spring form

2020-09-24

이승진

**학습목표**

select 태그 구현 방법

spring form 확장 태그

**목차**

[1. 개요 2](#_Toc52725816)

[2. 구현 방법 #1 3](#_Toc52725817)

[1) Student DTO 생성 3](#_Toc52725818)

[2) FormController 생성 4](#_Toc52725819)

[3) form/edit1.jsp 생성 5](#_Toc52725820)

[3. 구현 방법 #2 8](#_Toc52725821)

[1) FormController 수정 8](#_Toc52725822)

[2) form/edit2.jsp 생성 9](#_Toc52725823)

[4. 구현 방법 #3 12](#_Toc52725824)

[1) FormController 수정 12](#_Toc52725825)

[2) form/edit3.jsp 생성 14](#_Toc52725826)

# 개요

DB의 student 테이블의 departmentId 필드의 값은, 그 학생의 소속 학과의 id 이다.

학생 정보를 입력할 때, 학과 id를 입력하는 것은 바람직하지 않고,

학과 목록에서 학과를 선택해야 한다.

html 입력 form에서 학과 목록은 select 태그로 구현해야 한다.

학과 목록을 select 태그로 구현하는 방법을 알아보자.

# 구현 방법 #1

## Student DTO 생성

### src/main/java/net/skhu/dto/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  42  43  44  45  46  47  48  49 | package net.skhu.dto;  public class Student {  int id;  String studentNumber;  String name;  int departmentId;  int year;  public int getId() {  return id;  }  public void setId(int id) {  this.id = id;  }  public String getStudentNumber() {  return studentNumber;  }  public void setStudentNumber(String studentNumber) {  this.studentNumber = studentNumber;  }  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 int getYear() {  return year;  }  public void setYear(int year) {  this.year = year;  }  } |

## FormController 생성

지난 시간에 만든 mybatis1 프로젝트에 FormController 클래스를 생성하자.

### src/main/java/net/skhu/controller/FormController.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 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.Student;  @Controller  @RequestMapping("form")  public class FormController {  @GetMapping("edit1")  public String edit1(Model model) {  model.addAttribute("student", new Student());  return "form/edit1";  }  @PostMapping("edit1")  public String edit1(Model model, Student student) {  // TODO: DB 저장 기능을 구현해야 함.  model.addAttribute("message", "저장했습니다.");  return "form/edit1";  }  } |

## form/edit1.jsp 생성

### src/main/webapp/WEB-INF/views/form/edit1.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  61  62  63  64  65 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <link rel="stylesheet" type="text/css" href="/common.css" />  <style>  form { width: 600px; margin: auto;  box-shadow: 0 0 4px lightgray, 2px 2px 4px gray; overflow: auto; }  div.title { font-size: 20pt; padding: 10px; background-color: #eee; }  table { margin: 20px; }  td { min-width: 100; padding: 5px; }  td:nth-child(1) { text-align: right; }  button { margin: 5px 0 20px 20px; }  div.message { border-radius: 10px; padding: 12px; margin: 10px;  background-color: #ffa; border: 1px solid #cc0; }  </style>  </head>  <body>  <div class="container">  <form method="post">  <div class="title">학생 정보</div>  <table>  <tr>  <td>id</td>  <td><input type="text" value="${ student.id }" readonly disabled /></td>  </tr>  <tr>  <td>학번</td>  <td><input type="text" name="studentNumber" value="${ student.studentNumber }" /></td>  </tr>  <tr>  <td>이름</td>  <td><input type="text" name="name" value="${ student.name }" /></td>  </tr>  <tr>  <td>학과</td>  <td>  <select name="departmentId">  <option value="1" ${ student.departmentId == 1 ? "selected" : "" }>국어국문학</option>  <option value="2" ${ student.departmentId == 2 ? "selected" : "" }>영어영문학</option>  <option value="3" ${ student.departmentId == 3 ? "selected" : "" }>불어불문학</option>  <option value="4" ${ student.departmentId == 4 ? "selected" : "" }>소프트웨어공학과</option>  <option value="5" ${ student.departmentId == 5 ? "selected" : "" }>컴퓨터공학과</option>  <option value="6" ${ student.departmentId == 6 ? "selected" : "" }>정보통신공학과</option>  <option value="7" ${ student.departmentId == 7 ? "selected" : "" }>글로컬IT공학과</option>  </select>  </td>  <tr>  <td>학년</td>  <td><input type="text" name="year" value="${ student.year }" /></td>  </tr>  </table>  <button type="submit" class="btn">저장</button>    <c:if test="${ not empty message }">  <div class="message">  ${ message }  </div>  </c:if>  </form>  </div>  </body>  </html> |

