<https://github.com/axios/axios#axios-api>

✓ axios API.

▪ axios(config)

```
// POST request
axios({
  method: 'post',
  url: '/user',
  data: {
    name: '홍길동',
    userid: 'ssafy'
  }
});
```

```
// GET request
axios({
  method: 'get',
  url: '/user/ssafy',
  responseType: 'json'
})
.then(function (response) {
  // logic
});
```

✓ axios API.

- `axios.get(url[, config])`

```
axios({  
  method: 'get',  
  url: '/board/10'  
});
```

```
axios('/board/10'); // axios() default >> get
```

```
axios.get('/board/10');
```

✓ axios API.

- `axios.post(url[, config])`

```
axios({  
  method: 'post',  
  url: '/board',  
  data: {userid: 'ssafy', subject: '안녕하세요'}  
});
```

```
axios.post('/board', {  
  userid: 'ssafy',  
  subject: '안녕하세요'  
});
```

✓ axios API.

- `axios.put(url[, config])`

```
axios({  
  method: 'put',  
  url: '/board/10',  
  data: {subject: '안녕하세요', content: '오늘도 즐거운 하루 되세요 ^^' }  
});
```

```
axios.put('/board/10', {  
  subject: '안녕하세요',  
  content: '오늘도 즐거운 하루 되세요 ^^'  
});
```

✓ axios API.

- `axios.delete(url[, config])`

```
axios({
  method: 'delete',
  url: '/board/10'
});

axios('/board/10', {
  method: 'delete'
});

axios.delete('/board/10');
```

이어서..

삼성 청년 SW 아카데미

axios를 활용한 아파트매매정보

삼성 청년 SW 아카데미

✓ axios를 이용한 아파트 매매정보 (1/2).

```
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
</head>
<body>
  <div id="app">
    <div id="result">
      <table>
        <tr>
          <th>번호</th>
          <th>아파트명</th>
          <th>층수</th>
          <th>법정동</th>
          <th>거래금액</th>
        </tr>
        <tr v-for="apt, i in aptList">
          <td>{{i + 1}}</td>
          <td>{{apt.아파트}}</td>
          <td>{{apt.층}}</td>
          <td>{{apt.법정동}}</td>
          <td>{{apt.거래금액}}</td>
        </tr>
      </table>
    </div>
  </div>
</body>
```

test14.html

```
<script>
  const serviceKey =
    'serviceKey';
  const addr =
    'http://openapi.molit.go.kr/OpenAPI_.....' + serviceKey;

  new Vue({
    el: '#app',
    data: {
      aptList: [],
    },
    created() {
      axios
        .get(addr)
        .then((response) => {
          //console.log(response.data.response.body.items.item);
          this.aptList = response.data.response.body.items.item;
        })
        .catch((error) => {
          console.dir(error);
        });
    },
  });
</script>
```


✓ axios를 이용한 아파트 매매정보 (2/2).

The screenshot shows a web browser window displaying a Vue.js application. The application features a table with 10 rows of apartment listings. The columns are: 번호 (Number), 아파트명 (Apartment Name), 층수 (Floors), 법정동 (Statutory District), 거래금액 (Transaction Amount), and others. The table lists various apartments like '광화문풍림스페이스본' and '롯데미도파광화문빌딩'.

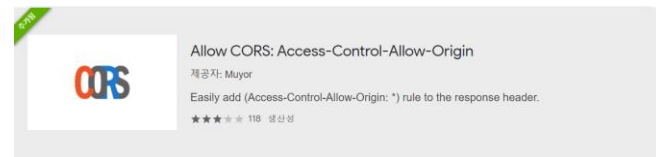
Below the table, the Vue DevTools component inspector is open, showing the component structure. The selected component is the root component, and its data is visible in the 'data' section. The data includes an array of apartment listings, with the first element selected and its details shown in the 'data' section.

The details of the selected apartment (index 0) are:

- 거래금액: 82,500
- 건축년도: 2008
- 년: 2015
- 도로명: "사직로8길"
- 도로명건물번호: "00004"
- 도로명건물부번호: "00000"
- 도로명주소: "11110"

