학과 등록 & 학생 등록

2019-12-03

이승진

목차

[1. 학과 목록 조회 2](#_Toc26286780)

[1) DepartmentDAO.java 2](#_Toc26286781)

[2) departmentList.jsp 3](#_Toc26286782)

[3) 실행 4](#_Toc26286783)

[2. 학과 등록 기능 구현 5](#_Toc26286784)

[1) Auto Increment 필드 5](#_Toc26286785)

[2) DepartmentDAO1.java 6](#_Toc26286786)

[3) Http Request의 유형 - GET / POST 7](#_Toc26286787)

[4) 리다이렉트(redirect) 8](#_Toc26286788)

[5) createDepartment1.jsp 9](#_Toc26286789)

[6) 실행 과정 설명 11](#_Toc26286790)

[3. 학생 등록 기능 구현 13](#_Toc26286791)

[1) StudentDAO4.java 13](#_Toc26286792)

[2) createStudent1.jsp 15](#_Toc26286793)

[3) studentList5.jsp 17](#_Toc26286794)

[4. 실습 18](#_Toc26286795)

# 학과 목록 조회

## DepartmentDAO.java

src/main/java/lecture1/jdbc1/DepartmentDAO.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 | package lecture1.jdbc1;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  import lecture1.DB;  public class DepartmentDAO {  public static List<Department> findAll() throws Exception {  String sql = "SELECT \* FROM department";  try (Connection connection = DB.getConnection("student1");  PreparedStatement statement = connection.prepareStatement(sql);  ResultSet resultSet = statement.executeQuery()) {  ArrayList<Department> list = new ArrayList<Department>();  while (resultSet.next()) {  Department department = new Department();  department.setId(resultSet.getInt("id"));  department.setDepartmentName(resultSet.getString("departmentName"));  list.add(department);  }  return list;  }  }  } |

## departmentList.jsp

WebContext/jdbc1/departmentList1.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"%>  <%@ page import="java.util.List, lecture1.jdbc1.\*" %>  <%  List<Department> list = DepartmentDAO.findAll();  %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>  <style>  body { font-family: 굴림체; }  thead th { background-color: #eee; }  table.table { width: 400px; }  </style>  </head>  <body>  <div class="container">  <h1>학과목록</h1>  <table class="table table-bordered table-condensed">  <thead>  <tr>  <th>ID</th>  <th>학과명</th>  </tr>  </thead>  <tbody>  <% for (Department department : list) { %>  <tr>  <td><%= department.getId() %></td>  <td><%= department.getDepartmentName() %></td>  </tr>  <% } %>  </tbody>  </table>  </div>  </body>  </html> |

## 실행

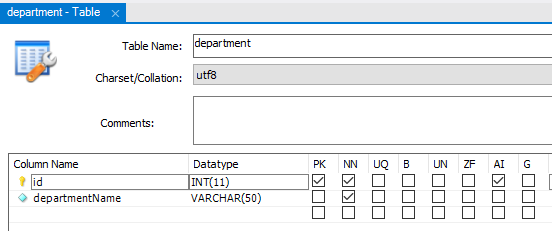


# 학과 등록 기능 구현

## Auto Increment 필드

department 테이블에서는 두 개의 필드가 있다.

|  |  |
| --- | --- |
| 필드명 | 타입 |
| id | int |
| departmentName | VARCHAR(50) |



이 테이블의 기본키(primary key)인 id 필드는, AI(Auto Increment) 유형의 필드이다.

AI 유형의 필드에는, 1 부터 시작하는 일련번호 값이 자동으로 부여된다.

따라서, 레코드를 INSERT/UPDATE 할 때, AI 유형의 필드값을 지정하면 안된다.

즉

|  |
| --- |
| INSERT department (id, departmentName)  VALUES (20, 'IT융합학부'); |

위의 SQL 문을 실행하면 오류가 발생한다.

|  |
| --- |
| INSERT department (departmentName)  VALUES ('IT융합학부'); |

위 SQL 문은 잘 실행된다.

## DepartmentDAO1.java

src/main/java/lecture1/jdbc1/DepartmentDAO1.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 | package lecture1.jdbc1;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.util.ArrayList;  import java.util.List;  import lecture1.DB;  public class DepartmentDAO1 {  public static List<Department> findAll() throws Exception {  String sql = "SELECT \* FROM department";  try (Connection connection = DB.getConnection("student1");  PreparedStatement statement = connection.prepareStatement(sql);  ResultSet resultSet = statement.executeQuery()) {  ArrayList<Department> list = new ArrayList<Department>();  while (resultSet.next()) {  Department department = new Department();  department.setId(resultSet.getInt("id"));  department.setDepartmentName(resultSet.getString("departmentName"));  list.add(department);  }  return list;  }  }  public static void insert(Department department) throws Exception {  String sql = "INSERT department (departmentName) VALUES (?)";  try (Connection connection = DB.getConnection("student1");  PreparedStatement statement = connection.prepareStatement(sql)) {  statement.setString(1, department.getDepartmentName());  statement.executeUpdate();  }  }  } |

INSERT department (departmentName) VALUES (?)

department 테이블에 새 레코드를 삽입(insert)하는 SQL 명령이다.

