**입력 폼 #2**

2021-10-17

이승진

**학습목표**

입력 폼(form) 구현하기

입력 폼에 입력된 데이터 꺼내기

입력 데이터 오류 검사하기

리다이렉트(redirect)

**목차**

[1. 개요 2](#_Toc69241883)

[1) session 2](#_Toc69241884)

[2) 리다이렉트 (redirect) 3](#_Toc69241885)

[3) 리다이렉트 과정 설명 #1 4](#_Toc69241886)

[4) 리다이렉트 과정 설명 #2 - 로그인 성공 5](#_Toc69241887)

[5) 로그인 실패 6](#_Toc69241888)

[6) model attribute 전달 7](#_Toc69241889)

[2. 로그인 구현 #1 8](#_Toc69241890)

[1) Form4Controller.java 8](#_Toc69241891)

[2) form4/login1.jsp 9](#_Toc69241892)

[3) 실행 과정 10](#_Toc69241893)

[3. 로그인 구현 #2 12](#_Toc69241894)

[1) Form4Controller.java 수정 12](#_Toc69241895)

[2) form4/login2.jsp 생성 13](#_Toc69241896)

[4. 로그인 구현 #3 14](#_Toc69241897)

[1) Form4Controller.java 수정 14](#_Toc69241898)

[2) src/main/java/net/skhu/controller/Form4Controller.java 14](#_Toc69241899)

[3) form4/login3.jsp 생성 16](#_Toc69241900)

[4) form4/login\_success.jsp 생성 17](#_Toc69241901)

[5. 회원가입 구현 #1 19](#_Toc69241902)

[1) Form5Controller.java 19](#_Toc69241903)

[2) form5/register1.jsp 20](#_Toc69241904)

[3) form5/register\_success1.jsp 22](#_Toc69241905)

[4) 실행 23](#_Toc69241906)

[6. 회원가입 구현 #2 24](#_Toc69241907)

[1) Member.java 생성 24](#_Toc69241908)

[2) Form5Controller.java 26](#_Toc69241909)

[3) form5/register2.jsp 28](#_Toc69241910)

[4) register\_success2.jsp 30](#_Toc69241911)

[7. 연습문제 31](#_Toc69241912)

[1) Student.java 31](#_Toc69241913)

[2) student/edit 32](#_Toc69241914)

# 개요

## session

session은 서버에 있는, 웹브라우저의 개인 사물함에 비유할 수 있다.

웹브라우저가 서버에 처음 요청(HTTP request)을 전달하면,

그 웹브라우저의 개인 사물함(session)이 서버의 메모리에 생성된다.

로그아웃 메뉴가 클릭되어 서버에 로그아웃 요청을 보내거나,

웹브라우저에서 아무런 동작이 없어 서버에 아무런 요청을 보내지 않고 타임아웃 시간을 초과하면,

서버 메모리의 개인 사물함(sessoin)이 제거된다.

### session 객체 사용법

액션 메소드의 파라미터 변수 목록에 HttpSession session 변수가 있으면,

자동으로 session 객체가 전달된다.

session에 아무 객체나 넣을 수 있다.

|  |
| --- |
| session.setAttribute("키", 데이터); |

예:

|  |
| --- |
| User user = new User();  session.setAttribute("key1", user); |

꺼내기

|  |
| --- |
| User u = (User)session.getAttribute("key1"); |

넣고 꺼낼 때 키 부분이 일치해야 한다.

세션에 등록한 데이터를 뷰에서 꺼내서 출력하기

|  |
| --- |
| ${ key1 } |

model attributge 출력 방법과 session attribute 출력 방법이 동일하다.

## 리다이렉트 (redirect)

로그인 버튼을 눌렀을 때, 로그인에 실패하면, 웹브라우저 창에 로그인 화면이 다시 나타난다.

로그인에 성공하면, 다른 페이지로 넘어간다.

게시글 편집 화면에서 저장 버튼을 누르면, 게시글 목록 페이지로 넘어간다.

이렇게 다른 페이지로 넘어가는 기능을 리다이렉트(redirect) 라고 부른다.

### 리다이렉트 메타 태그 (redirect meta tag)

<meta http-equiv="refresh" content="**0**; url=**http://www.google.com/**" />

이 태그가 리다이렉트 메타 태그이다.

웹페이지에 이 리다이렉트 메타 태그가 포함되어 있으면,

웹 브라우저는 **0**초 후에 **http://www.google.com/** URL로 넘어간다.

### redirect1.html

|  |
| --- |
| <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <meta http-equiv="refresh" content="0; url=http://www.google.com/" />  </head>  <body>  웹브라우저 창에, 이 페이지가 로드(load)되자 마자  http://www.google.com/ URL을 즉시 요청하게 된다.  </body>  </html> |

위 html 파일을 웹브라우저로 열어보자.

위 파일을 열자마자 http://www.google.com URL로 넘어갈 것이다. (리다이렉트)

### redirect2.html

|  |
| --- |
| <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />  <meta http-equiv="refresh" content="5; url=http://www.naver.com/" />  </head>  <body>  웹브라우저 창에, 이 페이지가 로드(load)되고 5초 후에  http://www.naver.com/ URL을 요청하게 된다.  </body>  </html> |

위 html 파일을 웹브라우저로 열어보자.

