**[ 09 ] 폼데이터**

**목표 : 데이터 입력을 받아 처리하는 기법들의 이해**

**HttpServletRequest, @RequestParam @ModelAttribute @PathVariable 등을 이용**

우리가 jsp&Servlet을 할 때도 사용자가 form에 있는 데이터를 즉 input 태그나 select 태그를 이용해서 데이터를 전송을 했다. 그러면 서버에서는 그 데이터를 받아서 적합한 알맞은 로직을 수행후 다시 클라이언트에게 응답을 해주는 로직을 많이 구현해 보았다.

어차피 프로그램이라는 것은 데이터, 즉 사용자가 어떠한 요청을 하면 그 요청에 알맞은 데이터를 처리하고 그것에 대한 응답을 하는 게 거의 대부분의 프로그래밍으로 구성되어 있다.

스프링에서도 어떻게 하면 폼에 있는 데이터를 서버쪽으로 보내고 서버쪽에서 그 데이터를 어떻게 처리하는지에 대해 살펴보자.

1. HttpServletRequest 클래스를 이용해서 form 데이터를 전송받는다

* 주소창 http://localhost:8181/ch18**/member/result**?id=aa&pw=11

서버IP주소 켄텍스트path 요청경로 httpServletRequest가 가져감

1. @RequesParam 어노테이션을 이용해서 데이터를 전송받는다

@RequestMapping("member/join2")

**public** String input2(@RequestParam("name") String name,

@RequestParam("id") String id,

@RequestParam("pw") String pw,

@RequestParam("age", defaultValue="1") **int** age,

@RequestParam("email") String email,

@RequestParam("address") String address,

Model model) {

model.addAttribute("name",name);

model.addAttribute("id", id);

model.addAttribute("pw",pw);

model.addAttribute("age",age);

model.addAttribute("email",email);

model.addAttribute("address",address);

**return** "member/result1";

}

1. 데이터(커맨드) 객체를 이용하여 데이터가 많을 경우 간단하게 사용할 수 있다.

@RequestMapping("member/join3")

**public** String input3(String name,String id,String pw,

**int** age,String email,String address,

Model model) {

System.***out***.println("3방법");

model.addAttribute("name",name);

model.addAttribute("id", id);

model.addAttribute("pw",pw);

model.addAttribute("age",age);

model.addAttribute("email",email);

model.addAttribute("address",address);

**return** "member/result1";

}

**public** **class** MemberInfo {

**private** String name;

**private** String id;

**private** String pw;

**private** **int** age;

**private** String email;

}

* + 데이터 커맨드를 이용한 개선 방법 : 코드 양이 적다
  + @ModelAttribute(“모델이름선언”) Emp sch를 이용하면 자동적으로 model객체안에 attribute가 Add와 setting이 동시에 된다.

@RequestMapping("member/join4")

**public** String input4(MemberDto memberDto, Model model) {

//스프링프레임워크가 자동으로 MemberInfo 객체 생성 후 setter를 이용하여 속성을 setting

model.addAttribute("member",memberDto);

**return** "member/result4";

}

@RequestMapping("member/join5")

**public** String input5(@ModelAttribute("member") MemberDto memberDto,

Model model) {

//스프링프레임워크가 자동으로 MemberDto의 속성에 setting 하고 model에 담는다

//커맨드 객체의 이름이 가독성이 적으면, @ModelAttribute 어노테이션을 이용해서 가독성이

높고 간결한 이름으로 바꿀 수 있다

**return** "member/result4";

}

@RequestMapping("member/join6")

**public** String input6(MemberDto memberDto,

Model model) {

//스프링프레임워크가 자동으로 MemberDto의 속성에 setting 하고 model에 담는다

(memberDto라는 속성명으로)

**return** "member/result6";

}

1. **@ModelAttribute :** 모든 요청경로에 공통으로 필요한 model attribute가 있을 경우 사용

* @ModelAttribute(“모델이름선언”)을 이용하면 자동적으로 모든 요청경로를 수행하기전 model객체안에 attribute Add와 setting이 동시에 된다. @ModelAttribute의 용도는 크게 2가지가 있다.