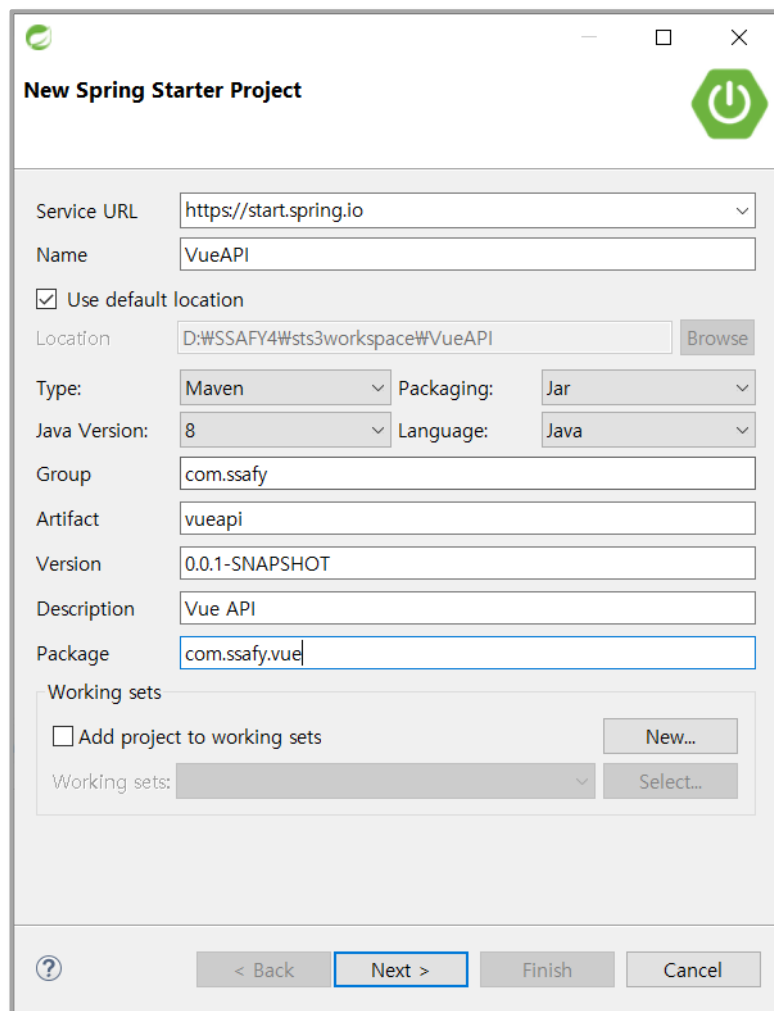
The console shows a message: "Access to XMLHttpRequest at 'http://openapi.molit.go.kr/OpenAPI_ToolInstallPackage/ test14.html:1 service/rest/RTMSOBSJS...6a07%2Bk3FV4baF9GXok1yxIEF%2BIwr2%2B%2F%2F4oVLT8bekKU%2Bk9ztkJ00wsBw%3D%3D' from origin 'http://127.0.0.1:5500' has been blocked by CORS policy: No 'Access-Control-Allow-Origin' header is present on the requested resource."

아래와 같이 에러가 날 경우 해결.
[크롬웹스토어](#)에서 설치 (테스트용)
 본질적 해결 >> CORS에 대해 공부!!!



✓ axios를 이용한 Spring REST API와의 통신.

- SpringBoot Project 생성.



New Spring Starter Project

Service URL:

Name:

☒ Use default location

Location:

Type: Packaging:

Java Version: Language:

Group:

Artifact:

Version:

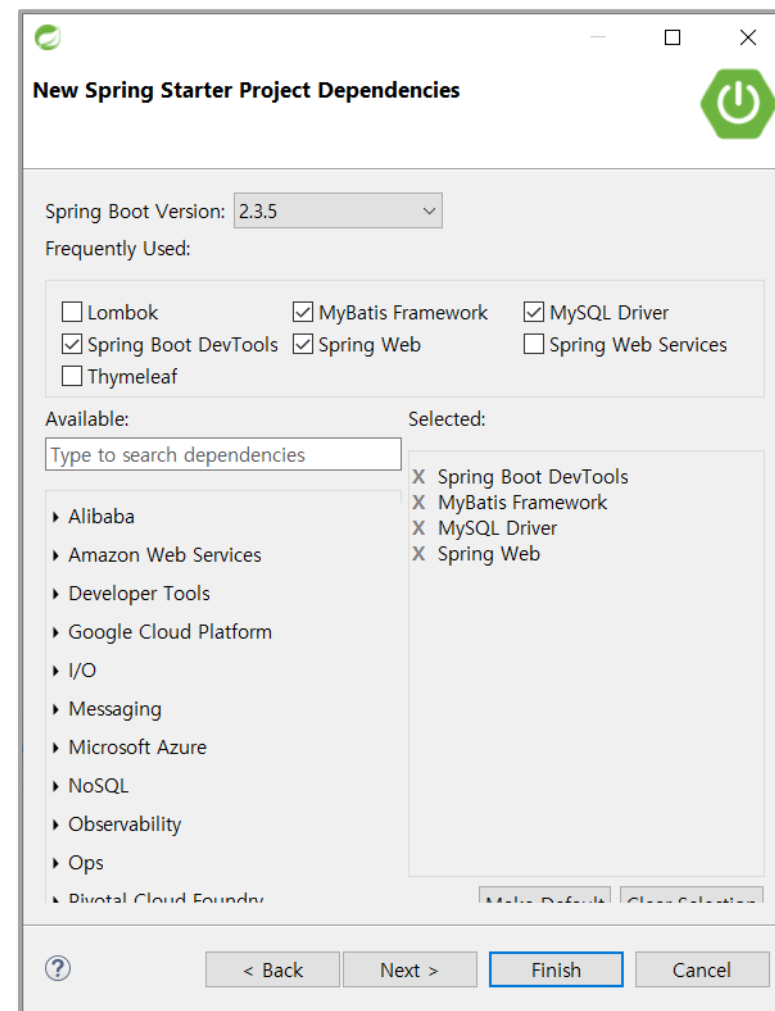
Description:

