Class 230116

✔️ ***Connection Pool Basic***

✔️ ***Connection Pool 설정***

✔️ ***Connection Pool 적용***

✔️ ***Summernote 적용***

Connection Pool Basic

JDBC를 사용할 때 가장 많은 리소스 즉 자원이 소모되는 부분이 디비 연동에 필요한 connection 객체를 생성하는 부분이다. 지금까지 방법들은 모두 JSP 에서 SQL 구문을 수행하기 위해서 connection 객체를 생성하고 사용후 제거하는 과정을 반복해왔다. 접속자가 많아질 경우 시스템의 성능을 급격하게 저하시키게 된다.

따라서 이러한 문제점을 해결하기 위한 방법으로 커넥션 풀을 이용하게 된다. 사용자가 접속 할때 마다 매번 새로운 connection객체를 생성하는것이 아니라 일정 개수의 Connection 객체를 미리 생성해 놓고 사용자의 요청이 있을 때마다 가용한 객체를 할당하고 다시 회수하는 방식이다.

Connection Pool 설정

1. context.xml

   데이터 베이스에 대한 커넥션 풀을 사용하기 위한 **설정을 정의**한다.

   위치는 WebContent > META-INF > **context.xml**

```
<?xml version="1.0" encoding="UTF-8"?>
<context>
  <Resource name="jdbc/univ"
  auth="Container"
  type="javax.sql.DataSource"
  driverClassName="com.mysql.jdbc.Driver"
  url="jdbc:mysql://localhost:3306/univ?serverTimezone=UTC"
  username="root"
  password="0000"
  maxTotal="16"
  maxIdle="4"
  maxWaitMillis="-1"/>
</context>
```

2. JDBC connector driver

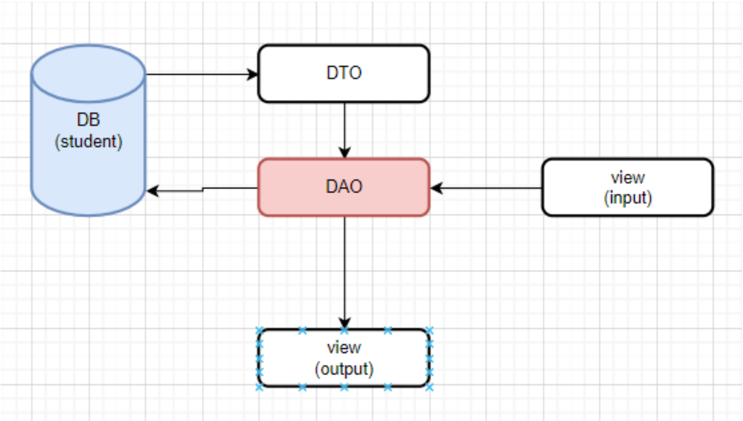
위치는 WebContent > WEB-INF > **lib**

3. ConnectionPool.java

```
package util;

import java.sql.*;
import javax.naming.*;
import javax.sql.DataSource;

public class ConnectionPool {
  private static DataSource _ds = null;

  public static Connection get() throws NamingException, SQLException {
    if(_ds == null) {
      _ds = (DataSource) (new InitialContext()).lookup("java:comp/env/jdbc/univ");
    }
    return _ds.getConnection();
  }
}
```

Connection Pool 적용

항상 DB 설계부터 시작



DTO

Data Transfer Object

사실 DTO 는 디비에서 데이터를 꺼낼때만 사용된다. DTO 파일은 데이터베이스의 테이블의 필드와 일대일 매칭이 되게 설계한다.

```
package jdbc;

public class StudentDTO {
  private String hakbun;
  private String name;
  private String dept;
  private String addr;

  public String getHakbun() {
    return hakbun;
  }
  public void setHakbun(String hakbun) {
    this.hakbun = hakbun;
  }
  public String getName() {
    return name;
  }
  public void setName(String name) {
    this.name = name;
  }
  public String getDept() {
    return dept;
  }
  public void setDept(String dept) {
    this.dept = dept;
  }
  public String getAddr() {
    return addr;
  }
  public void setAddr(String addr) {
    this.addr = addr;
  }
  public StudentDTO(String hakbun, String name, String dept, String addr) {
    super();
    this.hakbun = hakbun;
    this.name = name;
    this.dept = dept;
    this.addr = addr;
  }
}
```