@Controller

**public** **class** MyController {

**// 모든 요청경로를 수행할 때마다 list()메소드는 수행한다. list라는 model 어트리뷰트에 add된다**

@ModelAttribute("list")

**public** ArrayList<String> list(){

ArrayList<String> list = **new** ArrayList<String>();

list.add("테스트1"); list.add("테스트2");

**return** list;

}

@ModelAttribute("cnt")

**public** **int** cnt(){

**return** 5;

}

@RequestMapping("/")

**public** String input(Model model) {

// ArrayList<String> list = new ArrayList<String>();

// list.add("테스트1"); list.add("테스트2");

// model.addAttribute("list",list);

**return** "input";

}

1. @PathVariable을 이용하여 경로(path)에 변수를 넣어 요청 메소드에서 파라미터로 이용할 수 있다.

@RequestMapping("student/\*")

// \* : 0개 또는 그 이상의 글자

// ? : 1개 글자

//\*\* : 0개 또는 그 이상의 디렉토리 경로

**public** String student(HttpServletRequest httpServletRequest, Model model) {

String uri = httpServletRequest.getRequestURI();

String id = uri.substring(uri.lastIndexOf("/")+1);

model.addAttribute("id",id);

model.addAttribute("pw","안 넘어왔어");

**return** "member/result";

}

* + 개선방법

@RequestMapping("/student/{id}")

**public** String getStudent(@PathVariable String id, Model model){

// model.addAttribute("id", id);

**return** " member/result ";

}

-- 총예제 –

(web.xml)

<filter>

<filter-name>encodingFilter</filter-name>

<filter-class>org.springframework.web.filter.CharacterEncodingFilter</filter-class>

<init-param>

<param-name>encoding</param-name>

<param-value>utf-8</param-value>

</init-param>

<init-param>

<param-name>forceEncoding</param-name>

<param-value>true</param-value>

</init-param>

</filter>

<filter-mapping>

<filter-name>encodingFilter</filter-name>

<url-pattern>/\*</url-pattern>

</filter-mapping>

**(input.jsp)**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h2>ModelAttribute를 통해 cnt, list</h2>

<h3>cnt : ${cnt }</h3>

<h3>list : ${list }</h3>

<hr>

<%-- <form action="${conPath }/member/join1" method="post"> --%>

<%-- <form action="${conPath }/member/join2"> --%>

<%-- <form action="${conPath }/member/join3"> --%>

<%-- <form action="${conPath }/member/join4"> --%>

<%-- <form action="${conPath }/member/join5"> --%>

<form action=*"*${conPath }*/member/join6"*>

<table>

<caption>개인정보 입력</caption>

<tr><th>이름</th><td><input type=*"text"* name=*"name"*></td></tr>

<tr><th>ID</th><td><input type=*"text"* name=*"id"*></td></tr>

<tr><th>PW</th><td><input type=*"password"* name=*"pw"*></td></tr>

<tr><th>나이</th><td><input type=*"number"* name=*"age"*></td></tr>

<tr><th>메일</th><td><input type=*"email"* name=*"email"*></td></tr>

<tr><th>주소</th><td><input type=*"text"* name=*"address"*></td></tr>

<tr><td colspan=*"2"*><input type=*"submit"*></td></tr>

</table>

</form>

<hr>

<a href=*"*${conPath }*/studentId/aaa"*>studentId를 aaa로 전송</a><br>

<a href=*"*${conPath }*/studentId/홍길동"*>studentId를 홍길동으로 전송</a>

</body>

</html>

**(MyController.java)**

@Controller

**public** **class** MyController {

@ModelAttribute("cnt")

**public** **int** cnt() {

**return** 5;

}

@ModelAttribute("list")

**public** ArrayList<String> list(){

ArrayList<String> list = **new** ArrayList<String>();

list.add("test1"); list.add("test2"); list.add("test3");

**return** list;

}

@RequestMapping("/")

**public** String input(Model model) {

//model.addAttribute("cnt",5);

