학생 수정 삭제 기능

2019-12-11

이승진

목차

[1. DTO 클래스 구현 2](#_Toc26956240)

[1) Department.java 2](#_Toc26956241)

[2) Student.java 3](#_Toc26956242)

[2. DAO 클래스 구현 4](#_Toc26956243)

[1) DepartmentDAO.java 4](#_Toc26956244)

[2) StudentDAO.java 5](#_Toc26956245)

[3. pagination tag 파일 7](#_Toc26956246)

[1) pagination.tag 7](#_Toc26956247)

[4. JSP 구현 9](#_Toc26956248)

[1) studentList5.jsp 9](#_Toc26956249)

[2) studentCreate1.jsp 11](#_Toc26956250)

[3) studentEdit1.jsp 13](#_Toc26956251)

[4) studentDelete1.jsp 15](#_Toc26956252)

# DTO 클래스 구현

## Department.java

src/lecture1/jdbc2/Department.java

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22 | package lecture1.jdbc2;  public class Department {  int id;  String departmentName;  public int getId() {  return id;  }  public void setId(int id) {  this.id = id;  }  public String getDepartmentName() {  return departmentName;  }  public void setDepartmentName(String departmentName) {  this.departmentName = departmentName;  }  } |

## Student.java

src/lecture1/jdbc2/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  50  51  52  53  54  55  56  57  58  59 | package lecture1.jdbc2;  public class Student {  int id;  String studentNumber;  String name;  int departmentId;  int year;  String departmentName;  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;  }  public String getDepartmentName() {  return departmentName;  }  public void setDepartmentName(String departmentName) {  this.departmentName = departmentName;  }  } |

# DAO 클래스 구현

## DepartmentDAO.java

src/lecture1/jdbc2/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  29 | package lecture1.jdbc2;  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;  }  }  } |

## StudentDAO.java

src/lecture1/jdbc2/StudentDAO.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  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104 | package lecture1.jdbc2;  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 StudentDAO3 {  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(int currentPage, int pageSize) throws Exception {  String sql = "SELECT s.\*, d.departmentName" +  " FROM student s LEFT JOIN department d ON s.departmentId = d.id" +  " LIMIT ?, ?";  try (Connection connection = DB.getConnection("student1");  PreparedStatement statement = connection.prepareStatement(sql)) {  statement.setInt(1, (currentPage - 1) \* pageSize);  statement.setInt(2, pageSize);  try (ResultSet resultSet = statement.executeQuery()) {  ArrayList<Student> list = new ArrayList<Student>();  while (resultSet.next())  list.add(getStudentFrom(resultSet));  return list;  }  }  }  public static int count() throws Exception {  String sql = "SELECT COUNT(\*) FROM student";  try (Connection connection = DB.getConnection("student1");  PreparedStatement statement = connection.prepareStatement(sql);  ResultSet resultSet = statement.executeQuery()) {  if (resultSet.next())  return resultSet.getInt(1);  }  return 0;  }  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();  }  }  public static void update(Student student) throws Exception {  String sql = "UPDATE student " +  "SET name=?, studentNumber=?, departmentId=?, year=? " +  "WHERE id=?";  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.setInt(5, student.getId());  statement.executeUpdate();  }  }  public static void delete(int id) throws Exception {  String sql = "DELETE FROM student WHERE id=?";  try (Connection connection = DB.getConnection("student1");  PreparedStatement statement = connection.prepareStatement(sql)) {  statement.setInt(1, id);  statement.executeUpdate();  }  }  } |

# pagination tag 파일

## pagination.tag

WebContent/WEB-INF/tags/pagination.tag

|  |  |
| --- | --- |
| 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 | <%@ tag description="pagination" pageEncoding="UTF-8"%>  <%@ tag import="java.util.ArrayList" %>  <%@ attribute name="recordCount" type="java.lang.Integer" required="true" %>  <%@ attribute name="pageSize" type="java.lang.Integer" required="true" %>  <%@ attribute name="queryStringName" type="java.lang.String" required="true" %>  <%!  private class Page {  int page;  String label;    Page(int page, String label) {  this.page = page;  this.label = label;  }  }  %>  <%  int recordCount = (Integer)jspContext.getAttribute("recordCount");  int pageSize = (Integer)jspContext.getAttribute("pageSize");  String name = (String)jspContext.getAttribute("queryStringName");  int currentPage = 1;  if (request.getParameter(name) != null)  currentPage = Integer.parseInt(request.getParameter(name));  int pageCount = recordCount / pageSize;  if (pageCount \* pageSize < recordCount) ++pageCount;  String queryString = request.getQueryString();  if (queryString == null)  queryString = name + "=@@@";  else if (queryString.matches(".\*" + name + "=[0-9]+.\*"))  queryString = queryString.replaceAll(name + "=[0-9]+", name + "=@@@");  else  queryString = queryString + "&" + name + "=@@@";  Object temp = request.getAttribute("javax.servlet.forward.request\_uri");  String uri = (temp != null) ? temp.toString() : request.getRequestURI();  String url = uri + "?" + queryString;  if (currentPage > pageCount) currentPage = pageCount;  int base = ((currentPage - 1) / 10) \* 10;  ArrayList<Page> pages = new ArrayList<Page>();  if (base > 0) pages.add(new Page(base, "&lt;"));  for (int i = 1; i <= 10; ++i) {  int n = base + i;  if (n > pageCount) break;  pages.add(new Page(n, String.valueOf(n)));  }  int n = base + 11;  if (n <= pageCount)  pages.add(new Page(n, "&gt;"));  %>  <ul class="pagination">  <% for (Page p : pages) { %>  <li class='<%= p.page == currentPage ? "active" : "" %>'>  <a href='<%= url.replace("@@@", String.valueOf(p.page)) %>'><%= p.label %></a>  </li>  <% } %>  </ul> |