Package:

Working sets

☐ Add project to working sets

Working sets:



New Spring Starter Project Dependencies

Spring Boot Version:

Frequently Used:

☐ Lombok ☒ MyBatis Framework ☒ MySQL Driver

☒ Spring Boot DevTools ☒ Spring Web ☐ Spring Web Services

☐ Thymeleaf

Available:

Type to search dependencies

- Alibaba
- Amazon Web Services
- Developer Tools
- Google Cloud Platform
- I/O
- Messaging
- Microsoft Azure
- NoSQL
- Observability
- Ops
- Pivotal Cloud Foundry

Selected:

- X Spring Boot DevTools
- X MyBatis Framework
- X MySQL Driver
- X Spring Web

✓ axios를 이용한 Spring REST API와의 통신.

- pom.xml에 dependency추가.

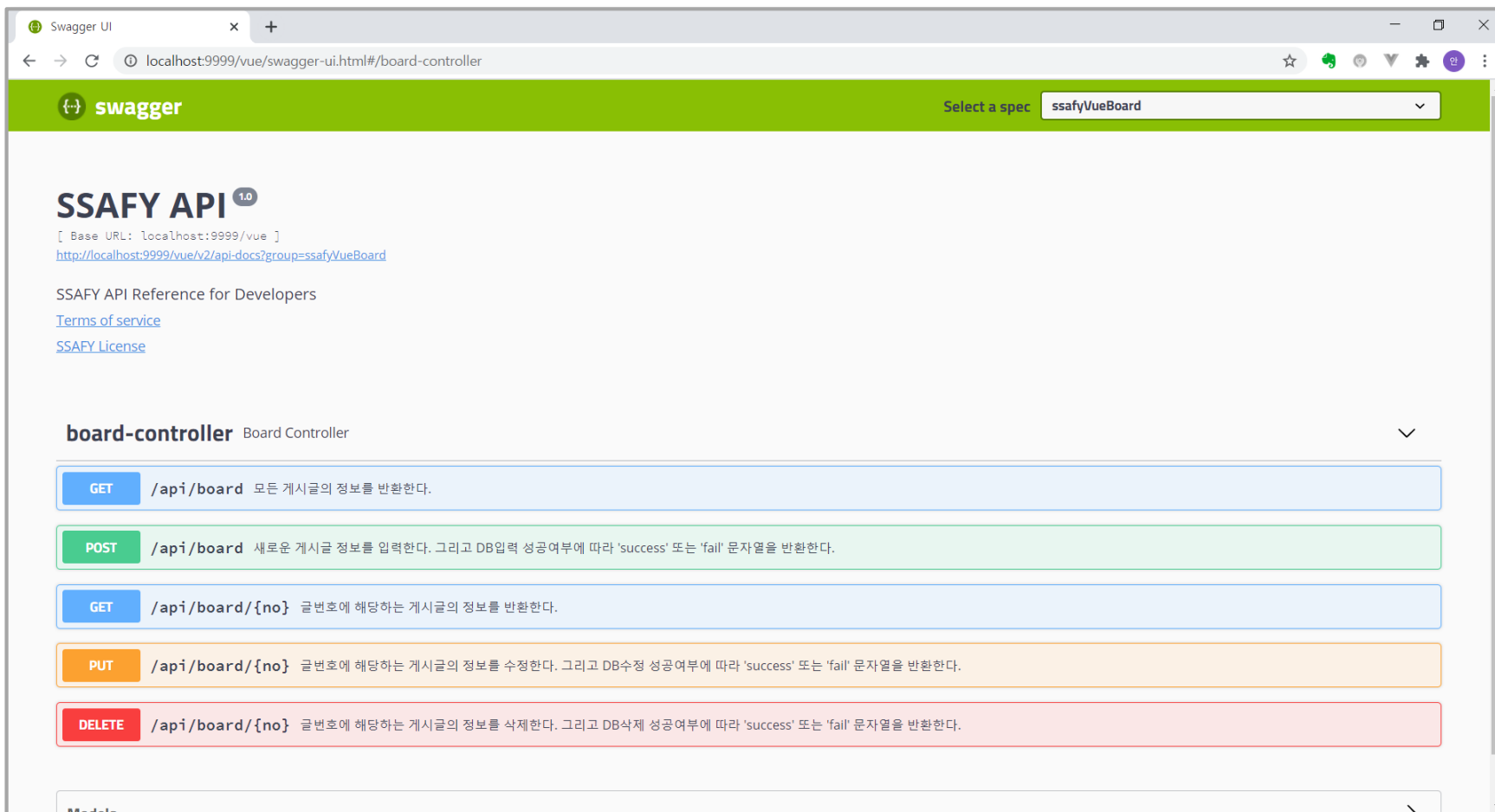
```
<!-- Swagger2 -->
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger2</artifactId>
  <version>2.9.2</version>
</dependency>
<dependency>
  <groupId>io.springfox</groupId>
  <artifactId>springfox-swagger-ui</artifactId>
  <version>2.9.2</version>
</dependency>

<!-- JSON Converter(컨트롤러에서 전달한 VO데이터를 JSON으로 변경하기 위해 사용) -->
<dependency>
  <groupId>com.fasterxml.jackson.core</groupId>
  <artifactId>jackson-databind</artifactId>
</dependency>
```