### selected 출력

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

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

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

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

<option value="5" ${ student.departmentId == 5 ? "selected" : "" }>컴퓨터공학과</option>

<option value="6" ${ student.departmentId == 6 ? "selected" : "" }>정보통신공학과</option>

<option value="7" ${ student.departmentId == 7 ? "selected" : "" }>글로컬IT공학과</option>

만약 "student" 이름의 model attribute 객체의 ("student" 이름으로 model에 등록된 객체의)

departmentId 속성의 값이 4 이라면, (getDepartmentId() 메소드의 리턴 값이 4 이라면)

위 소스코드의 실행 결과 출력은 다음과 같다.

<option value="1" >국어국문학</option>

<option value="2" >영어영문학</option>

<option value="3" >불어불문학</option>

<option value="4" selected>소프트웨어공학과</option>

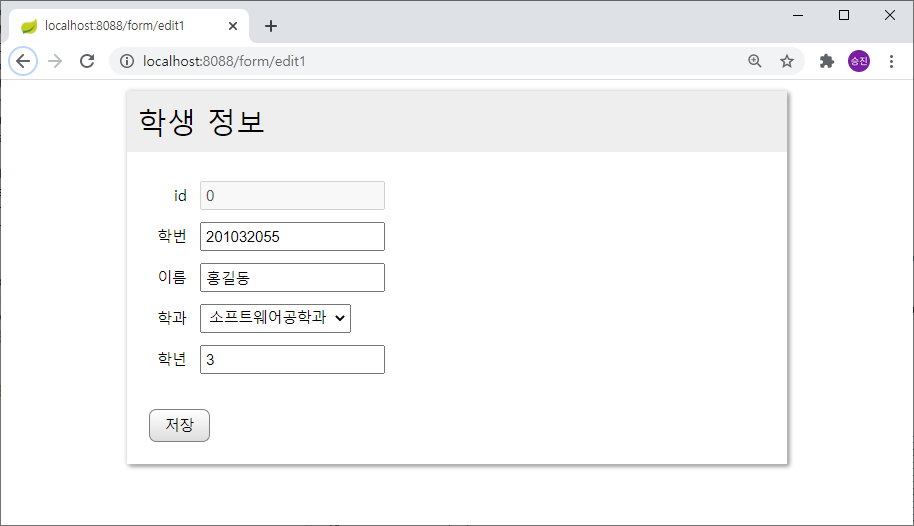
<option value="5" >컴퓨터공학과</option>

<option value="6" >정보통신공학과</option>

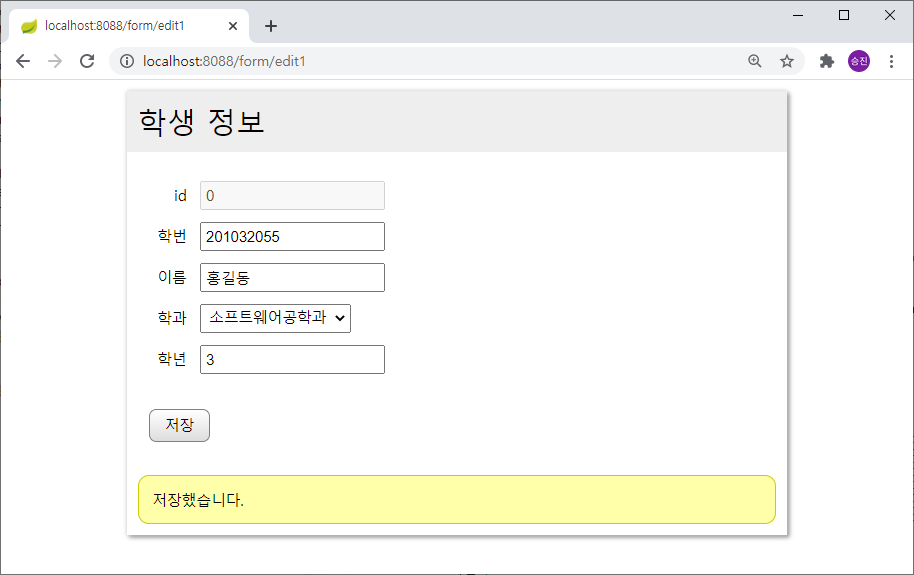
<option value="7" >글로컬IT공학과</option>

### 실행

http://localhost:8088/form/edit1



위와 같이 입력하고 저장 버튼을 클릭하자.