# JSP 구현

## studentList5.jsp

WebContent/jdbc2/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  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ page import="java.util.\*, lecture1.jdbc2.\*" %>  <%@ taglib tagdir="/WEB-INF/tags" prefix="my" %>  <%  int currentPage = 1;  int pageSize = 10;  String pg = request.getParameter("pg");  if (pg != null) currentPage = Integer.parseInt(pg);  List<Student> list = StudentDAO3.findAll(currentPage, pageSize);  int recordCount = StudentDAO3.count();  %>  <!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; }  div#create { width: 700px; }  div#create a { float: right; margin-bottom: 5px; }  </style>  </head>  <body>  <div class="container">  <h1>학생목록</h1>  <div id="create">  <a class="btn btn-primary btn-sm" href="studentCreate1.jsp">학생등록</a>  </div>  <table class="table table-bordered table-condensed">  <thead>  <tr>  <th>id</th>  <th>학번</th>  <th>이름</th>  <th>학과</th>  <th>학년</th>  </tr>  </thead>  <tbody>  <% for (Student student : list) { %>  <tr>  <td><%= student.getId() %></td>  <td><%= student.getStudentNumber() %></td>  <td><a href="studentEdit1.jsp?id=<%= student.getId() %>">  <%= student.getName() %></a>  </td>  <td><%= student.getDepartmentName() %></td>  <td><%= student.getYear() %></td>  </tr>  <% } %>  </tbody>  </table>  <my:pagination pageSize="<%= pageSize %>" recordCount="<%= recordCount %>" queryStringName="pg" />  </div>  </body>  </html> |

## studentCreate1.jsp

WebContent/jdbc2/studentCreate1.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> |

## studentEdit1.jsp

WebContent/jdbc2/studentEdit1.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  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ page import="java.util.List, lecture1.jdbc2.\*" %>  <%  if (request.getMethod().equals("POST")) {  Student student = new Student();  request.setCharacterEncoding("UTF-8");    String s = request.getParameter("id");  int id = Integer.parseInt(s);  student.setId(id);    String name = request.getParameter("name");  student.setName(name);    String studentNumber = request.getParameter("studentNumber");  student.setStudentNumber(studentNumber);  s = request.getParameter("departmentId");  int departmentId = Integer.parseInt(s);  student.setDepartmentId(departmentId);  s = request.getParameter("year");  int year = Integer.parseInt(s);  student.setYear(year);    StudentDAO3.update(student);  response.sendRedirect("studentList5.jsp");  return;  }  String s = request.getParameter("id");  int id = Integer.parseInt(s);  Student student = StudentDAO3.findById(id);  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 수정</h1>  <form method="post">  <table class="table table-bordered table-condensed">  <tr>  <td>ID</td>  <td>  <%= student.getId() %>  </td>  </tr>  <tr>  <td>이름</td>  <td>  <input type="text" name="name" value="<%= student.getName() %>" />  </td>  </tr>  <tr>  <td>학번</td>  <td>  <input type="text" name="studentNumber" class="form-control" value="<%= student.getStudentNumber() %>" />  </td>  </tr>  <tr>  <td>학과ID</td>  <td>  <select name="departmentId" class="form-control">  <% for (Department department : departments) { %>  <% int \_id = department.getId(); %>  <% String \_name = department.getDepartmentName(); %>  <% String \_selected = (\_id == student.getDepartmentId() ? "selected" : ""); %>  <option value="<%= \_id %>" <%= \_selected %> >  <%= \_name %>  </option>  <% } %>  </select>  </td>  </tr>  <tr>  <td>학년</td>  <td>  <input type="text" name="year" class="form-control" value="<%= student.getYear() %>" />  </td>  </tr>  </table>  <button type="submit" class="btn btn-primary">저장</button>  <a href="studentDelete1.jsp?id=<%= student.getId() %>" class="btn btn-warning"  onclick="confirm('삭제하시겠습니까?')">삭제</a>  <a href="javascript:window.history.back()" class="btn btn-info">돌아가기</a>  </form>  </div>  </body>  </html> |

## studentDelete1.jsp

WebContent/jdbc2/studentDelete1.jsp

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | <%@ page language="java" contentType="text/html; charset=UTF-8" pageEncoding="UTF-8"%>  <%@ page import="java.util.List, lecture1.jdbc2.\*" %>  <%  String s = request.getParameter("id");  int id = Integer.parseInt(s);  StudentDAO3.delete(id);  response.sendRedirect("studentList5.jsp");  %> |