- 제공된 프로젝트 import.
- vue_table.sql >> table 생성.

✓ axios를 이용한 Spring REST API와의 통신.

- VueApiApplication >> Run.
- <http://localhost:9999/vue/swagger-ui.html>



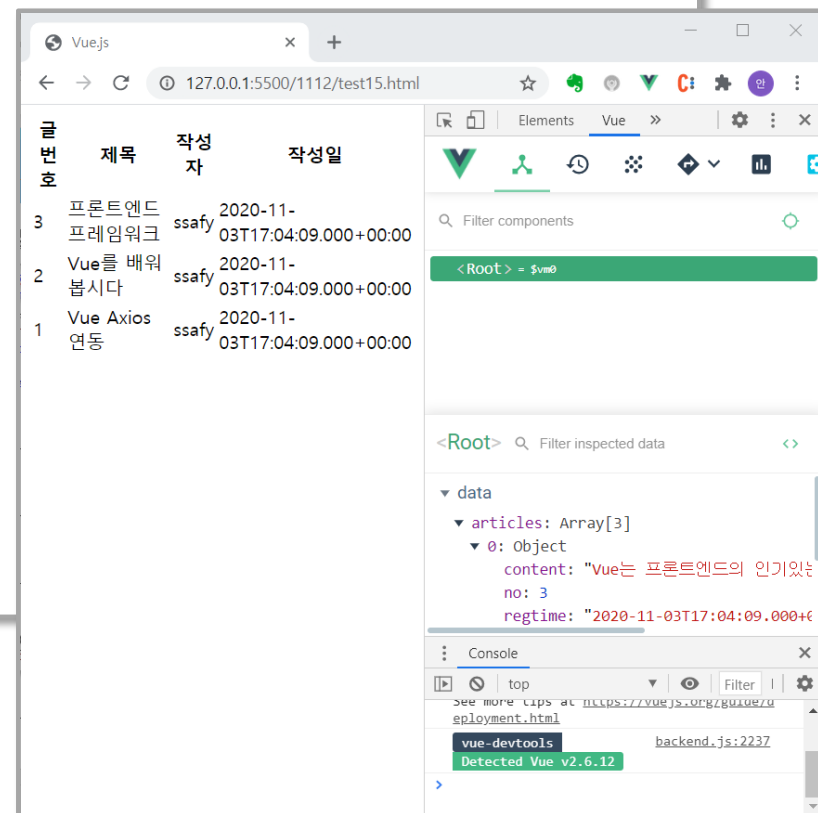
✓ axios를 이용한 Spring REST API와의 통신.

```
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>
<script src="https://unpkg.com/axios/dist/axios.min.js"></script>
</head>
<body>
  <div id="app">
    <div id="result">
      <table>
        <tr>
          <th>글번호</th>
          <th>제목</th>
          <th>작성자</th>
          <th>작성일</th>
        </tr>
        <tr v-for="article in articles">
          <td>{{article.no}}</td>
          <td>{{article.title}}</td>
          <td>{{article.writer}}</td>
          <td>{{article.regtime}}</td>
        </tr>
      </table>
    </div>
  </div>
</body>
```

test15.html

```
<script>
  const addr = 'http://localhost:9999/vue/api/board';

  new Vue({
    el: '#app',
    data: {
      articles: [],
    },
    created() {
      axios
        .get(addr)
        .then((response) => {
          this.articles = response.data;
        })
        .catch((error) => {
          console.dir(error);
        });
    },
  });
</script>
```



이어서..

삼성 청년 SW 아카데미

vue-router

삼성 청년 SW 아카데미

✓ vue-router.

- 라우팅 : 웹 페이지 간의 이동 방법.
- Vue.js의 공식 라우터.
- 라우터는 컴포넌트와 매핑.
- Vue를 이용한 SPA를 제작할 때 유용.
- URL에 따라 컴포넌트를 연결하고 설정된 컴포넌트를 보여준다.

<https://router.vuejs.org/kr>

✓ vue-router 설치.

▪ CDN 방식.

- `<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js"></script>`
- `<script src="https://unpkg.com/vue-router/dist/vue-router.js"></script>`

▪ NPM 방식.

- `npm install vue-router`

✓ vue-router 연결.