## 리다이렉트 과정 설명 #1

액션 메소드의 리턴값이 뷰의 이름이면,

액션 메소드가 리턴된 후 그 뷰가 실행되고,

뷰의 실행 결과 출력이 웹브라우저로 전송된다. (HTTP response)

액션 메소드의 리턴값이 리다이렉트 URL이면,

뷰가 실행되지 않고,

리다이렉트 메타 태그가 웹브라우저로 전송된다. (HTTP response)

리다이렉트 메타 태그에 리다이렉트URL이 포함된다.

|  |
| --- |
| return "redirect:리다이렉트할URL"; |

저장이나 삭제 작업이 성공한 후 목록 페이지로 넘어가는 것이 보통이다.

저장이나 삭제 작업을 구현한 액션 메소드에서, 리다이렉트 메타 태그를 리턴한다.

리다이렉트 메타 태그를 받은 웹브라우저는, 이 메타 태그에 포함된 리다이렉트URL을

즉시 서버에 다시 요청하게 된다.



(1) 사용자가 수정 화면에서 저장 버튼을 누르면

저장 액션 메소드 URL이 서버에 요청된다.

서버에서 저장 액션 메소드가 실행된다.

(2) 저장 액션 메소드에서 저장 작업이 성공하면, 다음과 같이 리턴한다.

return "redirect:목록액션메소드URL";

웹브라우저로 리다이렉트 메타 태그가 전송된다.

(3) 웹브라우저는 리다이레트 메타 태그를 받자마자 즉시 목록액션메소드URL을 서버에 요청한다.

서버에서 목록 액션 메소드가 실행된다.

목록 액션 메소드는 목록 뷰 파일의 이름을 리턴한다.

목록 뷰 파일이 실행된다.

(4) 뷰 파일의 실행결과 출력이 웹브라우저로 전송된다.

목록 페이지 html 태그들이 웹브라우저로 전송된다.

웹브라우저 창에는 목록 페이지가 그려진다.

웹브라우저 창의 주소칸은 가장 최근에 요청한 URL이 채워진다.

즉 목록 페이지 URL이 채워진다.

위 과정이 너무 빨리 진행되기 때문에,

사용자는 수정 화면에서 저장 버튼을 누르자마자, 목록 페이지로 바로 넘어간다고 느껴진다.

## 리다이렉트 과정 설명 #2 - 로그인 성공



(1) 사용자가 웹브라우저에서 로그인 URL을 입력하고 엔터키를 누른다.

로그인 URL 요청이 서버에 전송된다.

이 요청은 GET 방식 요청이다.

GET 방식 로그인 액션 메소드가 서버에서 실행된다.

이 액션 메소드는 로그인 뷰 파일의 이름을 리턴한다.

(2) 로그인 뷰 파일이 실행된다.

이 파일의 실행 결과 출력이 웹브라우저로 전송된다.

이 파일의 실행 결과 출력은 로그인 페이지 html 태그이다.

웹브라우저 창에 로그인 페이지가 그려진다.

웹브라우저 창의 주소칸은 가장 최근에 요청한 URL이 채워진다.

즉 로그인 URL이 채워진다.

(3) 사용자가 아이디 패스워드를 입력하고 로그인 버튼을 클릭하면,

로그인 URL 요청이 다시 서버에 전송된다.

이 요청은 POST 방식 요청이다.

이 요청에 들어있는 request parameter는, 입력된 아이디 패스워드이다

POST 방식 로그인 액션 메소드가 서버에서 실행된다.

이 액션 메소드는 다음과 같이 리턴한다.

return "redirect:로그인성공URL";

(4) 리다이렉트 메타 태그가 웹브라우저로 전송된다.

이 메타 태그에는 로그인 성공 액션메소드에 대한 URL이 들어있다.

(5) 웹브라우저는 리다이레트 메타 태그를 받자마자 즉시 로그인 성공 URL을 서버에 요청한다.

GET 방식 로그인 성공 액션 메소드가 서버에서 실행된다.

이 액션 메소드는 로그인 성공 뷰 파일의 이름을 리턴한다.

로그인 성공 뷰 파일이 실행된다.

(6) 뷰 파일의 실행결과 출력이 웹브라우저로 전송된다.

로그인 성공 페이지 html 태그들이 웹브라우저로 전송된다.

웹브라우저 창에는 로그인 성공 페이지가 그려진다.

그리고 웹브라우저 창의 주소칸에는, 가장 최근에 요청한 URL이 채워진다.

즉 로그인 성공 페이지 URL이 채워진다.

위 과정이 너무 빨리 진행되기 때문에,

사용자는 수정 화면에서 저장 버튼을 누르자마자, 목록 페이지로 바로 넘어간다고 느껴진다.

## 로그인 실패



로그인 실패의 경우, POST 방식 로그인 액션 메소드는 로그인 뷰의 이름을 다시 리턴한다.

그러면 다시 로그인 뷰 파일이 실행되고,

웹브라우저 창에 다시 로그인 페이지가 출력된다.

## model attribute 전달



model attribue는 액션 메소드에서 뷰에 전달하는 데이터이다.

액션 메소드 바로 뒤를 이어 실행되는 뷰에게만 전달할 수 있다.

그 이상으로 멀리 전달될 수는 없다.

어떤 액션 메소드에서, 다음 액션 메소드로 데이터를 전달하는 방법 중의 하나는

session attribute로 전달하는 것이다.