ResultSet resultSet = statement.executeQuery()) {

조회 SQL 명령(SELECT 명령)은 위와 같이 실행해야 한다.

statement.executeUpdate();

조회가 아닌 SQL 명령(INSERT/UPDATE/DELETE 명령)은 위와 같이 실행해야 한다.

## Http Request의 유형 - GET / POST

웹브라우저가 웹서버에 요청하는 HTTP Request의 유형에는

GET 방식 요청과 POST 방식 요청이 있다. (이 둘 외에도 더 있다)

### POST 방식 요청

<form method="post">

<button type="submit" class="btn btn-primary">저장</button>

</form>

method="post" 인 form 태그 내부의 type="submit" 버튼이 클릭될 때,

발생하는 HTTP Request 는 POST 방식이다.

### GET 방식 요청

method="post" 처럼, 요청 방식을 지정하지 않은 요청들은

전부 GET 방식이다.

<form>

<input type="text" name="srchText" />

<button type="submit" class="btn btn-primary">조회</button>

</form>

위 입력 폼에서 type="submit" 버튼이 클릭될 때,

발생하는 HTTP Request 는 GET 방식이다.

### GET 방식 요청과 query string

GET 방식 요청일 때, 서버에 전달되는 request parameter는

URL의 query string 이어야 한다.

### GET 방식 요청과 입력 폼

GET 방식 요청일 경우에,

입력 폼에 입력된 값은, URL의 query string 의 형태로 서버에 전달된다.

### POST 방식 요청과 입력 폼

GET 방식 요청일 경우에,

입력 폼에 입력된 값은, URL의 query string 의 형태가 아니고,

HTTP Request의 본문에 담겨서 서버에 전달된다.

## 리다이렉트(redirect)

response.sendRedirect(**URL**);

서버에서 위 Java 코드가 실행되면, 서버와 웹브라우저에서 다음과 같은 과정이 진행된다.

(1) 서버는 웹브라우저에게 다음과 같은 명령(메타 태그)를 전송한다.

'즉시 **URL** 을 다시 서버에 요청하라'

(2) 그 명령(메타 태그)를 받은 웹브라우저는,

명령에 포함된 **URL** 을 서버에 즉시 요청한다.

(3) 요청된 **URL** 이 서버에서 실행되고,

실행 결과가 서버로부터 웹브라우저로 출력(전송)된다.

(4) 웹브라우저 주소칸에는, 방금 서버에 요청한 **URL** 이 보이고,

웹브라우저 창에는, 상금 서버가 전송한 실행 결과 출력이 보인다.

긴 설명을 요약하자면,

서버에서 위 Java 코드가 실행되면, URL로 리다이렉트 하라는 명령이

서버로부터 웹브라우저로 전송된다.

## createDepartment1.jsp

WebContext/jdbc1/createDepartment1.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 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ page import="java.util.List, lecture1.jdbc1.\*" %>  <%  if (request.getMethod().equals("POST")) {  request.setCharacterEncoding("UTF-8");  Department department = new Department();  String s = request.getParameter("departmentName");  department.setDepartmentName(s);  DepartmentDAO1.insert(department);  response.sendRedirect("departmentList2.jsp");  return;  }  %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>  <style>  body { font-family: 굴림체; }  table.table { width: 500px; }  table td:nth-child(1) { background-color: #eee; }  </style>  </head>  <body>  <div class="container">  <h1>학과 등록</h1>  <form method="post">  <table class="table table-bordered table-condensed">  <tr>  <td>학과명</td>  <td>  <input type="text" name="departmentName" class="form-control" />  </td>  </tr>  </table>    <button type="submit" class="btn btn-primary">저장</button>  <a href="javascript:window.history.back()" class="btn btn-info">돌아가기</a>  </form>  </div>  </body>  </html> |

### if (request.getMethod().equals("POST")) {

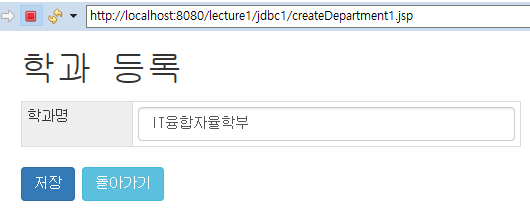
POST 방식 요청일 경우에만 true가 된다.