- 'routes' 옵션과 함께 router instance 생성.

```
// 라우트 컴포넌트
const Main = {
  template: '<div>메인 페이지</div>',
};
const Board = {
  template: '<div>자유 게시판</div>',
};

const router = new VueRouter({
  routes: [
    {
      path: '/',
      component: Main,
    },
    {
      path: '/board',
      component: Board,
    },
  ],
});
```

라우트 컴포넌트 정의

✓ vue-router 이동 및 렌더링.

- 네비게이션을 위해 router-link 컴포넌트를 사용.
- 속성은 'to' prop을 사용.
- 기본적으로 'router-link'는 'a' 태그로 렌더링.

```
<router-link to="/">HOME</router-link>  
<router-link to="/board">게시판</router-link>
```

- 현재 라우트에 맞는 컴포넌트가 렌더링된다.

```
<!-- 현재 라우트에 맞는 컴포넌트가 렌더링 -->  
<router-view></router-view>
```

✓ vue-router 이동 및 렌더링. router02



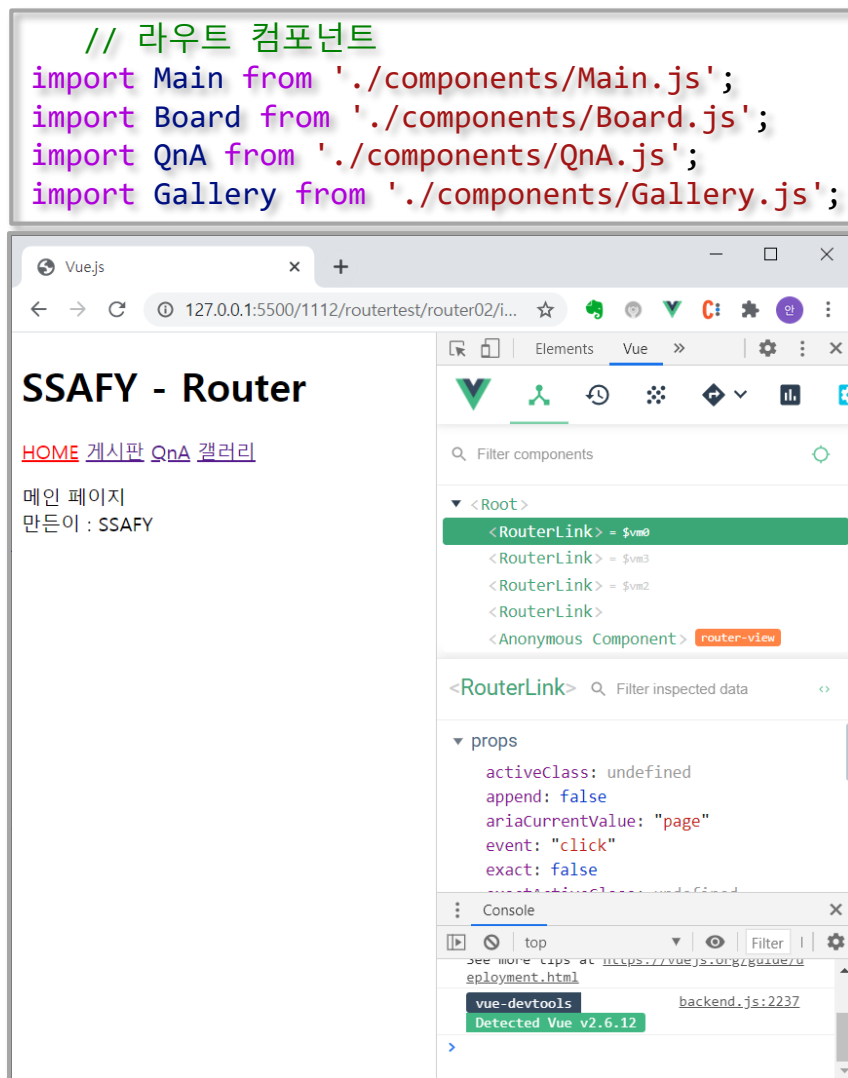
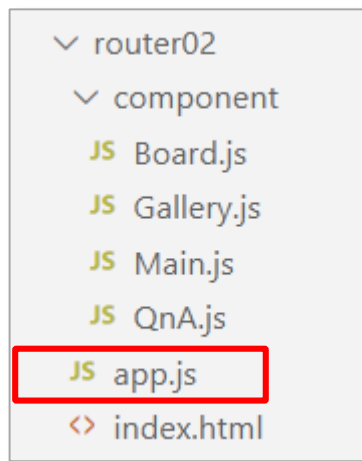
```
export default {
  template: '<div>자유 게시판</div>',
};
```

router02/components/Board.js

```
<div id="app">
  <h1>SSAFY - Router</h1>
  <p>
    <router-link to="/">HOME</router-link>
    <router-link to="/board">게시판</router-link>
    <router-link to="/qna">QnA</router-link>
    <router-link to="/gallery">갤러리</router-link>
  </p>
  <router-view></router-view>
  <div>만든이 : SSAFY</div>
</div>
<script type="module" src="app.js"></script>
```

router02/index.html

✓ vue-router 이동 및 렌더링. router02



✓ \$router, \$route

- 라우터 설정.

```
const router = new VueRouter({  
  routes: [  
    ...,  
    {  
      path: '/board/:no',  
      component: BoardView,  
    },  
    ...  
  ],  
});
```