# 로그인 구현 #1

## Form4Controller.java

src/main/java/net/skhu/controller/Form4Controller.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27 | package net.skhu.controller;  import java.util.Date;  import org.springframework.stereotype.Controller;  import org.springframework.ui.Model;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PostMapping;  @Controller  public class Form4Controller {  @GetMapping("form4/login1")  public String login1(Model model) {  model.addAttribute("now", new Date());  return "form4/login1";  }  @PostMapping("form4/login1")  public String login1(Model model, String userid, String password, Boolean autologin) {  model.addAttribute("now", new Date());  model.addAttribute("userid", userid);  model.addAttribute("password", password);  model.addAttribute("autologin", autologin);  return "form4/login1";  }  } |

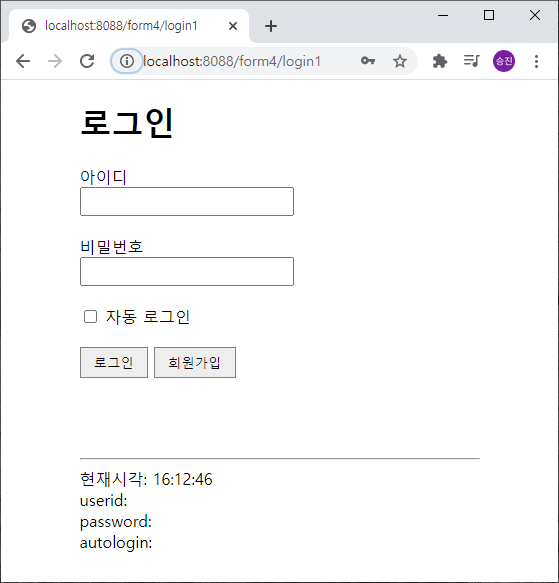
## form4/login1.jsp

src/main/webapp/WEB-INF/views/form/login1.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <style>  div.container { width: 400px; margin: 20px auto; }  form div { margin-bottom: 20px; }  label { display: block; }  input[type=text], input[type=password] { width: 200px; padding: 5px; }  .btn { border: 1px solid gray; text-decoration: none; padding: 0.4em 1em;  color: black; background-color: #eee; cursor: pointer; font-size: 10pt; }  hr { margin-top: 80px; }  </style>  </head>  <body>  <div class="container">  <form method="post">  <h1>로그인</h1>  <div>  <label>아이디</label>  <input type="text" class="text" name="userid" />  </div>  <div>  <label>비밀번호</label>  <input type="password" class="text" name="password" />  </div>  <div>  <label><input type="checkbox" name="autologin" /> 자동 로그인</label>  </div>  <button type="submit" class="btn">로그인</button>  <a href="/form5/register1" class="btn">회원가입</a>  </form>  <hr />  현재시각: <fmt:formatDate pattern="HH:mm:ss" value="${ now }" /> <br />  userid: ${ userid } <br />  password: ${ password } <br />  autologin: ${ autologin } <br />  </div>  </body>  </html> |

## 실행 과정

http://localhost:8088/form4/login1.jsp



http://localhost:8088/form4/login1

URL을 웹브라우저창의 주소칸에 입력하고 엔터키를 누르면

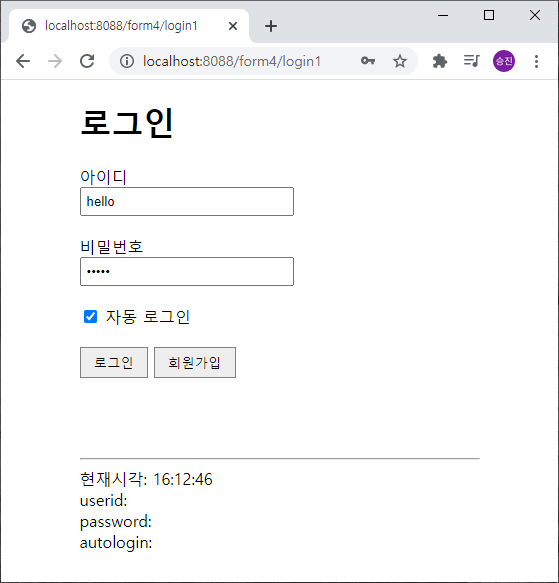
GET 방식 요청이 서버에 전송되어, GET 방식 login1 액션 메소드가 실행된다.

이 액션 메소드는 현재 시각을 모델에 추가한다.

login1.jsp 뷰 파일이 실행된다.

이때 모델에는 현재 시각만 들어있을 뿐이고, userid, password, autoloin 값은 들어있지 않다.

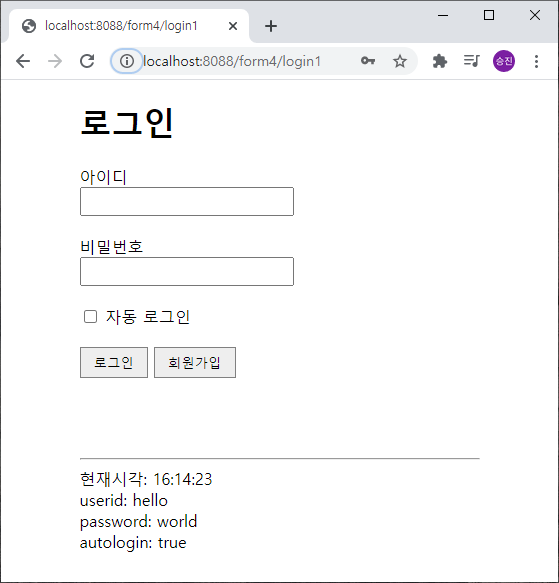
위 로그인 폼에서 아이디 비밀번호를 입력하고 자동 로그인을 체크하고 로그인 버튼을 클릭하자.



POST 방식 요청이 서버에 전송되어, POST 방식 login1 액션 메소드가 실행된다.

