

Vue Router

Bok, Jong Soon
javaexpert@nate.com
<https://github.com/swacademy/Vue.js>

SPA

- 처음 HTML Page 하나를 Load한 다음, 그 이후 사용자의 요청에 따라 Ajax로 정보를 받아오면서 동적으로 Page를 Update하는 Web Application.
- 일반적인 Web Application은 Page 이동 시 대상 URL을 Server에 요청해서 전체 Page를 Loading한다.
- SPA는 Page 이동을 Client에서 처리한다.
- Page 이동 시 Ajax를 사용해서 필요한 때에 Data를 받아와서 View를 화면에 표시한다.
- 전체 Page에 해당하는 HTML을 모두 받아오는 데 필요한 Overhead가 줄어들기 때문에 Application의 속도가 향상되며, 더욱 매끄러운 User Experience를 제공할 수 있다.

SPA (Cont.)

- SPA를 구현하기 위해 고려할 사항
 - Client-side에서 History를 관리하는 Page 이동(Routing 관리)
 - 비동기로 Data 받아오기
 - View Rendering
 - Module화된 Code 관리
- Router 또는 Routing Library Module이 이런 기능들을 제공한다.

Vue Router

- Vue Library를 이용하여 Single Page Application을 구현할 때 사용하는 Library.
- Vue Router Installation
 - CDN 방식
 - NPM 방식
- CDN 방식

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

- NPM 방식

```
npm install vue-router
```

Vue Router 등록

- Vue Router를 설치한 후, 다음과 같은 코드를 통해 Router Instance를 생성하고 Vue Instance에 등록한다.

```
// 라우터 인스턴스 생성
var router = new VueRouter({
  // 라우터 옵션
})

// 인스턴스에 라우터 인스턴스를 등록
new Vue({
  router: router
})
```

Vue Router Options

- Vue Router를 등록 후, Router에 Option을 정의한다.
- 대부분의 SPA에서는 아래와 같이 2개 Option을 필수로 지정한다.
 - **routes**
 - Routing 할 URL과 Component 값 지정
 - **mode**
 - URL의 Hash 값 제거 속성

```
new VueRouter({  
  mode: 'history',  
  routes: [  
    { path: '/login', component: LoginComponent },  
    { path: '/home', component: HomeComponent }  
  ]  
})
```

router-view

- Browser의 주소 창에서 URL이 변경되면, 앞에서 정의한 **routes** 속성에 따라 해당 Component가 화면에 나타난다.
- 이 때 나타나는 지점이 Template의 **<router-view>**이다.

```
<div id="app">  
  <router-view></router-view> <!-- LoginComponent 또는 HomeComponent -->  
</div>
```

- 앞에서 정의한 Routing Option 기준으로 /login은 LoginComponent를 /home은 HomeComponent를 화면에 표시한다.

router-link

- 일반적으로 Web Page에서 Page 이동을 할 때는 사용자가 URL을 다 입력해서 이동하지 않는다.
- 이 때 화면에서 특정 Link를 Click해서 Page를 이동할 수 있게 해줘야 하는데 그게 바로 **<router-link>** 입니다.

```
<router-link to="이동할 URL"></router-link>
```

```
<div>  
  <router-link to="/login"></router-link>  
</div>
```


Lab

```
8      <script src="https://unpkg.com/vue@2.6.10/dist/vue.js"></script>
9      <script src="https://unpkg.com/vue-router/dist/vue-router.js"></script>
10 </head>
11 <body>
12     <div id="app">
13         <router-link to="/top">최상위 페이지</router-link>
14         <router-link to="/users">사용자 목록 페이지</router-link>
15         <router-view></router-view>
16     </div>
```

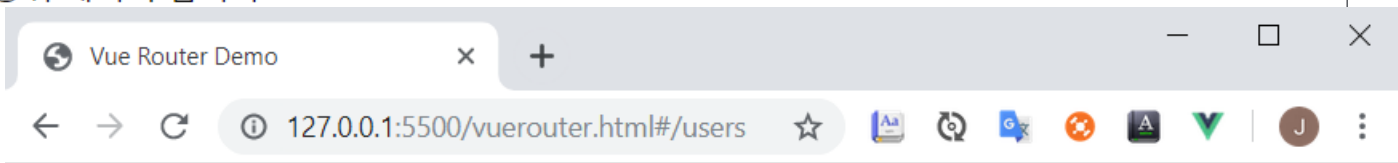
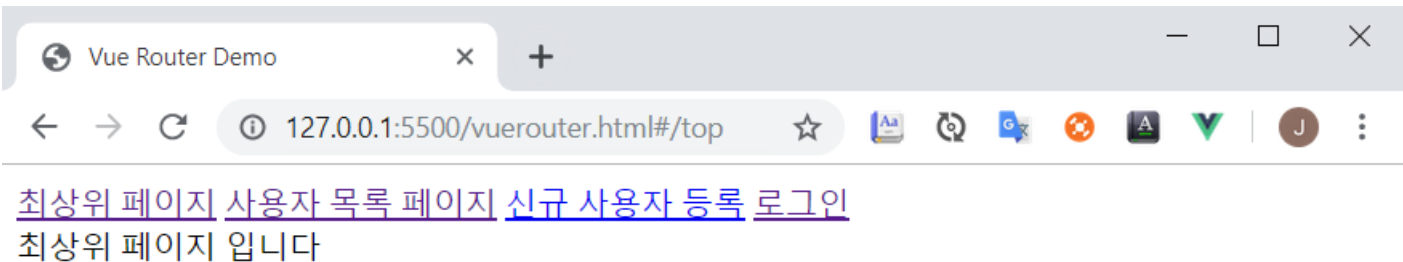
Lab (Cont.)

```
17 <script>
18   //Route 옵션을 지정해서 라우터 인스턴스를 생성
19   let router = new VueRouter({
20     routes : [
21       {
22         path : '/top',
23         component : {
24           template : '<div>최상위 페이지</div>'
25         }
26       },
27       {
28         path : '/users',
29         component : {
30           template : '<div>사용자 목록 페이지</div>'
31         }
32       }
33     ]
34   })
35   let app = new Vue({
36     router : router
37   }).$mount('#app');
38 </script>
```