- 라우터 링크.

```
<router-link to="/board/30">30번 게시글</router-link>
```

- 전체 라우터 정보

```
this.$router
```

- 현재 호출된 해당 라우터 정보

```
this.$route
```

```
this.$route.params.no
```

```
this.$route.path
```

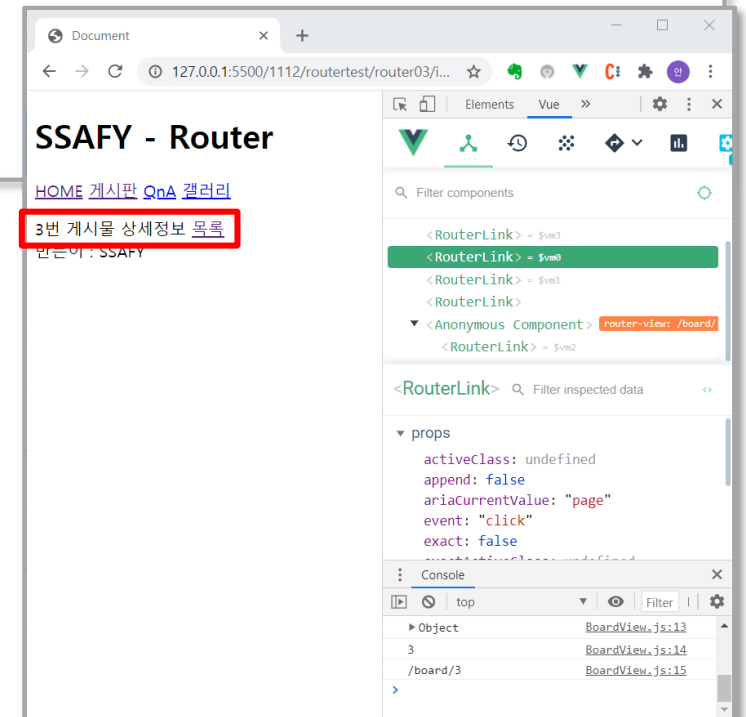
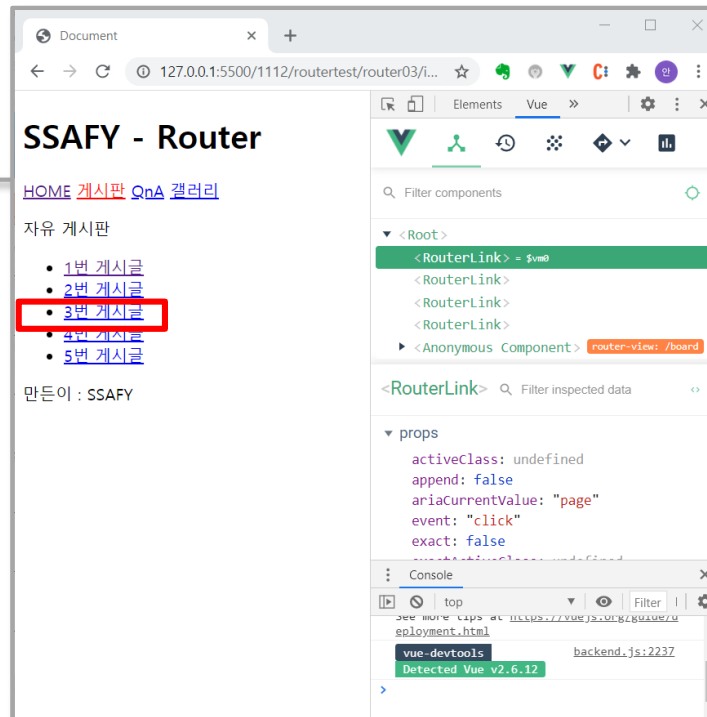
✓ vue-router 이동 및 렌더링. router03

✓ router03
 ✓ components
 JS Board.js
 JS BoardView.js
 JS Gallery.js
 JS Main.js
 JS QnA.js
 JS app.js
 <> index.html

```

export default {
  template: `<div>
    자유 게시판
    <ul>
      <li v-for="i in 5">
        <router-link :to="'/board/' + i">{{i}}번 게시물</router-link>
      </li>
    </ul>
  </div>`,
};
  
```

router03/components/Board.js



✓ vue-router 이동 및 렌더링. router03

- router03
 - components
 - Board.js
 - BoardView.js**
 - Gallery.js
 - Main.js
 - QnA.js
 - app.js
 - index.html

```
export default {
  template: `<div>
    {{no}}번 게시물 상세정보
    <router-link to="/board">목록</router-link>
  </div>`,
  data() {
    return {
      no: 0,
    };
  },
  created() {
    console.dir(this.$router); // 라우터 전체의 정보
    console.dir(this.$route); // 현재 호출된 해당 라우터 정보
    console.log(this.$route.params.no);
    console.log(this.$route.path);
    this.no = this.$route.params.no;
  },
};
```

router03/components/BoardView.js

▶ VueRouter	BoardView.js:12
▶ Object	BoardView.js:13
3	BoardView.js:14
/board/3	BoardView.js:15

✓ 이름을 가지는 라우트.

