Axios

Axios

- **Axios**

  Vue.js에서 권고하는 Promise 기반의 HTTP 통신 라이브러리
  * 우선은 JavaScript의 ajax와 같은 기능을 수행한다고 생각하면 됨

- **Promise**

  자바스크립트에서 비동기 작업을 수행하기 위한 방법 중 하나
  → Callback, Promise, Promise + generator, async/await

Axios

- **참고**

vue-resource의 경우 옛날에 Vuejs에서 사용하던 라이브러리이기 때문에 vue 관련된 레퍼런스를 찾을 때 vue-resource를 사용 한다면 상당히 오래전의 코드일 가능성이 높으므로 동작하지 않을 가능성이 높음

* 더 이상 Vuejs의 공식 라이브러리가 아님

Axios

- **Axios 설치**

npm install axios

```
user@DESKTOP-I9RHACC MINGW64 /d/vuejs
$ npm install axios

added 9 packages in 677ms

1 package is looking for funding
  run `npm fund` for details
```

Axios

- **Vuews/ex3_axios/GetUser.vue**

Button 클릭 시 Axios로 샘플 데이터 받아오기

```vue
<template>
    <button @click="getData">GetUser</button>
</template>

<script>
import axios from 'axios';

export default {
    methods: {
        getData: function() {
            axios.get('https://jsonplaceholder.typicode.com/users')
            .then(function(response) {
                console.log(response);
            })
            .catch(function(error) {
                console.log(error);
            });
        }
    }
}
</script>
```

REST API를 호출할 수 있는 샘플 데이터를 제공하는 사이트

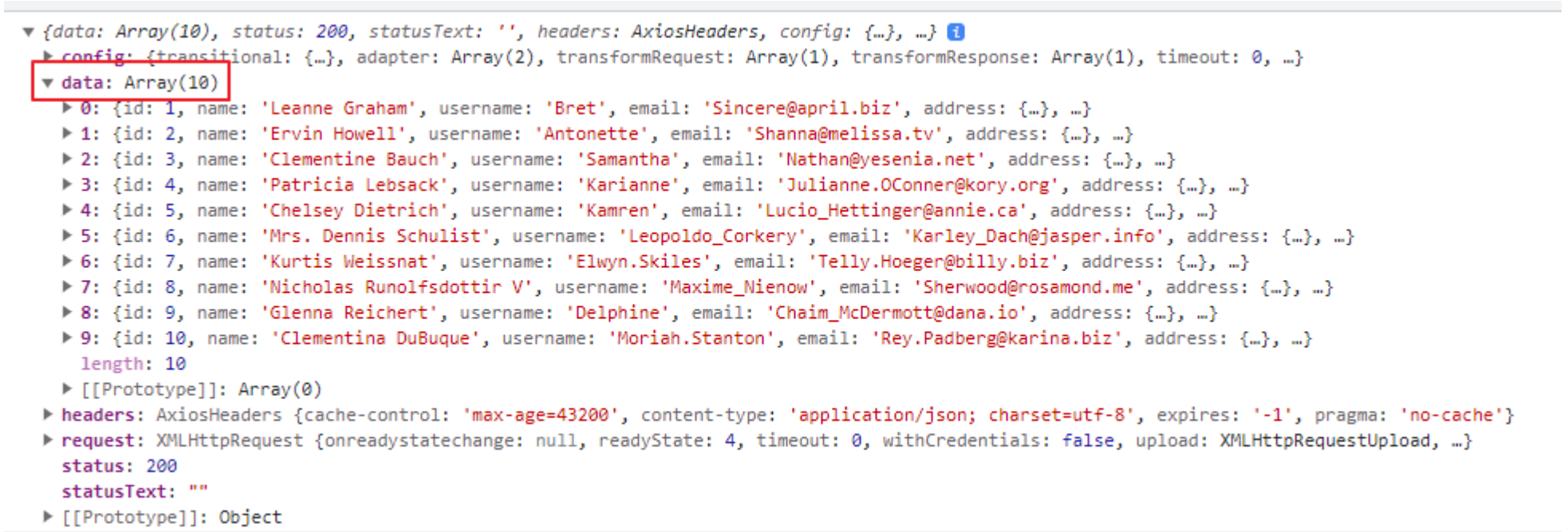
성공

실패

Axios

- **콘솔 로그 확인**

10개의 샘플 데이터를 확인할 수 있고, respons에 대한 값이 아래와 같으니
response.data를 통해 필요한 유저 데이터만 꺼낼 수 있는 것을 확인 해보기