저장 버튼을 누른 후 화면이다.

이름 입력 칸에 홍길동 값이 채워져 있는 이유는 무엇인가?

학과 목록에서 소프트웨어공학과가 선택되어 있는 이유는 무엇인가?

### 개선할 점

학과 목록을 hard coding 하는 것은 바람직하지 않다.

학과 목록을 DB에서 조회해야 한다.

# 구현 방법 #2

## FormController 수정

### src/main/java/net/skhu/controller/FormController.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 | package net.skhu.controller;  import org.springframework.beans.factory.annotation.Autowired;  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.Student;  import net.skhu.mapper.DepartmentMapper;  @Controller  @RequestMapping("/form")  public class FormController {  @Autowired DepartmentMapper departmentMapper;  @GetMapping("edit1")  public String edit1(Model model) {  model.addAttribute("student", new Student());  return "form/edit1";  }  @PostMapping("edit1")  public String edit1(Model model, Student student) {  // TODO: DB 저장 기능을 구현해야 함.  model.addAttribute("message", "저장했습니다.");  return "form/edit1";  }  @GetMapping("edit2")  public String edit2(Model model) {  model.addAttribute("student", new Student());  model.addAttribute("departments", departmentMapper.findAll());  return "form/edit2";  }  @PostMapping("edit2")  public String edit2(Model model, Student student) {  // TODO: DB 저장 기능을 구현해야 함.  model.addAttribute("message", "저장했습니다.");  model.addAttribute("departments", departmentMapper.findAll());  return "form/edit2";  }  } |

## form/edit2.jsp 생성

### src/main/webapp/WEB-INF/views/form/edit2.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  61  62  63  64  65 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <link rel="stylesheet" type="text/css" href="/common.css" />  <style>  form { width: 600px; margin: auto;  box-shadow: 0 0 4px lightgray, 2px 2px 4px gray; overflow: auto; }  div.title { font-size: 20pt; padding: 10px; background-color: #eee; }  table { margin: 20px; }  td { min-width: 100; padding: 5px; }  td:nth-child(1) { text-align: right; }  button { margin: 5px 0 20px 20px; }  div.message { border-radius: 10px; padding: 12px; margin: 10px;  background-color: #ffa; border: 1px solid #cc0; }  </style>  </head>  <body>  <div class="container">  <form method="post">  <div class="title">학생 정보</div>  <table>  <tr>  <td>id</td>  <td><input type="text" value="${ student.id }" readonly disabled /></td>  </tr>  <tr>  <td>학번</td>  <td><input type="text" name="studentNumber" value="${ student.studentNumber }" /></td>  </tr>  <tr>  <td>이름</td>  <td><input type="text" name="name" value="${ student.name }" /></td>  </tr>  <tr>  <td>학과</td>  <td>  <select name="departmentId">  <c:forEach var="d" items="${ departments }">  <option value="${ d.id }" ${ student.departmentId == d.id ? "selected" : "" }>  ${ d.departmentName }  </option>  </c:forEach>  </select>  </td>  <tr>  <td>학년</td>  <td><input type="text" name="year" value="${ student.year }" /></td>  </tr>  </table>  <button type="submit" class="btn">저장</button>    <c:if test="${ not empty message }">  <div class="message">  ${ message }  </div>  </c:if>  </form>  </div>  </body>  </html> |

<c:forEach var="d" items="${ departments }">

<option value="${ d.id }" ${ student.departmentId == d.id ? "selected" : "" }>

${ d.departmentName }

</option>

</c:forEach>

만약 "student" 이름의 model attribute 객체의 ("student" 이름으로 model에 등록된 객체의)

departmentId 속성의 값이 4 이라면, (getDepartmentId() 메소드의 리턴 값이 4 이라면)

위 소스코드의 실행 결과 출력은 다음과 같다.