이 액션 메소드는 현재 시각을 모델에 추가한다.

userid, password, autoloin 값모 모델에 추가한다.



로그인 버튼을 누르면,

이전에 입력했던 입력 값이 초기화된다.

이 문제를 해결하자.

# 로그인 구현 #2

## Form4Controller.java 수정

src/main/java/net/skhu/controller/Form4Controller.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42 | package net.skhu.controller;  import java.util.Date;  import org.springframework.stereotype.Controller;  import org.springframework.ui.Model;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PostMapping;  @Controller  public class Form4Controller {  @GetMapping("form4/login1")  public String login1(Model model) {  model.addAttribute("now", new Date());  return "form4/login1";  }  @PostMapping("form4/login1")  public String login1(Model model, String userid, String password, Boolean autologin) {  model.addAttribute("now", new Date());  model.addAttribute("userid", userid);  model.addAttribute("password", password);  model.addAttribute("autologin", autologin);  return "form4/login1";  }  @GetMapping("form4/login2")  public String login2(Model model) {  model.addAttribute("now", new Date());  return "form4/login2";  }  @PostMapping("form4/login2")  public String login2(Model model, String userid, String password, Boolean autologin) {  model.addAttribute("now", new Date());  model.addAttribute("userid", userid);  model.addAttribute("password", password);  model.addAttribute("autologin", autologin);  return "form4/login2";  }  } |

login1 액션 메소드와 login2 액션 메소드를 비교하면,

리턴되는 뷰의 이름만 다르다.

## form4/login2.jsp 생성

src/main/webapp/WEB-INF/views/form/login2.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <style>  div.container { width: 400px; margin: 20px auto; }  form div { margin-bottom: 20px; }  label { display: block; }  input[type=text], input[type=password] { width: 200px; padding: 5px; }  .btn { border: 1px solid gray; text-decoration: none; padding: 0.4em 1em;  color: black; background-color: #eee; cursor: pointer; font-size: 10pt; }  hr { margin-top: 80px; }  </style>  </head>  <body>  <div class="container">  <form method="post">  <h1>로그인</h1>  <div>  <label>아이디</label>  <input type="text" class="text" name="userid" value="${ userid }" />  </div>  <div>  <label>비밀번호</label>  <input type="password" class="text" name="password" />  </div>  <div>  <label>  <input type="checkbox" name="autologin" ${ autologin ? "checked" : "" } /> 자동 로그인  </label>  </div>  <button type="submit" class="btn">로그인</button>  <a href="/form5/register1" class="btn">회원가입</a>  </form>  <hr />  현재시각: <fmt:formatDate pattern="HH:mm:ss" value="${ now }" /> <br />  userid: ${ userid } <br />  password: ${ password } <br />  autologin: ${ autologin } <br />  </div>  </body>  </html> |

login1.jsp 파일과 login2.jsp 파일을 비교하면,

노란색으로 칠한 부분만 다르다.

이전에 입력했던 입력 값이 초기화되지 않기 위한 코드를 추가하였다.

# 로그인 구현 #3

## Form4Controller.java 수정

## src/main/java/net/skhu/controller/Form4Controller.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77 | package net.skhu.controller;  import java.util.Date;  import javax.servlet.http.HttpSession;  import org.springframework.stereotype.Controller;  import org.springframework.ui.Model;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PostMapping;  @Controller  public class Form4Controller {  @GetMapping("form4/login1")  public String login1(Model model) {  model.addAttribute("now", new Date());  return "form4/login1";  }  @PostMapping("form4/login1")  public String login1(Model model, String userid, String password, Boolean autologin) {  model.addAttribute("now", new Date());  model.addAttribute("userid", userid);  model.addAttribute("password", password);  model.addAttribute("autologin", autologin);  return "form4/login1";  }  @GetMapping("form4/login2")  public String login2(Model model) {  model.addAttribute("now", new Date());  return "form4/login2";  }  @PostMapping("form4/login2")  public String login2(Model model, String userid, String password, Boolean autologin) {  model.addAttribute("now", new Date());  model.addAttribute("userid", userid);  model.addAttribute("password", password);  model.addAttribute("autologin", autologin);  return "form4/login2";  }  @GetMapping("form4/login3")  public String login3(Model model) {  model.addAttribute("now", new Date());  return "form4/login3";  }  @PostMapping("form4/login3")  public String login3(Model model, String userid, String password, Boolean autologin, HttpSession session) {  String errorMsg = null;  if (userid == null || userid.length() == 0)  errorMsg = "사용자 아이디를 입력하세요";  else if (password == null || password.length() == 0)  errorMsg = "비밀번호를 입력하세요";  else if (userid.equals(password) == false)  errorMsg = "비밀번호 불일치";  else {  session.setAttribute("userid", userid);  session.setAttribute("autologin", autologin);  return "redirect:login\_success";  }  model.addAttribute("now", new Date());  model.addAttribute("userid", userid);  model.addAttribute("password", password);  model.addAttribute("autologin", autologin);  model.addAttribute("errorMsg", errorMsg);  return "form4/login3";  }  @GetMapping("form4/login\_success")  public String login\_success(Model model) {  return "form4/login\_success";  }  } |

POST 방식의 login3 메소드는

입력된 아이디와 비밀번호가 일치하지 않으면, 뷰의 이름을 리턴한다.

return "form4/login3";

"form4/login3" 뷰에 전달할 데이터는 model에 저장한다. (model attribute)

입력된 아이디와 비밀번호가 일치하면, 리다이렉트 URL을 리턴한다.