**return** "input";

}

// @RequestMapping("studentId/\*")

// public String studentId(HttpServletRequest request, Model model) throws UnsupportedEncodingException {

// // \* : 0자 이상의 글자

// // \*\* : 0개 이상의 디렉토리 경로

// // ? : 1개 이상의 글자

// String uri = request.getRequestURI(); // ch08/studentId/aaa

// String id = uri.substring(uri.lastIndexOf("/")+1);

// id = URLDecoder.decode(id, "utf-8");

// model.addAttribute("id", id);

// return "studetId";

// }

@RequestMapping("studentId/{id}")

**public** String studentId(@PathVariable("id") String id, Model model) {

// model.addAttribute("id", id);

**return** "studetId";

}

}

**(MemberController.java)**

**import** com.lec.ch08.vo.MemberDto;

@Controller

@RequestMapping("member")

**public** **class** MemberController {

@ModelAttribute("cnt")

**public** **int** cnt() {

**return** 5;

}

@ModelAttribute("list")

**public** ArrayList<String> list(){

ArrayList<String> list = **new** ArrayList<String>();

list.add("test1"); list.add("test2"); list.add("test3");

**return** list;

}

@RequestMapping("join1")

**public** String join1(HttpServletRequest request, Model model) {

String name = request.getParameter("name");

String id = request.getParameter("id");

String pw = request.getParameter("pw");

**int** age = Integer.*parseInt*(request.getParameter("age"));

String email = request.getParameter("email");

String address = request.getParameter("address");

model.addAttribute("name", name);

model.addAttribute("id", id);

model.addAttribute("pw", pw);

model.addAttribute("age", age);

model.addAttribute("email", email);

model.addAttribute("address", address);

**return** "member/result1";

}

@RequestMapping("join2")

**public** String join2(@RequestParam("name") String membername,

@RequestParam("id") String memberid,

@RequestParam("pw") String pw,

@RequestParam("age") **int** age,

@RequestParam("email") String email,

@RequestParam("address") String address,

Model model) {

model.addAttribute("name", membername);

model.addAttribute("id", memberid);

model.addAttribute("pw", pw);

model.addAttribute("age", age);

model.addAttribute("email", email);

model.addAttribute("address", address);

**return** "member/result1";

}

@RequestMapping("join3")

**public** String join3(String name, String id, String pw,

**int** age, String email, String address, Model model) {

model.addAttribute("name", name);

model.addAttribute("id", id);

model.addAttribute("pw", pw);

model.addAttribute("age", age);

model.addAttribute("email", email);

model.addAttribute("address", address);

**return** "member/result1";

}

@RequestMapping("join4")

// public String join4(String name, String id, String pw,

// int age, String email, String address, Model model) {

// MemberDto memberDto = new MemberDto();

// memberDto.setName(name);

// memberDto.setId(id);

// memberDto.setPw(pw);

// memberDto.setAge(age);

// memberDto.setEmail(email);

// memberDto.setAddress(address);

// model.addAttribute("member", memberDto);

// return "member/result4";

// }

**public** String join4(MemberDto memberDto, Model model) {

model.addAttribute("member", memberDto);

**return** "member/result4";

}

@RequestMapping("join5")

**public** String join5(@ModelAttribute("member") MemberDto memberDto) {

**return** "member/result4";

}

@RequestMapping("join6")

**public** String join6(MemberDto memberDto) {

// model.addAttribute("memberDto", memberDto);