웹브라우저에서 아래 URL을 서버에 요청할 때는, GET 방식 요청이다.

http://localhost:8080/lecture1/jdbc1/createDepartment1.jsp

따라서, 이클립스에서 createDepartment1.jsp를 처음 실행할 때는,

위 if 문은 false가 된다.



위 화면에서 '저장' 버튼을 클릭하면,

현재 URL(http://localhost:8080/lecture1/jdbc1/createDepartment1.jsp)을

POST 방식으로 요청하게 된다.

이때는 우의 if 문이 true가 된다.

### request.setCharacterEncoding("UTF-8");

request parameter 데이터에 한글이 포함되어 있는 경우에,

request parameter 값을 꺼내기 전에 먼저 위 코드를 실행해야 한다.

### 학과 등록(INSERT)

Department department = new Department();

String s = request.getParameter("departmentName");

department.setDepartmentName(s);

DepartmentDAO1.insert(department);

### response.sendRedirect("departmentList2.jsp");

현재 서버에 요청된 UR은 (http://localhost:8080/lecture1/jdbc1/createDepartment1.jsp) 이다.

이 URL에서 파일명 부분을 departmentList2.jsp 으로 변경하면,

(http://localhost:8080/lecture1/jdbc1/departmentList2.jsp) 이다.

위 Java 코드를 실행하면,

즉시 (http://localhost:8080/lecture1/jdbc1/departmentList2.jsp) URL을 서버에 다시 요청하라는 명령이

웹브라우저로 전송된다.

### return;

JSP 파일 중간에서 return 문이 실행되면,

이 return 문 아래 내용들을 실행되지 않고, 출력되지도 않는다.

## 실행 과정 설명

### (1)

(http://localhost:8080/lecture1/jdbc1/createDepartment1.jsp) URL이 처음 서버에 요청되어서

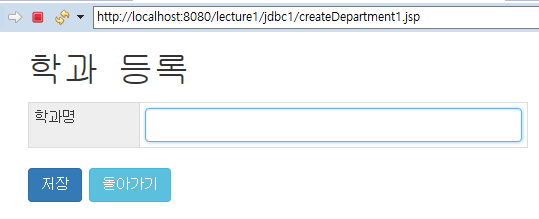
createDepartment1.jsp 파일이 서버에서 실행될 때는 GET 방식 요청이다.

### (2)

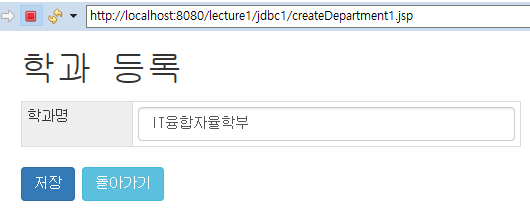
서버에서 reateDepartment1.jsp 실행

GET 방식 요청이므로 if 문은 false가 되고, if 문의 본문은 실행되지 않는다.

학과 등록 입력폼이 웹브라우저에 출력된다.



### (3)



웹브라우저에서 입력폼에 위와 같이 입력하고 저장 버튼을 클릭하면,

(http://localhost:8080/lecture1/jdbc1/createDepartment1.jsp) URL이 서버에 다시 요청된다.

이것은 POST 방식 요청이다.

### (4)

서버에서 reateDepartment1.jsp 실행

POST 방식 요청이므로 if 문은 true가 되고, if 문의 본문이 실행된다.

새 학과 레코드가 department 테이블에 INSERT 된다.

(http://localhost:8080/lecture1/jdbc1/departmentList2.jsp) URL로 즉시 리다이렉트 하라는 명령이

웹브라우저로 전송된다.

학과 등록 입력폼이 웹브라우저에 출력되지 않는다. (return 하기 때문에)

### (5)

웹브라우저는

(http://localhost:8080/lecture1/jdbc1/departmentList2.jsp) URL을 서버에 즉시 요청한다.



