

8. Spring mvc-2

1. **HttpServletRequest** 클래스
2. **@RequestParam** 어노테이션
3. 데이터(커맨드) 객체
4. **@PathVariable**



1. HttpServletRequest 클래스

□ HttpServletRequest 클래스 이용 데이터 전송 방법

```
@RequestMapping("board/confirmId")
public String confirmId(HttpServletRequest httpServletRequest,
                        Model model){
    String id=httpServletRequest.getParameter("id");
    String pw=httpServletRequest.getParameter("pw");
    model.addAttribute("id",id);
    model.addAttribute("pw",pw);
    return "board/confirmId";
}
```



The screenshot shows a web browser window with the URL `http://localhost:8181/ex/board/confirmId?id=abcd&pw=1234`. Below the URL bar, there is a form with two input fields: "ID : abcd" and "PW : 1234". To the right of the form, the HTML output is displayed, showing the rendered page structure.

URL: `http://localhost:8181/ex/board/confirmId?id=abcd&pw=1234`

Form Fields:

- ID : abcd
- PW : 1234

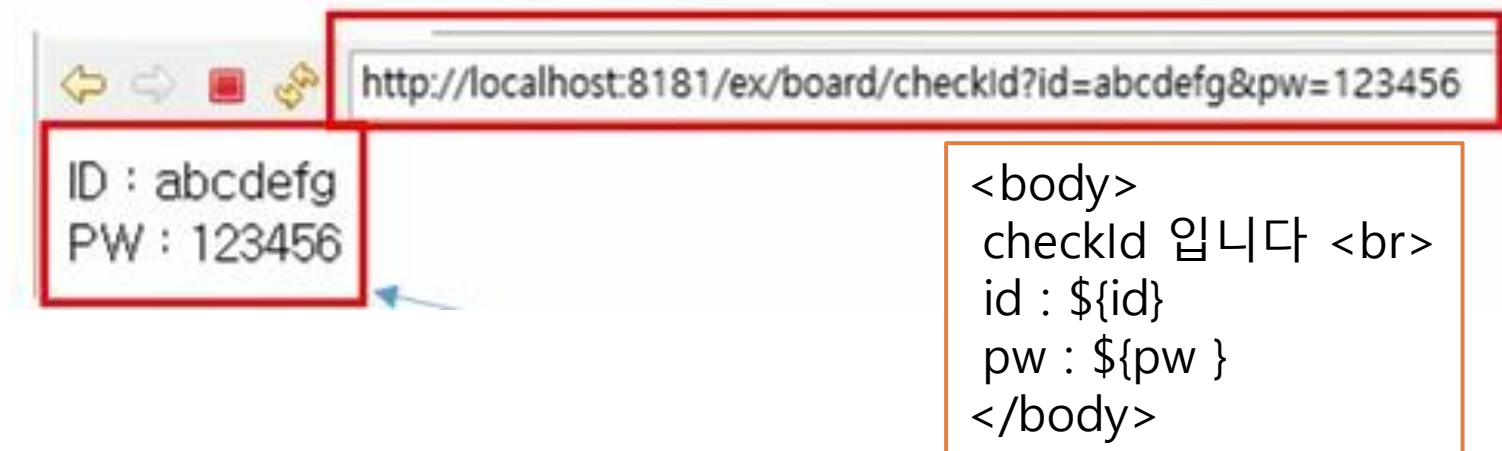
HTML Output:

```
<body>
comirmId 입니다 <br>
id : ${id}
pw : ${pw }
</body>
```

2. @RequestParam 어노테이션

- @RequestParam 어노테이션을 이용해서 데이터 전송
 - ▣ id와 pw 값 전달하지 않았을 때 결과 확인(404에러 발생)

```
@RequestMapping("board/checkId")  
public String checkId(@RequestParam("id") String id,  
    @RequestParam("pw") String pw, Model model){  
    model.addAttribute("id", id);  
    model.addAttribute("pw", pw);  
    return "board/checkId";  
}
```



3. 데이터(커맨드) 객체

□ 데이터(커맨드) 객체 이용

■ 데이터가 많을 경우 데이터 객체를 이용하여 소스코드를 줄임

■ 기존 방법



```
@RequestMapping("member/join")
public String joinData(@RequestParam("name") String name,
    @RequestParam("id") String id,
    @RequestParam("pw") String pw,
    @RequestParam("email") String email, Model model){
    Member member=new Member();
    member.setName(name);
    member.setId(id);
    member.setPw(pw);
    member.setEmail(email);
    model.addAttribute("memberInfo", member);
    return "member/join";
}
```

3. 데이터(커맨드) 객체

- ▣ 개선 방법 : 코드양 줄임



```
@RequestMapping("member/join")  
public String joinData(Member member){  
    return "member/join";  
}
```

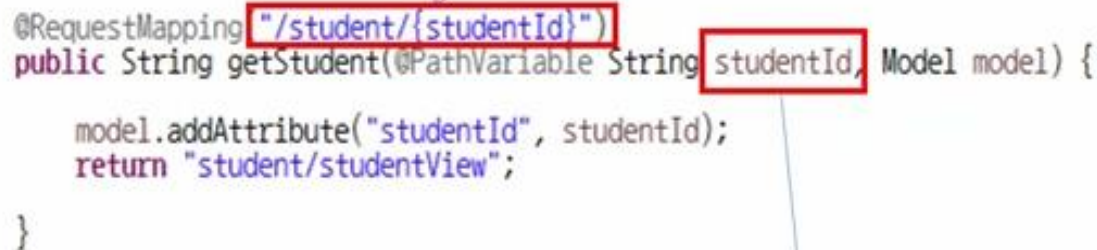
4. @PathVariable

□ @PathVariable 어노테이션

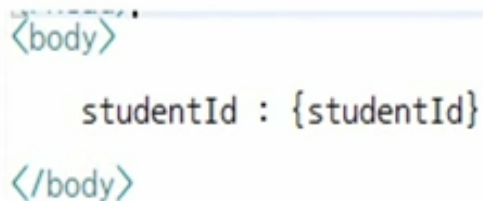
- 경로(path)에 변수를 넣어 요청 메소드에서 파라미터로 이용 가능



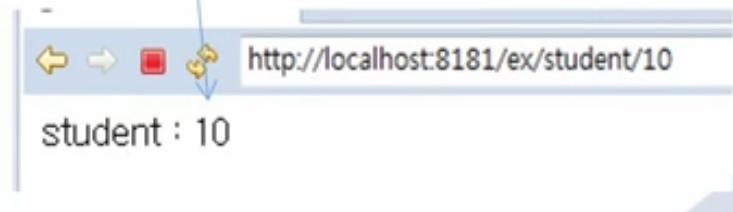
http://localhost:8181/ex/student/10



```
@RequestMapping("/student/{studentId}")
public String getStudent(@PathVariable String studentId, Model model) {
    model.addAttribute("studentId", studentId);
    return "student/studentView";
}
```



```
<body>
    studentId : {studentId}
</body>
```



http://localhost:8181/ex/student/10

student : 10