reutrn "redirect:login\_success";

login\_success 액션 메소드에 전달할 데이터는 session에 저장한다. (session attribute)

## form4/login3.jsp 생성

src/main/webapp/WEB-INF/views/form/login3.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/fmt" prefix="fmt" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <style>  div.container { width: 400px; margin: 20px auto; }  form div { margin-bottom: 20px; }  label { display: block; }  input[type=text], input[type=password] { width: 200px; padding: 5px; }  .btn { border: 1px solid gray; text-decoration: none; padding: 0.4em 1em;  color: black; background-color: #eee; cursor: pointer; font-size: 10pt; }  hr { margin-top: 80px; }  .error { color: red; margin: 5px; }  </style>  </head>  <body>  <div class="container">  <form method="post">  <h1>로그인</h1>  <div>  <label>아이디</label>  <input type="text" class="text" name="userid" value="${ userid }" />  </div>  <div>  <label>비밀번호</label>  <input type="password" class="text" name="password" />  </div>  <div>  <label>  <input type="checkbox" name="autologin" ${ autologin ? "checked" : "" } /> 자동 로그인  </label>  </div>  <button type="submit" class="btn">로그인</button>  <a href="/form5/register1" class="btn">회원가입</a>  </form>  <div class="error">${ errorMsg }</div>  <hr />  현재시각: <fmt:formatDate pattern="HH:mm:ss" value="${ now }" /> <br />  userid: ${ userid } <br />  password: ${ password } <br />  autologin: ${ autologin } <br />  </div>  </body>  </html> |

노란색으로 칠한 부분을 제외하면, login2.jsp 구현과 동일하다.

## form4/login\_success.jsp 생성

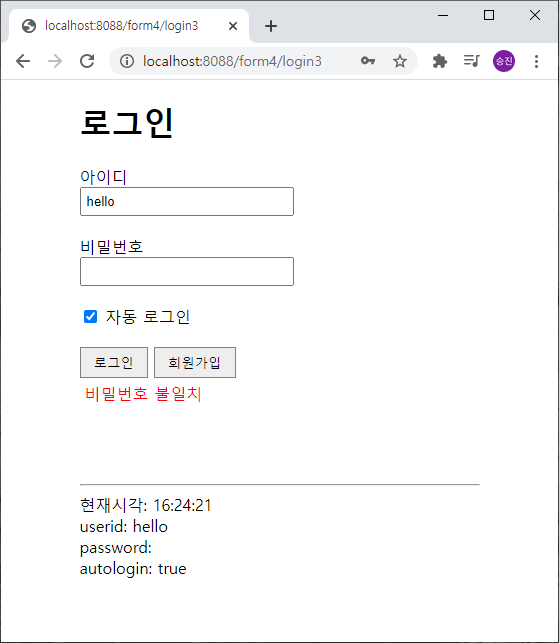
src/main/webapp/WEB-INF/views/form/login\_success.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <style>  body { font-family: 굴림체; }  div.container { width: 310px; margin: 20px auto; }  table { width: 300px; border-collapse: collapse; }  thead tr { background-color: #eee; }  td, th { border: 1px solid #aaa; padding: 5px; }  td:nth-child(1) { background-color: #eee; }  </style>  </head>  <body>  <div class="container">  <h1>로그인 성공</h1>  <table>  <tr>  <td>사용자 아이디</td>  <td>${ userid }</td>  </tr>  <tr>  <td>자동 로그인</td>  <td>${ autologin }</td>  </tr>  </table>  </div>  </body>  </html> |

${ userid }

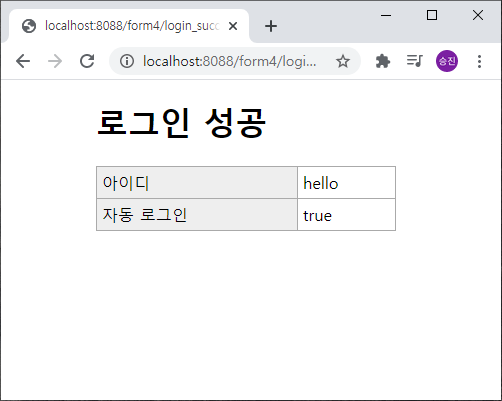
`

model attribuet 값 뿐만 아니라 session attribuet의 값도 위와 같은 방법으로 출력할 수 있다.



아이디, 비밀번호가 동일한 문자열이 아니면,

위와 같은 입력 화면이 다시 출력된다.



아이디, 비밀번화가 동일한 문자열이면,

위와 같은 로그인 성공 URL로 리다이렉트한다.

# 회원가입 구현 #1

## Form5Controller.java

src/main/java/net/skhu/controller/Form5Controller.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55 | package net.skhu.controller;  import javax.servlet.http.HttpSession;  import org.springframework.stereotype.Controller;  import org.springframework.ui.Model;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.RequestMapping;  @Controller  public class Form5Controller {  @GetMapping("form5/register1")  public String register1(Model model) {  return "form5/register1";  }  @PostMapping("form5/register1")  public String register1(Model model, HttpSession session, String userid, String name,  String password1, String password2, String email, int departmentId)  {  String errorMsg = null;  if (userid == null || userid.length() == 0)  errorMsg = "사용자 아이디를 입력하세요";  else if (name == null || name.length() == 0)  errorMsg = "이름을 입력하세요";  else if (password1 == null || password1.length() == 0)  errorMsg = "비밀번호1을 입력하세요";  else if (password2 == null || password2.length() == 0)  errorMsg = "비밀번호2를 입력하세요";  else if (password1.equals(password2) == false)  errorMsg = "비밀번호 불일치";  else if (email == null || email.length() == 0)  errorMsg = "이메일 주소를 입력하세요";  else {  session.setAttribute("userid", userid);  session.setAttribute("name", name);  session.setAttribute("email", email);  return "redirect:register\_success1";  }  model.addAttribute("userid", userid);  model.addAttribute("name", name);  model.addAttribute("email", email);  model.addAttribute("departmentId", departmentId);  model.addAttribute("errorMsg", errorMsg);  return "form5/register1";  }  @RequestMapping("form5/register\_success1")  public String register\_success1(Model model) {  return "form5/register\_success1";  }  } |