- 라우트는 연결하거나 탐색을 수행할 때 이름이 있는 라우트를 사용.
- Router Instance를 생성하는 동안 routes 옵션에 지정.

```
const router = new VueRouter({
  routes: [
    ...,
    {
      path: '/board',
      name: 'board',
      component: Board,
    },
    {
      path: '/board/:no',
      name: 'boardview',
      component: BoardView,
    },
    ...
  ],
});
```

```
<router-link :to="{name: 'boardview', params: {no: 3}}">3번 게시글</router-link>
<a @click="$router.push({name: 'boardview', params: {no: 3}});">3번 게시글</a>
```

✓ 이름을 가지는 라우트. router04

▼ router04
 ▼ components
 JS Board.js
 JS BoardView.js
 JS Gallery.js
 JS Main.js
 JS QnA.js
 JS app.js
 <> index.html

```
// 라우터 객체 생성시 이름설정
{
  path: '/board/:no',
  name: 'boardview',
  component: BoardView,
}
```

router04/app.js

```
export default {
  template: `<div>
    자유 게시판
    <ul>
      <li v-for="i in 5">
        <router-link :to="{name: 'boardview', params: {no: i}}">
          {{i}}번 게시물
        </router-link>
      </li>
    </ul>
  </div>`,
};
```

router04/components/Board.js

✓ 이름을 가지는 라우트. router04

✓ router04
 ✓ components
 JS Board.js
 JS BoardView.js
 JS Gallery.js
 JS Main.js
 JS QnA.js
 JS app.js
 <> index.html

```

export default {
  template: `<div>
    {{no}}번 게시물 상세정보
    <hr>
    <router-link :to="{name: 'board'}">목록</router-link>
  </div>`,
  data() {
    return {
      no: 0,
    };
  },
  created() {
    console.dir(this.$route); // 현재 호출된 해당 라우터 정보
    this.no = this.$route.params.no;
  },
};
  
```

router04/components/BoardView.js

```

<router-link :to="{name: 'main'}">HOME</router-link>
<router-link :to="{name: 'board'}">게시판</router-link>
<router-link :to="{name: 'qna'}">QnA</router-link>
<router-link :to="{name: 'gallery'}">갤러리</router-link>
  
```

router04/index.html

✓ 프로그래밍 방식 라우터.

- <router-link>를 사용하여 선언적 네비게이션용 anchor 태그를 만드는 것 외에도 라우터의 instance method를 사용하여 프로그래밍으로 수행.

```
$router.push('home')
```

```
$router.push({ path: 'home' })
```

```
$router.push({ name: 'boardview', params: { no: 3} })
```

```
$router.push({ path: '/board', query: {pg: 1} })
```

✓ 프로그래밍 방식 라우터. router05

✓ router05
 ✓ components
 JS Board.js
 JS BoardView.js
 JS Gallery.js
 JS Main.js
 JS QnA.js
 JS app.js
 <> index.html

```
// 라우터 객체 생성시 이름설정
{
  path: '/board/:no',
  name: 'boardview',
  component: BoardView,
}
```

router05/app.js

```
export default {
  template: `<div>
    자유 게시판
    <ul>
      <li v-for="i in 5">
        <a :href="'#' + i"
          @click="$router.push({name: 'boardview', params: {no: i}})">
          {{i}}번 게시물</a>
        </li>
      </ul>
    </div>`,
};
```

router05/components/Board.js

✓ 프로그래밍 방식 라우터. router05

- router05
 - components
 - Board.js
 - BoardView.js**
 - Gallery.js
 - Main.js
 - QnA.js
 - app.js
 - index.html**

```
export default {
  template: `<div>
    {{no}}번 게시물 상세정보
    <a href="#board" @click="$router.push('/board')">게시판</a>
  </div>`,
  data() {
    return {
      no: 0,
    };
  },
  created() {
    console.dir(this.$route); // 현재 호출된 해당 라우터 정보
    this.no = this.$route.params.no;
  },
};
```

