1) 프론트 컨트롤러 패턴 실행 프로세스를 스프링 MVC의 주요 구성 요소를 통해 설명하시오.

프론트 컨트롤러는 DispatcherServlet을 의미한다.

- ① DispatcherServlet이 HTTP 요청을 받는다.
- ② DispatcherServlet은 서브 Controller로 HTTP 요청 위임을 받는다. 이 단계에서 DispatcherServeet은 서브 Controller를 스프링 빈으로 등록한다.
- ③ 서브 Controller는 클라이언트의 요청 처리를 위해 DAO 객체를 호출한다.
- ④ DAO 객체는 리소스를 엑세스하여 Model 객체를 생성 후 요청 결과를 리턴한다.
- ⑤ DispatcherServlet은 처리 결과에 적합한 뷰에 화면 처리 요청을 한다.
- ⑥ 선택된 뷰는 화면에 모델 객체를 가져와 화면을 처리한다.
- ⑦ 이과정을 통해 결국 HTTP요청에 응답을 한다.

2) @ModelAttribute, @PathVariable, @RequestParam 차이점을 설명하시오.

@PathVariable

URL 경로 내의 변수 값을 @PathVariable 적용 변수로 전달한다. 변수를 페이지로 전달해 요청 페이지에서 변수를 사용할 수 있다.

```
//http://localhost:8080/web /try/thankyou
@RequestMapping(value="/try/{msg}", method = RequestMethod.GET)
public String getUserTest(@PathVarible("msg") String msg) {
    logger.info(msg);...
```

@RequestParam

요청 파라미터 값을 @RequestParam 적용 변수로 전달한다. 변수를 로컬 함수내에서만 쓸 수 있다. 요청된 페이지로 변수를 전달하지 못한다.

```
// http://localhost:8080/web/tryA?msg=thankyou
@RequestMapping(value="tryA", method = RequestMethod.GET)
public String getUserTest1(@RequestParam("msg") String msg) {
    logger.info(msg);....
```

@ModelAttribute

요청 파라미터 값을 @ModelAttribute 적용 변수로 전달한다. @RequestParam과 다른점은 배열 형태의 값을 지정할 수 있다. 예를 들면 tryC, tryD 양쪽 URL에 대응하는 메서드를 정의할수 있다.

```
// http://localhost:8080/web/tryC?msg=thankyou
@RequestMapping(value={"/tryC", "/tryD"}, method = RequestMethod.GET)
public String getUserTest2(@ModelAttribute("msg") String msg) {
    logger.info(msg);......
```

3) @RequestMapping, @RestController, @RequestBody, @ResponseBody의미를 설명하시오.

@RequestMapping

URL과 컨트롤러 메서드의 매핑을 설정하는 애노테이션이다.

@RestController

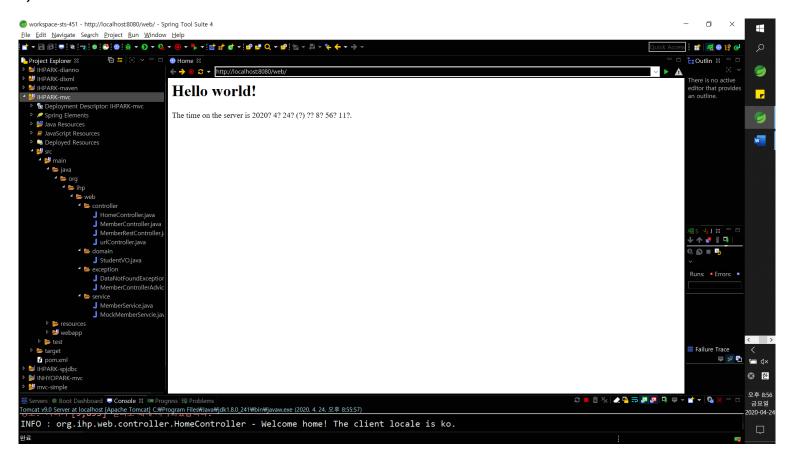
REST API를 제공하는 컨트롤러를 의미한다.@Controller와 @ResponseBody의 의미를 합친 애노테이션이다.