## form5/register1.jsp

src/main/webapp/WEB-INF/views/form5/register1.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <style>  div.container { width: 400px; margin: 20px auto; }  form div { margin-bottom: 20px; }  label { display: block; }  input, select { width: 200px; padding: 5px; }  button { padding: 0.4em 1.5em; }  div.error { margin-top: 20px; color: red; }  </style>  </head>  <body>  <div class="container">  <form method="post">  <h1>회원가입</h1>  <div>  <label>사용자 아이디</label>  <input type="text" name="userid" value="${ userid }" />  </div>  <div>  <label>이름</label>  <input type="text" name="name" value="${ name }" />  </div>  <div>  <label>비밀번호1</label>  <input type="password" name="password1" />  </div>  <div>  <label>비밀번호2</label>  <input type="password" name="password2" />  </div>  <div>  <label>이메일</label>  <input type="email" name="email" value="${ email }" />  </div>  <div>  <label>전공</label>  <select name="departmentId">  <option value="1" ${ departmentId == 1 ? "selected" : "" }>소프트웨어공학</option>  <option value="2" ${ departmentId == 2 ? "selected" : "" }>컴퓨터공학</option>  <option value="3" ${ departmentId == 3 ? "selected" : "" }>정보통신공학</option>  <option value="4" ${ departmentId == 4 ? "selected" : "" }>글로컬IT</option>  </select>  </div>  <button type="submit" class="btn">회원가입</button>  </form>  <c:if test="${ not empty errorMsg }">  <div class="error">${ errorMsg }</div>  </c:if>  </div>  </body>  </html> |

<c:if test="${ **not empty** errorMsg }"

<div class="error">${ errorMsg }</div>

</c:if>

errorMsg 모델 애트리뷰트 값이 not empty 인 경우에만,

이 내용이 출력된다.

## form5/register\_success1.jsp

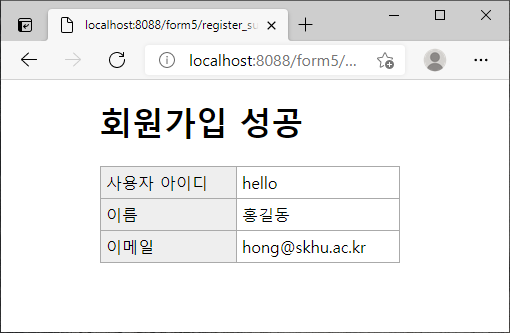
src/main/webapp/WEB-INF/views/form5/register\_success1.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <style>  div.container { width: 310px; margin: 20px auto; }  table { width: 300px; border-collapse: collapse; }  thead tr { background-color: #eee; }  td, th { border: 1px solid #aaa; padding: 5px; }  td:nth-child(1) { background-color: #eee; }  </style>  </head>  <body>  <div class="container">  <h1>회원가입 성공</h1>  <table>  <tr>  <td>사용자 아이디</td>  <td>${ userid }</td>  </tr>  <tr>  <td>이름</td>  <td>${ name }</td>  </tr>  <tr>  <td>이메일</td>  <td>${ email }</td>  </tr>  </table>  </div>  </body>  </html> |

## 실행

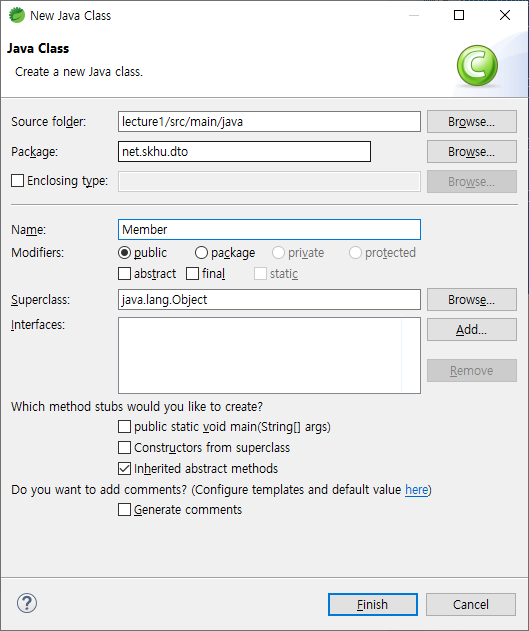
http://localhost:8088/form5/register1





# 회원가입 구현 #2

## Member.java 생성



### src/main/java/net/skhu/dto/Member.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59 | package net.skhu.dto;  public class Member {  String userid;  String name;  String password1;  String password2;  int departmentId;  String email;  public String getUserid() {  return userid;  }  public void setUserid(String userid) {  this.userid = userid;  }  public String getName() {  return name;  }  public void setName(String name) {  this.name = name;  }  public String getPassword1() {  return password1;  }  public void setPassword1(String password1) {  this.password1 = password1;  }  public String getPassword2() {  return password2;  }  public void setPassword2(String password2) {  this.password2 = password2;  }  public int getDepartmentId() {  return departmentId;  }  public void setDepartmentId(int departmentId) {  this.departmentId = departmentId;  }  public String getEmail() {  return email;  }  public void setEmail(String email) {  this.email = email;  }  } |

이 클래스는, 회원 가입 데이터를 채워서 전달하기 위한 클래스이다.