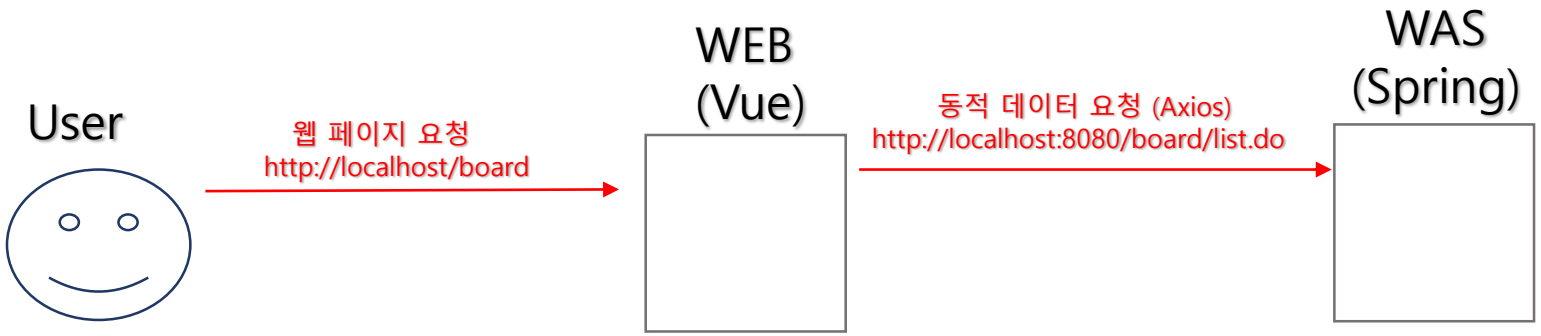
스프링 연동

스프링 연동

- 스프링과의 통신 방식

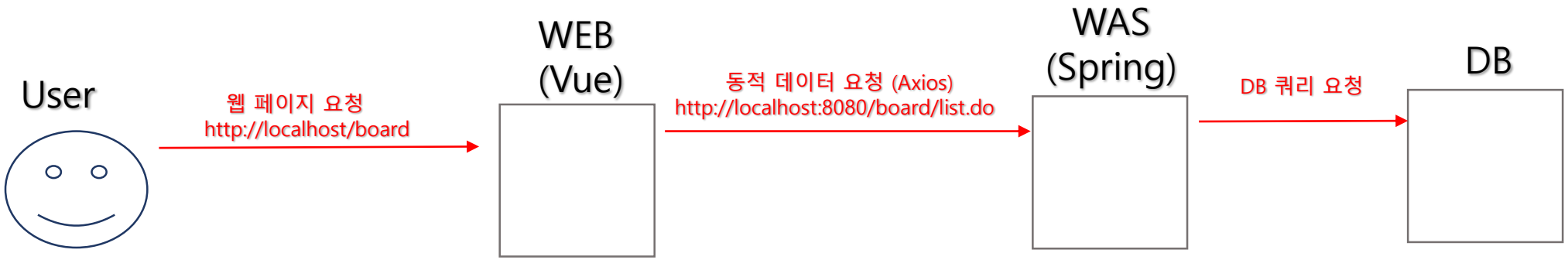
스프링 연동

- 스프링과의 통신 방식



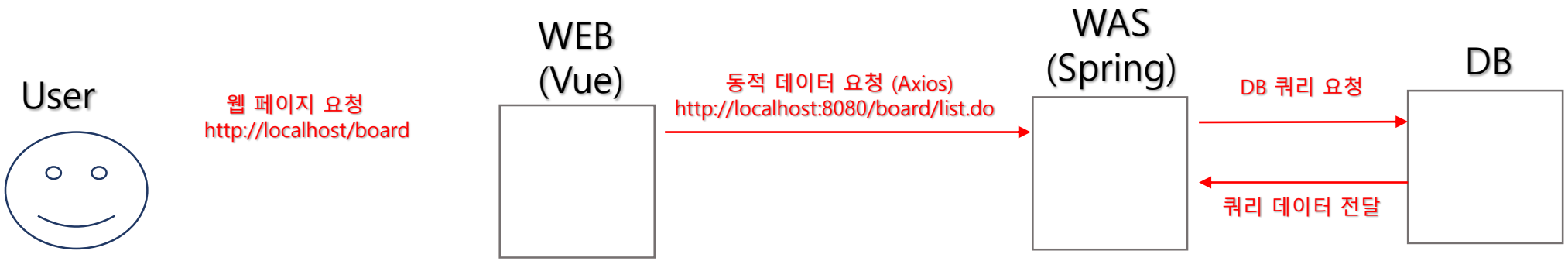
스프링 연동

- 스프링과의 통신 방식



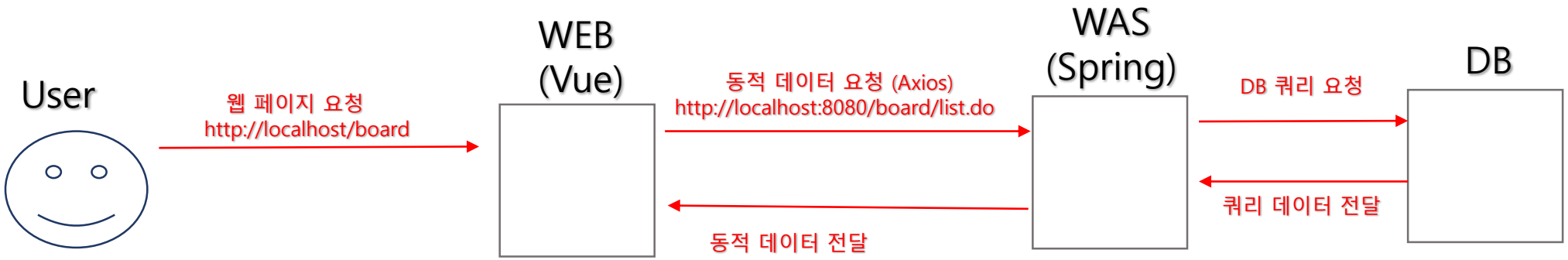
스프링 연동

- 스프링과의 통신 방식



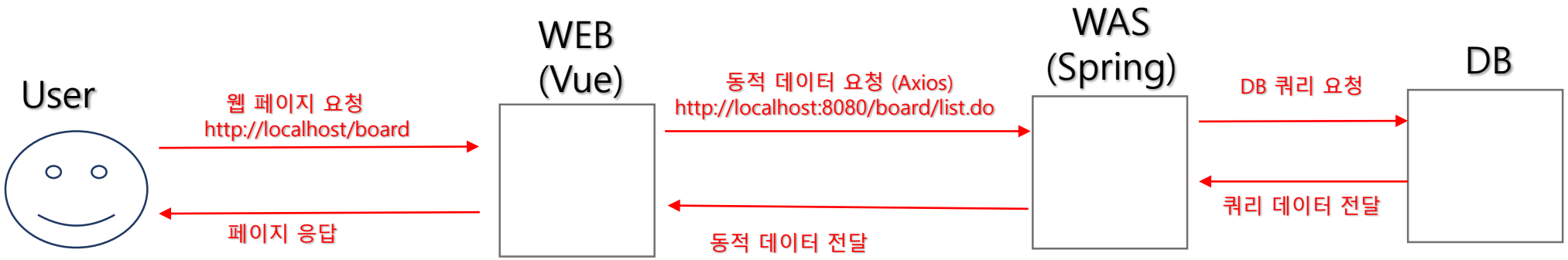
스프링 연동

- 스프링과의 통신 방식



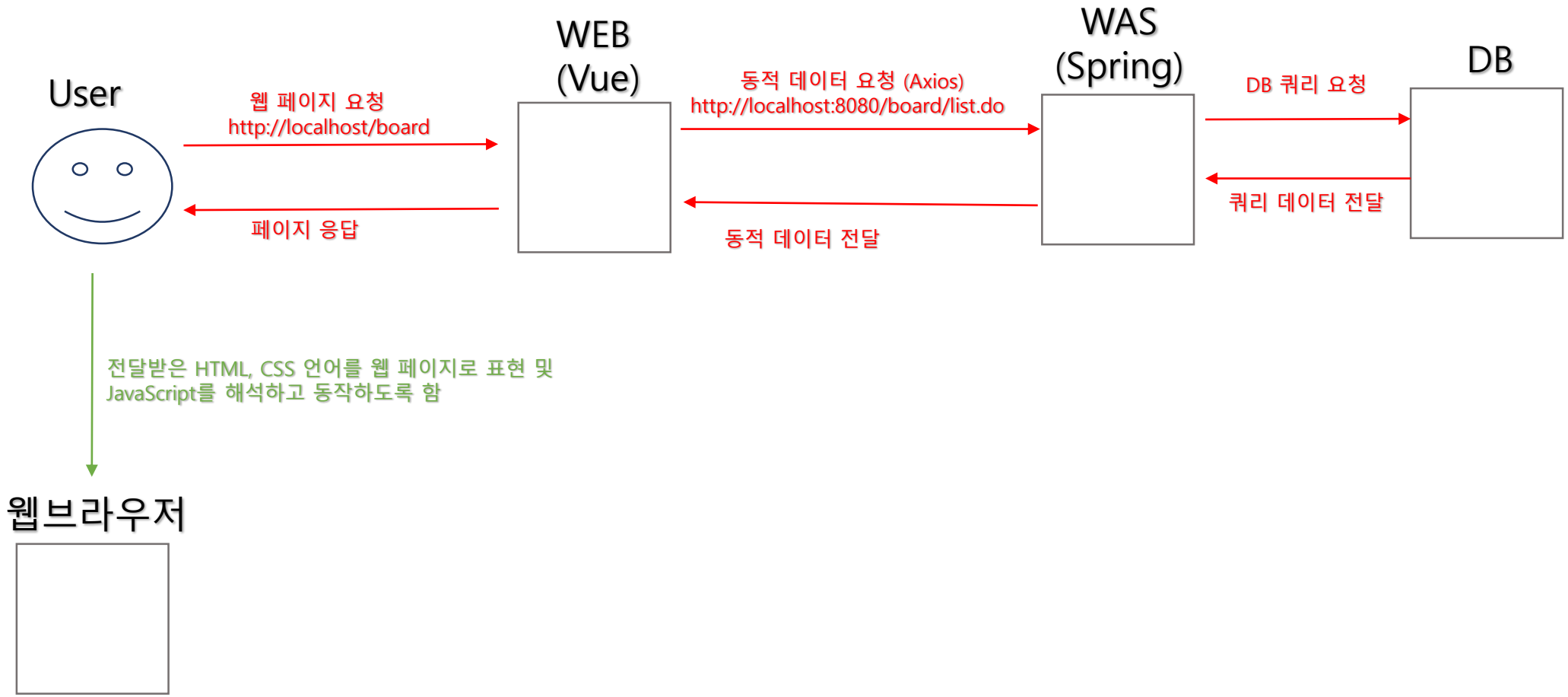
스프링 연동

- 스프링과의 통신 방식



스프링 연동

- 스프링과의 통신 방식



스프링 연동

- **WEB-INF/web.xml**

CORS를 해결하기 위해 아래 코드 붙여 넣기 후 서버 재시작

```
<filter>
    <filter-name>CorsFilter</filter-name>
    <filter-class>org.apache.catalina.filters.CorsFilter</filter-class>
    <init-param>
        <param-name>cors.allowed.origins</param-name>
        <param-value>*</param-value>
    </init-param>
    <init-param>
        <param-name>cors.allowed.methods</param-name>
        <param-value>GET,POST,HEAD,OPTIONS,PUT,DELETE</param-value>
    </init-param>
    <init-param>
        <param-name>cors.allowed.headers</param-name>
        <param-value>Content-Type,X-Requested-With,accept,Origin,Access-Control-Request-Method,Access-Control-Request-Headers</param-value>