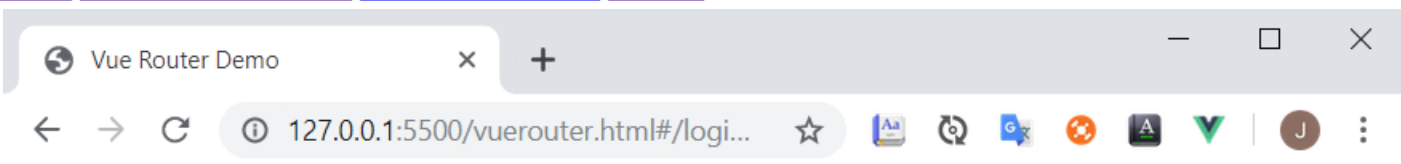
← → ↻ ⓘ 127.0.0.1:5500/vuerouter.html#/users

최상위 페이지 사용자 목록 페이지
사용자 목록 페이지

Lab (Cont.)



[최상위 페이지](#) [사용자 목록 페이지](#) [신규 사용자 등록](#) [로그인](#)
[한지민](#)
[김지민](#)

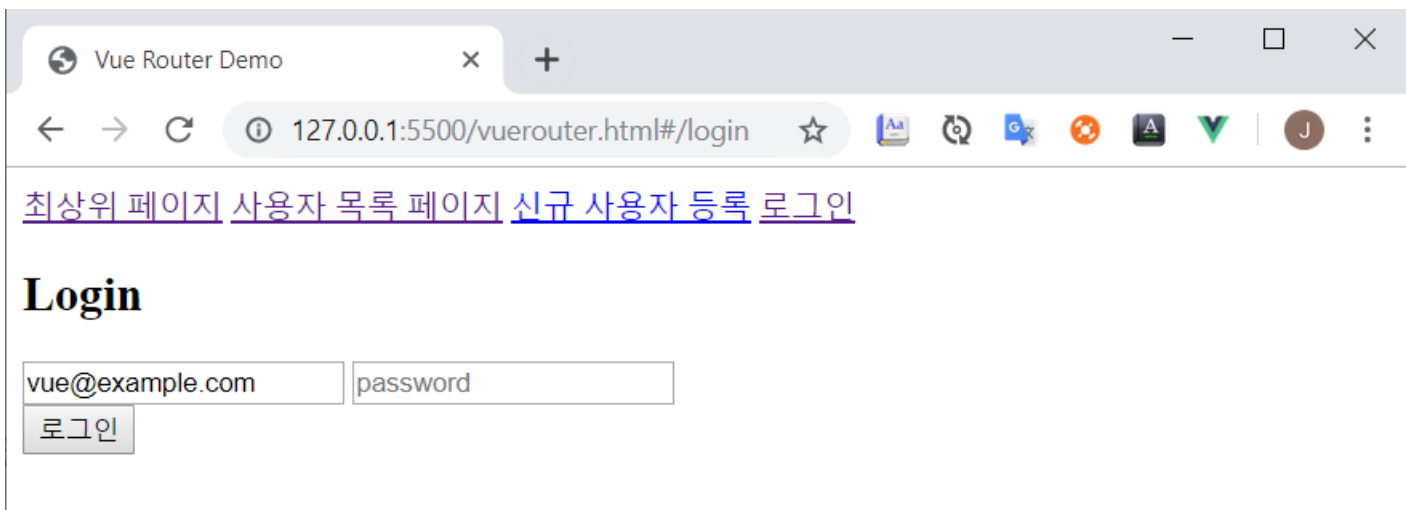


[최상위 페이지](#) [사용자 목록 페이지](#) [신규 사용자 등록](#) [로그인](#)

Login

로그인이 필요합니다

<input type="text" value="vue@example.com"/>	<input type="password" value="password"/>
<input type="button" value="로그인"/>	



[최상위 페이지](#) [사용자 목록 페이지](#) [신규 사용자 등록](#) [로그인](#)

Login

<input type="text" value="vue@example.com"/>	<input type="password" value="password"/>
<input type="button" value="로그인"/>	