이렇게 데이터를 채워서 전달하는 용도의 클래스를 DTO(Data Transfer Object)이라고 부른다.

## Form5Controller.java

src/main/java/net/skhu/controller/Form5Controller.java 수정

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92 | package net.skhu.controller;  import javax.servlet.http.HttpSession;  import org.springframework.stereotype.Controller;  import org.springframework.ui.Model;  import org.springframework.web.bind.annotation.GetMapping;  import org.springframework.web.bind.annotation.PostMapping;  import org.springframework.web.bind.annotation.RequestMapping;  import net.skhu.dto.Member;  @Controller  public class Form5Controller {  @GetMapping("form5/register1")  public String register1(Model model) {  return "form5/register1";  }  @PostMapping("form5/register1")  public String register1(Model model, HttpSession session, String userid, String name,  String password1, String password2, String email, int departmentId)  {  String errorMsg = null;  if (userid == null || userid.length() == 0)  errorMsg = "사용자 아이디를 입력하세요";  else if (name == null || name.length() == 0)  errorMsg = "이름을 입력하세요";  else if (password1 == null || password1.length() == 0)  errorMsg = "비밀번호1을 입력하세요";  else if (password2 == null || password2.length() == 0)  errorMsg = "비밀번호2를 입력하세요";  else if (password1.equals(password2) == false)  errorMsg = "비밀번호 불일치";  else if (email == null || email.length() == 0)  errorMsg = "이메일 주소를 입력하세요";  else {  session.setAttribute("userid", userid);  session.setAttribute("name", name);  session.setAttribute("email", email);  return "redirect:register\_success1";  }  model.addAttribute("userid", userid);  model.addAttribute("name", name);  model.addAttribute("email", email);  model.addAttribute("departmentId", departmentId);  model.addAttribute("errorMsg", errorMsg);  return "form5/register1";  }  @RequestMapping("form5/register\_success1")  public String register\_success1(Model model) {  return "form5/register\_success1";  }  @GetMapping("form5/register2")  public String register2(Model model) {  return "form5/register1";  }  @PostMapping("form5/register2")  public String register2(Model model, HttpSession session, Member member)  {  String errorMsg = null;  if (member.getUserid() == null || member.getUserid().length() == 0)  errorMsg = "사용자 아이디를 입력하세요";  else if (member.getName() == null || member.getName().length() == 0)  errorMsg = "이름을 입력하세요";  else if (member.getPassword1() == null || member.getPassword1().length() == 0)  errorMsg = "비밀번호1을 입력하세요";  else if (member.getPassword2() == null || member.getPassword2().length() == 0)  errorMsg = "비밀번호2를 입력하세요";  else if (member.getPassword1().equals(member.getPassword2()) == false)  errorMsg = "비밀번호 불일치";  else if (member.getEmail() == null || member.getEmail().length() == 0)  errorMsg = "이메일 주소를 입력하세요";  else {  session.setAttribute("member", member);  return "redirect:register\_success2";  }  model.addAttribute("member", member);  model.addAttribute("errorMsg", errorMsg);  return "form5/register2";  }  @RequestMapping("form5/register\_success2")  public String register\_success2(Model model) {  return "form5/register\_success2";  }  } |

@PostMapping("form5/register2")

public String register2(Model model, HttpSession session, Member member)

액션 메소드의 파라미터가 객체인 경우에, request parameter 데이터가 이 객체에 자동으로 채워져 전달된다.

이때 이 객체의 set 메소드 이름과 request parameter 데이터의 이름이 일치하는 것만 채워진다.

|  |  |
| --- | --- |
| set 메소드 | request parameter 이름 |
| setUserid | userid |
| setName | name |
| setPassword1 | password1 |
| setPassword2 | password2 |
| setEmail | email |

model.addAttribute("member", member);

request parameter 데이터가 채워진 Member 객체를 model 에 채워서 뷰에 전달한다.

이렇게 전달되는 model attribute의 이름은 "member" 이고, 값은 Member 객체이다.

session.setAttribute("member", member);

request parameter 데이터가 채워진 Member 객체를 session 에 채워서 다음 액션 메소드에 전달한다.

## form5/register2.jsp

src/main/webapp/WEB-INF/views/form5/register2.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ taglib uri="http://java.sun.com/jsp/jstl/core" prefix="c" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <style>  div.container { width: 400px; margin: 20px auto; }  form div { margin-bottom: 20px; }  label { display: block; }  input, select { width: 200px; padding: 5px; }  button { padding: 0.4em 1.5em; }  div.error { margin-top: 20px; color: red; }  </style>  </head>  <body>  <div class="container">  <form method="post">  <h1>회원가입</h1>  <div>  <label>사용자 아이디</label>  <input type="text" class="text" name="userid" value="${ member.userid }" />  </div>  <div>  <label>이름</label>  <input type="text" class="text" name="name" value="${ member.name }" />  </div>  <div>  <label>비밀번호1</label>  <input type="password" class="text" name="password1" />  </div>  <div>  <label>비밀번호2</label>  <input type="password" class="text" name="password2" />  </div>  <div>  <label>이메일</label>  <input type="email" class="text" name="email" value="${ member.email }" />  </div>  <div>  <label>전공</label>  <select name="departmentId">  <option value="1" ${ member.departmentId == 1 ? "selected" : "" }>소프트웨어공학</option>  <option value="2" ${ member.departmentId == 2 ? "selected" : "" }>컴퓨터공학</option>  <option value="3" ${ member.departmentId == 3 ? "selected" : "" }>정보통신공학</option>  <option value="4" ${ member.departmentId == 4 ? "selected" : "" }>글로컬IT</option>  </select>  </div>  <button type="submit" class="btn">회원가입</button>  </form>  <c:if test="${ not empty errorMsg }">  <div class="error">${ errorMsg }</div>  </c:if>  </div>  </body>  </html> |