    </init-param>
    <init-param>
        <param-name>cors.exposed.headers</param-name>
        <param-value>Access-Control-Allow-Origin,Access-Control-Allow-Credentials</param-value>
    </init-param>
    <init-param>
    <!-- 쿠키 통신을 안하는데 이걸 true로 하면 4XX 서버 에러가 뜬다 -->
        <param-name>cors.support.credentials</param-name>
        <param-value>false</param-value>
    </init-param>
    <init-param>
        <param-name>cors.preflight.maxage</param-name>
        <param-value>10</param-value>
    </init-param>
</filter>
<filter-mapping>
    <filter-name>CorsFilter</filter-name>
    <url-pattern>/*</url-pattern>
</filter-mapping>
```

스프링 연동

- **CORS (Cross-Origin Resource Sharing)**

보안적인 이유로 서로 다른 출처끼리의 통신을 브라우저가 차단함

프로토콜(http, https), 포트 번호, 도메인의 일치 여부에 따라 같은 출처인지를 비교하고
하나라도 다를 경우 CORS 에러를 발생시키기 때문에 서로 다른 출처끼리 통신이 필요하다면
이를 허용할 수 있도록 별도의 설정을 해주어야 함

스프링 연동

- **Controller**

  컨트롤러의 어노테이션을 @RestController로 변경

```
@RestController
@RequestMapping("board")
public class BoardController {
    public static final String UPLOAD_

    @Autowired
```

스프링 연동

- **Controller**

반환 타입을 ResponseEntity<?> 으로 변경

```
@GetMapping("list.do")
public ResponseEntity<?> boardList(@RequestParam(value="search
                        @RequestParam(value="cpage", defaultVa
                        HttpSession session,
                        Model model) {
```

스프링 연동

- **Controller**

  모든 객체를 묶어서 보내기 위해 아래와 같이 HashMap으로 코드 작성

```
Map<String, Object> response = new HashMap<>();
response.put("list", list);
response.put("row", row);
response.put("pi", pi);
response.put("msg", msg);
response.put("status", status);
```

스프링 연동

- **Controller**

  리턴에 ResponseEntity<> 객체 생성 및 response 반환

```
return new ResponseEntity<>(response, HttpStatus.OK); /
```

스프링 연동

- **Vue.js**

axios에 로컬 주소를 입력하고 반환 값 확인하기

```
methods: {
    getData: function() {
        axios.get('http://localhost/board/list.do')
        .then((response) => {
            console.log(response);
        })
        .catch((error) => {
            console.log(error);
        });
    }
}
```