Sec01

Vue + Springboot 연동하기

작성자: 강태경



# Vue + Springboot

Front-end Vue와 Backend Springboot의 연동은(연결) axios 통신을 통해 이루어짐,  
axios의 각 함수(get(url), post(url), put(url), delete(url) ) 를 url 정해서 호출하면 그 url 과 똑같은 어노테이션 함수가 호출되면서 연결됨

* Axios.create() : springboot 와 연결할 주소:포트 를 정의하는 함수,   
  전송할 문서 타입을 추가하기도 함
* DataService.js : axios 함수(get(), post(), put(), delete() )들을 모든 컴포넌트(페이지)에서 가져와서 쓸 수 있게 공통함수(공용 함수)로 만들어 놓은 자바스크립트 소스,  
  각각의 컴포넌트에서는 import 해서 사용하면 됨  
  front 요청 -------(연결)-------- backend 응답   
  axios.get(url) --------------- @GetMapping(url) 함수명(@PathVariable 매개변수)  
  axios.post(url) -------------- @PostMapping(url) 함수명(@RequestBody 매개변수)  
  axios.put(url) -------------- @PutMapping(url) 함수명(@RequestBody 매개변수)  
  axios.delete(url) ------------ @DeleteMapping(url) 함수명(@PathVariable 매개변수)
* 개발 시 frontend(Vue), backend(Springboot) 서버는 각각 가동시켜야 함
* CORS 보안 : 한 사이트에서 URL 기본 주소(포트 , 아이피 등)를 달리 사용하면 blocking(막힘) 되는 기능(웹 브라우저 기본 보안)  
  해킹을 방지하기 위한 목적으로 기본적으로 웹브라우저에 있는 보안 기능  
  해결방법)  
  1. Springboot 에서 설정 :  
   1) @CrossOrigin(origins = “url”) : CORS 를 허용할 url 주소를 매개변수로 코딩  
   장점 : 간략  
   단점 : 각 컨트롤러 클래스마다 @ 적용해야함  
   @CrossOrigin(origins = “Vue IP주소:포트”)   
   ex) @CrossOrigin(origins = "http://localhost:8081")  
   2) CORS 보안을 허용할 클래스를 작성해도 가능  
   장점 : 클래스 1개만 작성하면 모든 컨트롤러 클래스에 적용됨  
   단점 : 복잡성  
  2. Vue 에서 설정 :  
   1) Vue에서 proxy 설정을 통해 CORS 보안을 해결하기도 함  
  3. 외부 사이트를 통한 해결 :  
   1) CORS 보안을 해결하기 위한 외부 사이트가 있음.  
   외부 proxy 사이트를 제공받아 허용하기도 함