${ member.userid }

여기서 member 부분은, model attribute의 이름이다.

member 이름의 model attribute 값은 Member 객체이다.

이 객체의 getUserid() 메소드의 리턴값을 출력한다.

userid 멤버 변수 값이 아니고, getUserid() 메소드의 리턴값을 출력한다.

## register\_success2.jsp

src/main/webapp/WEB-INF/views/form5/register\_success2.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <style>  div.container { width: 310px; margin: 20px auto; }  table { width: 300px; border-collapse: collapse; }  thead tr { background-color: #eee; }  td, th { border: 1px solid #aaa; padding: 5px; }  td:nth-child(1) { background-color: #eee; }  </style>  </head>  <body>  <div class="container">  <h1>회원가입 성공</h1>  <table>  <tr>  <td>사용자 아이디</td>  <td>${ member.userid }</td>  </tr>  <tr>  <td>이름</td>  <td>${ member.name }</td>  </tr>  <tr>  <td>이메일</td>  <td>${ member.email }</td>  </tr>  <tr>  <td>전공ID</td>  <td>${ member.departmentId }</td>  </tr>  </table>  </div>  </body>  </html> |

# 연습문제

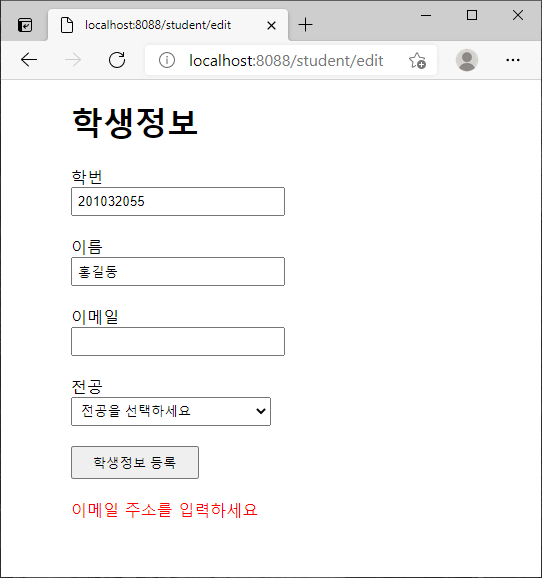
학생 정보 입력 페이지를 구현하라.

## Student.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41 | package net.skhu.dto;  public class Student {  String studentNo;  String name;  int departmentId;  String email;  public String getStudentNo() {  return studentNo;  }  public void setStudentNo(String studentNo) {  this.studentNo = studentNo;  }  public String getName() {  return name;  }  public void setName(String name) {  this.name = name;  }  public int getDepartmentId() {  return departmentId;  }  public void setDepartmentId(int departmentId) {  this.departmentId = departmentId;  }  public String getEmail() {  return email;  }  public void setEmail(String email) {  this.email = email;  }  } |

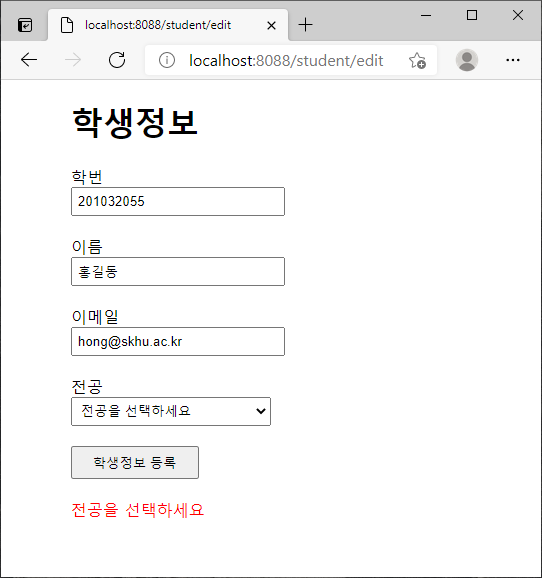
## student/edit

### http://localhost:8088/student/edit



이메일 주소를 입력하지 않고, 학생정보 등록 버튼을 클릭하면

위와 같이 에러가 출력된다.



전공을 선택하지 않고, 학생정보 등록 버튼을 클릭하면

위와 같이 에러가 출력된다.

### 힌트

<select name="departmentId">

<option value="0" ${ member.departmentId == 0 ? "selected" : "" }>전공을 선택하세요</option>

<option value="1" ${ member.departmentId == 1 ? "selected" : "" }>소프트웨어공학</option>

<option value="2" ${ member.departmentId == 2 ? "selected" : "" }>컴퓨터공학</option>

<option value="3" ${ member.departmentId == 3 ? "selected" : "" }>정보통신공학</option>

<option value="4" ${ member.departmentId == 4 ? "selected" : "" }>글로컬IT</option>

</select>