<option value="1" >국어국문학</option>

<option value="2" >영어영문학</option>

<option value="3" >불어불문학</option>

<option value="4" selected>소프트웨어공학과</option>

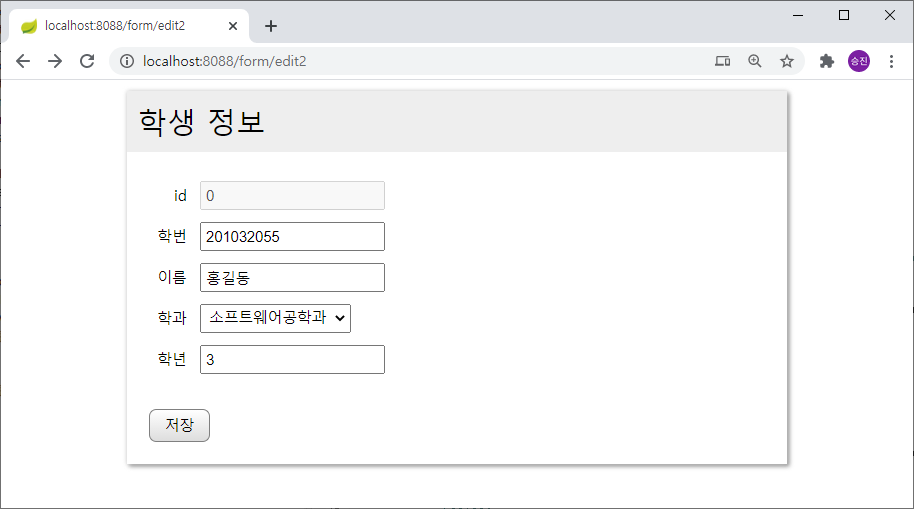
<option value="5" >컴퓨터공학과</option>

<option value="6" >정보통신공학과</option>

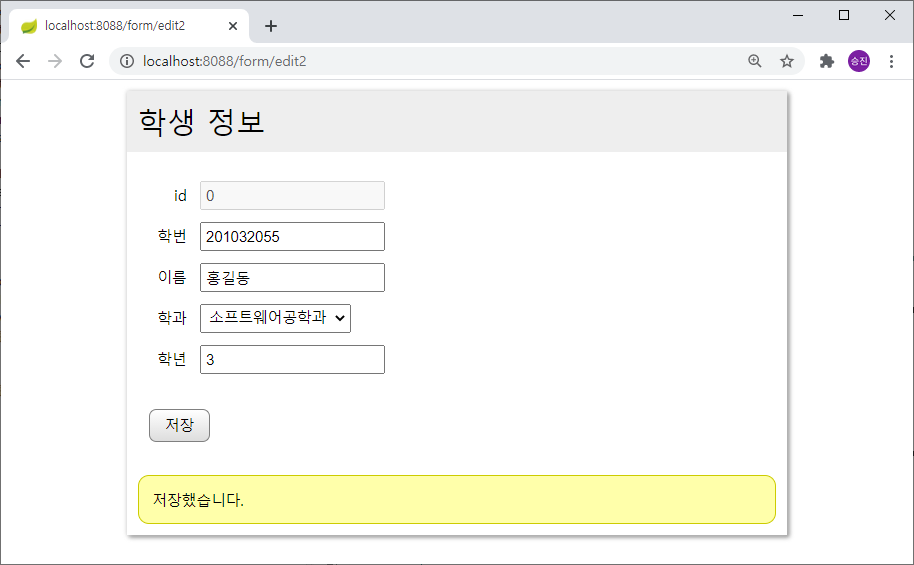
<option value="7" >글로컬IT공학과</option>

### 실행

http://localhost:8088/form/edit2



위와 같이 입력하고 저장 버튼을 클릭하자.