@RequestBody

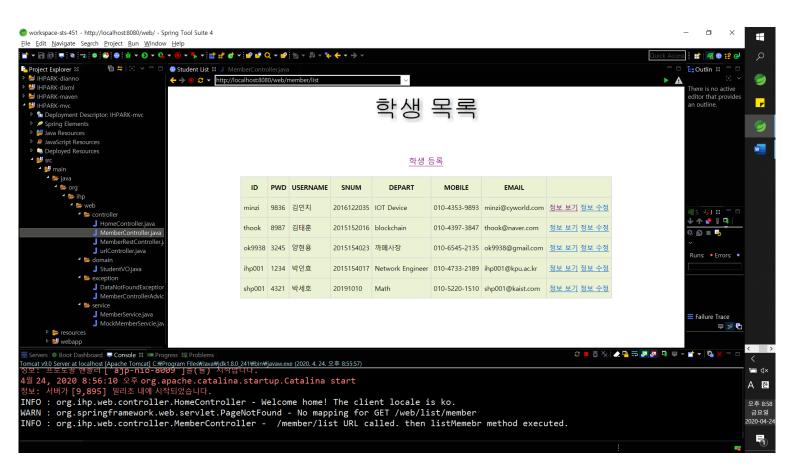
컨트롤러 메서드 매개 변수에 @RequestBody가 애노테이션 된 경우, 스프링은 요청된 HTTP request body를 해당 매개 변수에 바인딩한다.

@ResponseBody

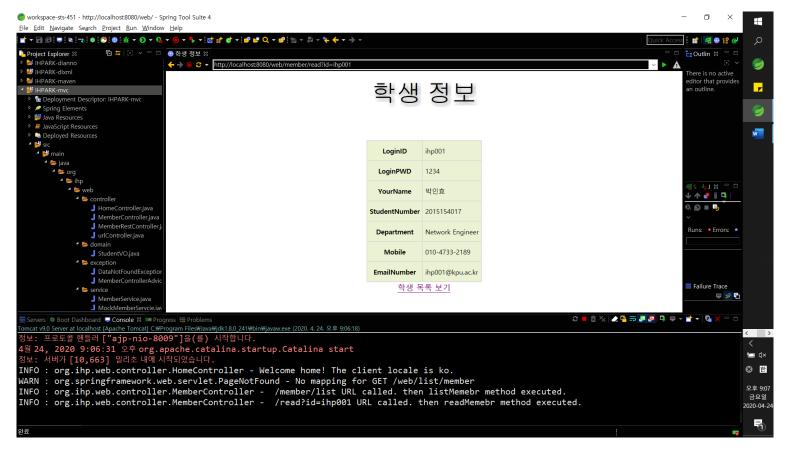
컨트롤러 메소드가 @ResponseBody로 어노테이션 된 경우, 스프링은 반환 값을 나가는 HTTP response body에 바인딩한다. 스프링은 요청된 메시지의 HTTP 헤더에 있는 Content-Type을 기반으로 HTTP Message Converter를 사용하여 반환 값을 HTTP response body로 변환한다.