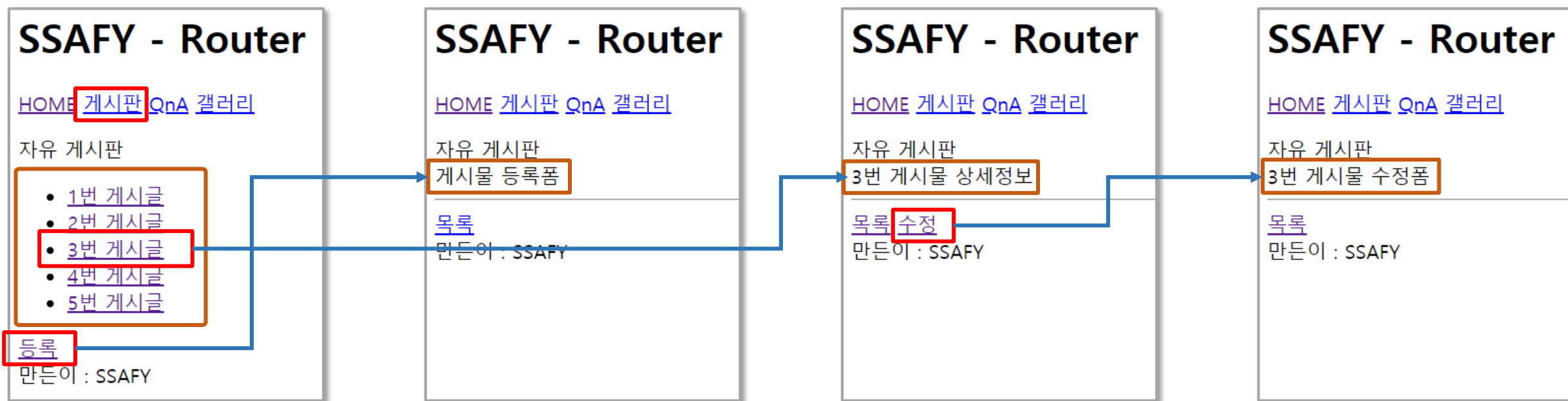
router05/components/BoardView.js

```
<a href="#main" @click="$router.push('/')">HOME</a>
<a href="#board" @click="$router.push('/board')">게시판</a>
<a href="#qna" @click="$router.push({name: 'qna'})">QnA</a>
<a href="#gallery" @click="$router.push({name: 'gallery'})">갤러리</a>
```

router05/index.html

✓ 중첩된 라우트.

- 앱 UI는 일반적으로 여러 단계로 중첩 된 컴포넌트 구조임.
- URL의 세그먼트가 중첩 된 컴포넌트의 특정 구조와 일치하는 것을 활용.



✓ 중첩된 라우트. router06

- ▼ router06
 - ▼ components
 - ▼ board
 - JS BoardList.js
 - JS BoardUpdate.js
 - JS BoardView.js
 - JS BoardWrite.js
 - JS Board.js
 - JS Gallery.js
 - JS Main.js
 - JS QnA.js
 - JS app.js**
 - <> index.html

```
{
  path: '/board',
  name: 'board',
  component: Board,
  redirect: '/board/list',
  children: [
    {
      path: 'list',
      name: 'boardlist',
      component: BoardList,
    },
    {
      path: 'write',
      name: 'boardwrite',
      component: BoardWrite,
    },
    ...
  ],
},
```

router06/app.js

// 라우트 컴포넌트

```
import Main from './components/Main.js';
import Board from './components/Board.js';
import QnA from './components/QnA.js';
import Gallery from './components/Gallery.js';
import BoardView from './components/board/BoardView.js';
import BoardUpdate from './components/board/BoardUpdate.js';
import BoardWrite from './components/board/BoardWrite.js';
import BoardList from './components/board/BoardList.js';
```


✓ 중첩된 라우트. router06

- ▼ router06
 - ▼ components
 - ▼ board
 - JS BoardList.js**
 - JS BoardUpdate.js
 - JS BoardView.js**
 - JS BoardWrite.js
 - JS Board.js
 - JS Gallery.js
 - JS Main.js
 - JS QnA.js
 - JS app.js
 - <> index.html

```
export default {
  template: `<div>
    <ul>
      <li v-for="i in 5">
        <router-link :to="'/board/detail/' + i">{{i}}번 게시글</router-link>
      </li>
    </ul>
    <router-link to="/board/write">등록</router-link>
  </div>`,
};
```

router06/components/BoardList.js

```
export default {
  template: `<div>
    {{no}}번 게시물 상세정보
    <hr>
    <div>
      <router-link :to="{name: 'boardlist'}">목록</router-link>
      <router-link :to="'/board/update/' + no">수정</router-link>
    </div>
  </div>`,
  data() { ... },
  created() { ... },
};
```

router06/components/BoardView.js

✓ 라우트 리다이렉트.

```
const router = new VueRouter({
  mode: 'history',
  routes: [
    ...,
    {
      path: '/board',
      name: 'board',
      component: Board,
      redirect: '/board/list',      or redirect: {name: 'list'}
      children: [
        ...
      ],
    },
  ],
});
```

내일 방송에서 만나요!

삼성 청년 SW 아카데미