# 구현 방법 #3

## FormController 수정

### src/main/java/net/skhu/controller/FormController.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 | package net.skhu.controller;  import org.springframework.beans.factory.annotation.Autowired;  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.Student;  import net.skhu.mapper.DepartmentMapper;  @Controller  @RequestMapping("/form")  public class FormController {  @Autowired DepartmentMapper departmentMapper;  @GetMapping("edit1")  public String edit1(Model model) {  model.addAttribute("student", new Student());  return "form/edit1";  }  @PostMapping("edit1")  public String edit1(Model model, Student student) {  // TODO: DB 저장 기능을 구현해야 함.  model.addAttribute("message", "저장했습니다.");  return "form/edit1";  }  @GetMapping("edit2")  public String edit2(Model model) {  model.addAttribute("student", new Student());  model.addAttribute("departments", departmentMapper.findAll());  return "form/edit2";  }  @PostMapping("edit2")  public String edit2(Model model, Student student) {  // TODO: DB 저장 기능을 구현해야 함.  model.addAttribute("message", "저장했습니다.");  model.addAttribute("departments", departmentMapper.findAll());  return "form/edit2";  }  @GetMapping("edit3")  public String edit3(Model model) {  model.addAttribute("student", new Student());  model.addAttribute("departments", departmentMapper.findAll());  return "form/edit3";  }  @PostMapping("edit3")  public String edit3(Model model, Student student) {  // TODO: DB 저장 기능을 구현해야 함.  model.addAttribute("message", "저장했습니다.");  model.addAttribute("departments", departmentMapper.findAll());  return "form/edit3";  }  } |

edit2 메소드 구현과 edit3 메소드 구현에는 차이가 없다.

## form/edit3.jsp 생성

### src/main/webapp/WEB-INF/views/form/edit3.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 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>  <%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">  <link rel="stylesheet" type="text/css" href="/common.css" />  <style>  form { width: 600px; margin: auto;  box-shadow: 0 0 4px lightgray, 2px 2px 4px gray; overflow: auto; }  div.title { font-size: 20pt; padding: 10px; background-color: #eee; }  table { margin: 20px; }  td { min-width: 100; padding: 5px; }  td:nth-child(1) { text-align: right; }  button { margin: 5px 0 20px 20px; }  div.message { border-radius: 10px; padding: 12px; margin: 10px;  background-color: #ffa; border: 1px solid #cc0; }  </style>  </head>  <body>  <div class="container">  <form:form method="post" modelAttribute="student">  <div class="title">학생 정보</div>  <table>  <tr>  <td>id</td>  <td><form:input path="id" readonly="true" disabled="true" /></td>  </tr>  <tr>  <td>학번</td>  <td><form:input path="studentNumber" /></td>  </tr>  <tr>  <td>이름</td>  <td><form:input path="name" /></td>  </tr>  <tr>  <td>학과</td>  <td>  <form:select path="departmentId"  items="${ departments }" itemLabel="departmentName" itemValue="id" />  </td>  <tr>  <td>학년</td>  <td><form:input path="year" /></td>  </tr>  </table>  <button type="submit" class="btn">저장</button>    <c:if test="${ not empty message }">  <div class="message">  ${ message }  </div>  </c:if>  </form:form>  </div>  </body>  </html> |

<%@ taglib uri="http://www.springframework.org/tags/form" prefix="form" %>

spring form 확장 태그를 사용하기 위한 선언이다.

spring form 확장 태그를 사용하면, 입력 폼을 조금 더 간결하게 구현할 수 있다.

특히 select 태그 구현이 간결해진다.

<form:form method="post" modelAttribute="student">

spring form 확장 태그이다.

이 입력 폼에서 수정할 객체는 "student" 이름의 model attribute 이다.

모델에 이 이름의 객체가 있어야 한다. 없으면 에러가 발생한다.

<form:input path="id" readonly="true" disabled="true" />

이 코드는 아래 코드와 동일하다.

<input type="text" value="${ student.id }" readonly disabled />

student는 <form:form> 태그에 선언된 model attribuet 이다.

<form:input path="studentNumber" />

이 코드는 아래 코드와 동일하다.

<input type="text" name="studentNumber" value="${ student.studentNumber }" />

student는 <form:form> 태그에 선언된 model attribuet 이다.

<form:input path="name" />

이 코드는 아래 코드와 동일하다.

<input type="text" name="name" value="${ student.name }" />

<form:select path="departmentId"

items="${ departments }" itemLabel="departmentName" itemValue="id" />

이 코드는 아래 코드와 동일하다.

<select name="departmentId">

<c:forEach var="d" items="${ departments }">

<option value="${ d.id }" ${ student.departmentId == d.id ? "selected" : "" }>

${ d.departmentName }

</option>

</c:forEach>

</select>

<form:input path="year" />

이 코드는 아래 코드와 동일하다.

<input type="text" name="year" value="${ student.year }" />