# 학생 등록 기능 구현

## StudentDAO4.java

src/main/java/lecture1/jdbc1/StudentDAO4.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 | package lecture1.jdbc1;  import java.sql.Connection;  import java.sql.PreparedStatement;  import java.sql.ResultSet;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.List;  import lecture1.DB;  public class StudentDAO4 {  private static Student getStudentFrom(ResultSet resultSet) throws SQLException {  Student student = new Student();  student.setId(resultSet.getInt("id"));  student.setStudentNumber(resultSet.getString("studentNumber"));  student.setName(resultSet.getString("name"));  student.setDepartmentId(resultSet.getInt("departmentId"));  student.setYear(resultSet.getInt("year"));  student.setDepartmentName(resultSet.getString("departmentName"));  return student;  }  public static List<Student> findAll() throws Exception {  String sql = "SELECT s.\*, d.departmentName " +  "FROM student s LEFT JOIN department d ON s.departmentId = d.id";  try (Connection connection = DB.getConnection("student1");  PreparedStatement statement = connection.prepareStatement(sql);  ResultSet resultSet = statement.executeQuery()) {  ArrayList<Student> list = new ArrayList<Student>();  while (resultSet.next())  list.add(getStudentFrom(resultSet));  return list;  }  }  public static Student findById(int id) throws Exception {  String sql = "SELECT s.\*, d.departmentName " +  "FROM student s LEFT JOIN department d ON s.departmentId = d.id " +  "WHERE s.id = ?";  try (Connection connection = DB.getConnection("student1");  PreparedStatement statement = connection.prepareStatement(sql)) {  statement.setInt(1, id);  try (ResultSet resultSet = statement.executeQuery()) {  if (resultSet.next())  return getStudentFrom(resultSet);  else  return null;  }  }  }  public static void insert(Student student) throws Exception {  String sql = "INSERT student (name, studentNumber, departmentId, year) VALUES (?, ?, ?, ?)";  try (Connection connection = DB.getConnection("student1");  PreparedStatement statement = connection.prepareStatement(sql)) {  statement.setString(1, student.getName());  statement.setString(2, student.getStudentNumber());  statement.setInt(3, student.getDepartmentId());  statement.setInt(4, student.getYear());  statement.executeUpdate();  }  }  } |

## createStudent1.jsp

WebContext/jdbc1/createStudent1.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  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 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ page import="java.util.List, lecture1.jdbc1.\*" %>  <%  if (request.getMethod().equals("POST")) {  Student student = new Student();  request.setCharacterEncoding("UTF-8");    String name = request.getParameter("name");  student.setName(name);    String studentNumber = request.getParameter("studentNumber");  student.setStudentNumber(studentNumber);  System.out.println(student.getStudentNumber());  String s = request.getParameter("departmentId");  int departmentId = Integer.parseInt(s);  student.setDepartmentId(departmentId);  s = request.getParameter("year");  int year = Integer.parseInt(s);  student.setYear(year);    StudentDAO4.insert(student);  response.sendRedirect("studentList5.jsp");  return;  }  List<Department> departments = DepartmentDAO.findAll();  %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>  <style>  body { font-family: 굴림체; }  table.table { width: 500px; }  table td:nth-child(1) { background-color: #eee; }  </style>  </head>  <body>  <div class="container">  <h1>학생 등록</h1>  <form method="post">  <table class="table table-bordered table-condensed">  <tr>  <td>이름</td>  <td>  <input type="text" name="name" class="form-control" />  </td>  </tr>  <tr>  <td>학번</td>  <td>  <input type="text" name="studentNumber" class="form-control" />  </td>  </tr>  <tr>  <td>학과</td>  <td>  <select name="departmentId" class="form-control">  <% for (Department department : departments) { %>  <% int id = department.getId(); %>  <% String name = department.getDepartmentName(); %>  <option value="<%= id %>">  <%= name %>  </option>  <% } %>  </select>  </td>  </tr>  <tr>  <td>학년</td>  <td>  <input type="text" name="year" class="form-control" />  </td>  </tr>  </table>    <button type="submit" class="btn btn-primary">저장</button>  <a href="javascript:window.history.back()" class="btn btn-info">돌아가기</a>  </form>  </div>  </body>  </html> |

## studentList5.jsp

WebContext/jdbc1/3) studentList5.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 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ page import="java.util.List, lecture1.jdbc1.\*" %>  <%  List<Student> list = StudentDAO.findAll();  %>  <!DOCTYPE html>  <html>  <head>  <meta http-equiv="Content-Type" content="text/html; charset=utf-8">  <meta name="viewport" content="width=device-width, initial-scale=1">  <link rel="stylesheet" href="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/css/bootstrap.min.css">  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.1.1/jquery.min.js"></script>  <script src="https://maxcdn.bootstrapcdn.com/bootstrap/3.3.7/js/bootstrap.min.js"></script>  <style>  body { font-family: 굴림체; }  thead th { background-color: #eee; }  table.table { width: 700px; }  </style>  </head>  <body>  <div class="container">  <h1>학생목록</h1>  <table class="table table-bordered table-condensed">  <thead>  <tr>  <th>학번</th>  <th>이름</th>  <th>학과</th>  <th>학년</th>  </tr>  </thead>  <tbody>  <% for (Student student : list) { %>  <tr>  <td><%= student.getStudentNumber() %></td>  <td><%= student.getName() %></td>  <td><%= student.getDepartmentName() %></td>  <td><%= student.getYear() %></td>  </tr>  <% } %>  </tbody>  </table>  <a class="btn btn-primary" href="createStudent1.jsp">학생등록</a>  </div>  </body>  </html> |

# 실습

교수(professor) 등록 기능을 구현하라.