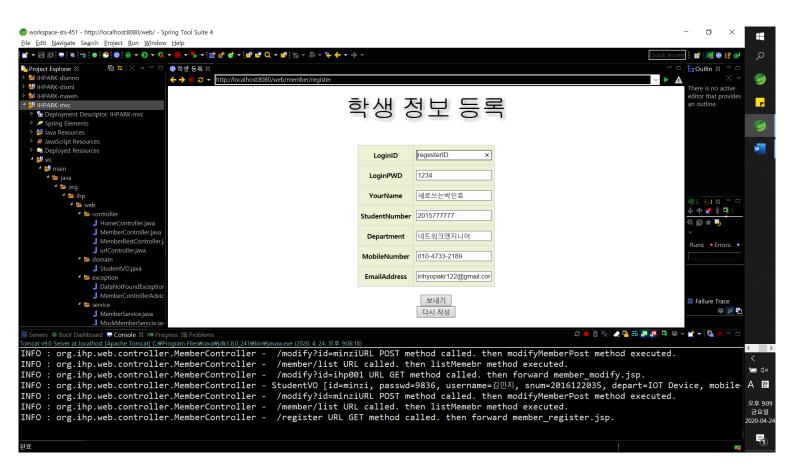
✓ HomeController.java home.jsp



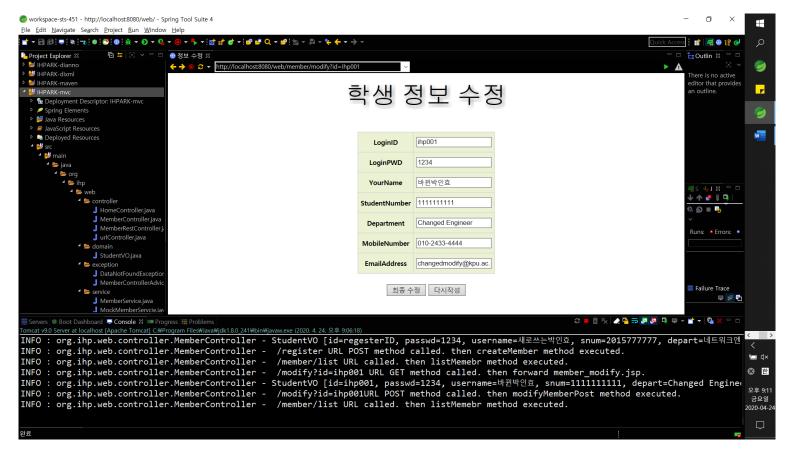
✓ MemberController.java, member_list.jsp



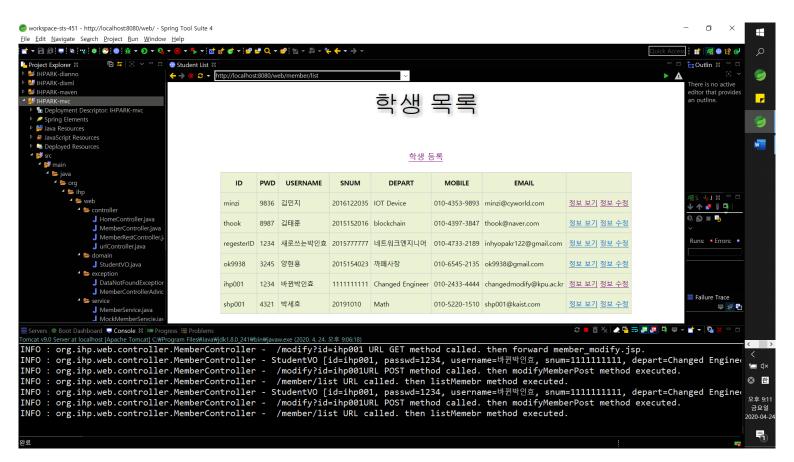
✓ MemberController.java, member_read.jsp



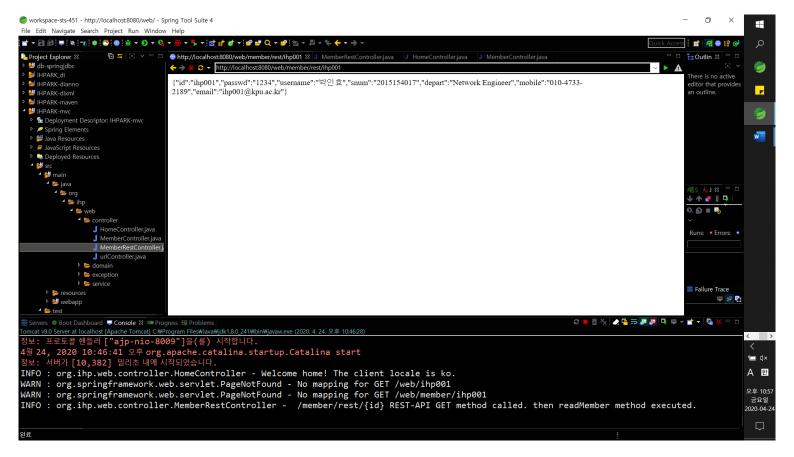
✓ MemberController.java, member_register.jsp



✓ MemberController.java, member_modify.jsp



✓ 학생 등록, 학생 정보 수정을 하고 난 뒤 학생의 목록



✓ MemberRestController.java에 아이디 ihp001 요청