**return** "member/result6";

}

}

**(member/result1.jsp)**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h1>ModelAttribute를 통해 cnt, list</h1>

<h2>cnt : ${cnt }</h2>

<h2>list : ${list }</h2>

<hr>

<h1>result1 : 개인정보</h1>

<h2>이름 : ${name }</h2>

<h2>ID : ${id }</h2>

<h2>PW : ${pw }</h2>

<h2>나이 : ${age }</h2>

<h2>메일 : ${email }</h2>

<h2>주소 : ${address }</h2>

<button onclick="history.back()">뒤로가기</button>

<button onclick=*"location.href='*${conPath}*/'"*>처음부터 다시 시작하기</button>

</body>

</html>

**(member/result4.jsp)**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h1>ModelAttribute를 통해 cnt, list</h1>

<h2>cnt : ${cnt }</h2>

<h2>list : ${list }</h2>

<hr>

<h1>result1 : 개인정보</h1>

<h2>이름 : ${member.name }</h2>

<h2>ID : ${member.id }</h2>

<h2>PW : ${member.pw }</h2>

<h2>나이 : ${member.age }</h2>

<h2>메일 : ${member.email }</h2>

<h2>주소 : ${member.address }</h2>

<button onclick="history.back()">뒤로가기</button>

<button onclick=*"location.href='*${conPath}*/'"*>처음부터 다시 시작하기</button>

</body>

</html>

**(member/result6.jsp)**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h1>ModelAttribute를 통해 cnt, list</h1>

<h2>cnt : ${cnt }</h2>

<h2>list : ${list }</h2>

<hr>

<h1>result1 : 개인정보</h1>

<h2>이름 : ${memberDto.name }</h2>

<h2>ID : ${memberDto.id }</h2>

<h2>PW : ${memberDto.pw }</h2>

<h2>나이 : ${memberDto.age }</h2>

<h2>메일 : ${memberDto.email }</h2>

<h2>주소 : ${memberDto.address }</h2>

<button onclick="history.back()">뒤로가기</button>

<button onclick=*"location.href='*${conPath}*/'"*>처음부터 다시 시작하기</button>

</body>

</html>

**(studentId.jsp)**

<%@ page language=*"java"* contentType=*"text/html; charset=UTF-8"*

pageEncoding=*"UTF-8"*%>

<%@ taglib prefix=*"c"* uri=*"http://java.sun.com/jsp/jstl/core"*%>

<c:set var=*"conPath"* value=*"*${pageContext.request.contextPath }*"*/>

<!DOCTYPE html>

<html>

<head>

<meta charset=*"UTF-8"*>

<title>Insert title here</title>

<link href=*"*${conPath }*/css/style.css"* rel=*"stylesheet"*>

</head>

<body>

<h1>ModelAttribute를 통해 cnt, list</h1>

<h2>cnt : ${cnt }</h2>

<h2>list : ${list }</h2>

<hr>

<h1>요청경로로 전달 받은 값</h1>

<h2>ID : ${id }</h2>

</body>

</html>

(style.jsp)

@charset *"UTF-8"*;

**body**{background-color: *#FFFFDD*;}

**span**{color:*blue*;}

|  |  |
| --- | --- |
| **문제 1**  **(수행내용)** | **Spring framework를 이용하여 아래 예시와 같은 UI 화면을 스프링 폼데이터로 처리하도록 구현하시오.**  **<세부 요구사항>**  ▪ Spring 프로젝트 내의 web.xml에 한글처리를 추가하고, css를 추가하기 위한 servlet-context.xml에 설정을 추가한다.  ▪ @ModelAttribute를 이용하여 현재 총 몇 명인지를 모든 페이지에 출력한다.  ▪ 컨트롤러에서 처리할 요청 경로와 method방식은 아래의 예시화면과 같이 한다.  ▪ 클라이언트에게 제공될 View 단 생성(**student/input.jsp, student/result.jsp**) |
| **과제물결과**  **(예시)** |  |
| **과제물제출**  **및 보관** | ▪ 파일명은 uitest 프로젝트와 실행화면(입력화면.png, 결과화면.png)  ▪ 결과물 제출 : 위의 프로젝트 폴더와 실행화면이 압축된 파일 : 훈련생성명\_인터페이스구현평가.zip  ▪ 평가자료 보관 방법 : 파일로 보관 |
| **평가항목** | ▪ lombok을 활용한 Dto 생성  ▪ web.xml과 servlet-context.xml에 올바른 설정  ▪ controller에서 @ModelAttribute를 이용한 구현  ▪ controller에서 같은 요청 경로를 다른 method로 처리하는 방식 구현  ▪ View UI 페이지 구현 |