DAO

Data Access Object

실제 DB와 연결되는 매서드 등과 SQL 쿼리 등을 작성하게 된다.

```
package jdbc;

import java.sql.*;
import java.util.ArrayList;
import javax.naming.NamingException;
import util.*;

public class StudentDAO {
  public static int insert(String hakbun, String name, String dept, String addr) throws NamingException, SQLException {

    //C R U D
    String sql = "INSERT INTO student VALUES(?,?,?,?)";

    //Connection Pool 사용
    Connection conn = ConnectionPool.get();

    PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setString(1, hakbun);
      pstmt.setString(2, name);
      pstmt.setString(3, dept);
      pstmt.setString(4, addr);

    int result = pstmt.executeUpdate();

    return result;
  }

  public static ArrayList<StudentDTO> getList() throws NamingException, SQLException {
    String sql = "SELECT * FROM student";

    Connection conn = ConnectionPool.get();

    PreparedStatement pstmt = conn.prepareStatement(sql);

    ResultSet rs = pstmt.executeQuery();

    ArrayList<StudentDTO> students = new ArrayList<StudentDTO>();

    while(rs.next()) {
      students.add(new StudentDTO(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4)));
    }

    return students;
  }

  public static StudentDTO getDetail(String hakbun) throws NamingException, SQLException {
    String sql = "SELECT * FROM student WHERE hakbun=?";

    Connection conn = ConnectionPool.get();

    PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setString(1, hakbun);

    ResultSet rs = pstmt.executeQuery();

    rs.next();

    String name = rs.getString(2);
    String dept = rs.getString(3);
    String addr = rs.getString(4);

    StudentDTO student = new StudentDTO(hakbun, name, dept, addr);

    return student;
  }
}
```

View

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>TBForm.jsp</title>
</head>
<body>
  <form action="TBInsert.jsp">
    학번<input type="text" name="hakbun"><br>
    이름<input type="text" name="name"><br>
    전공<input type="text" name="dept"><br>
    주소<input type="text" name="addr">
    <input type="submit" value="전송">
  </form>
</body>
</html>
```

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import = "jdbc.*" %>
<%-- <!-- Step 1 Import SQL Packages -->
<%@ page import="java.sql.*" %> --%>

<%
  //한글 처리
  request.setCharacterEncoding("utf-8");
%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%
  /* //Step 2 Load JDBC Driver
  try {
    Class.forName("com.mysql.jdbc.Driver");
  } catch(ClassNotFoundException err) {
    out.print("JDBC Driver loading error<br>" + err.getMessage());
  }

  //Step 3 Creare Connection Object
  Connection conn = null;
  try {
    conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/univ","root","0000");
  } catch(SQLException err) {
    out.print("Connection Object error<br>" + err.getMessage());
  } */

  //Step 4 Create Statement Object
  String hakbun = request.getParameter("hakbun");
  String name = request.getParameter("name");
  String dept = request.getParameter("dept");
  String addr = request.getParameter("addr");

  /* String sql ="INSERT student VALUES(?, ?, ?, ?)" ;


  PreparedStatement pstmt = conn.prepareStatement(sql);
    pstmt.setString(1, hakbun);
    pstmt.setString(2, name);
    pstmt.setString(3, dept);
    pstmt.setString(4, addr);

  //Step 5 Execute SQL Query
  pstmt.executeUpdate();

  //Step 6 Close Connection
  pstmt.close();
  conn.close(); */

  int result = StudentDAO.insert(hakbun, name, dept, addr);

%>
</body>
</html>
```

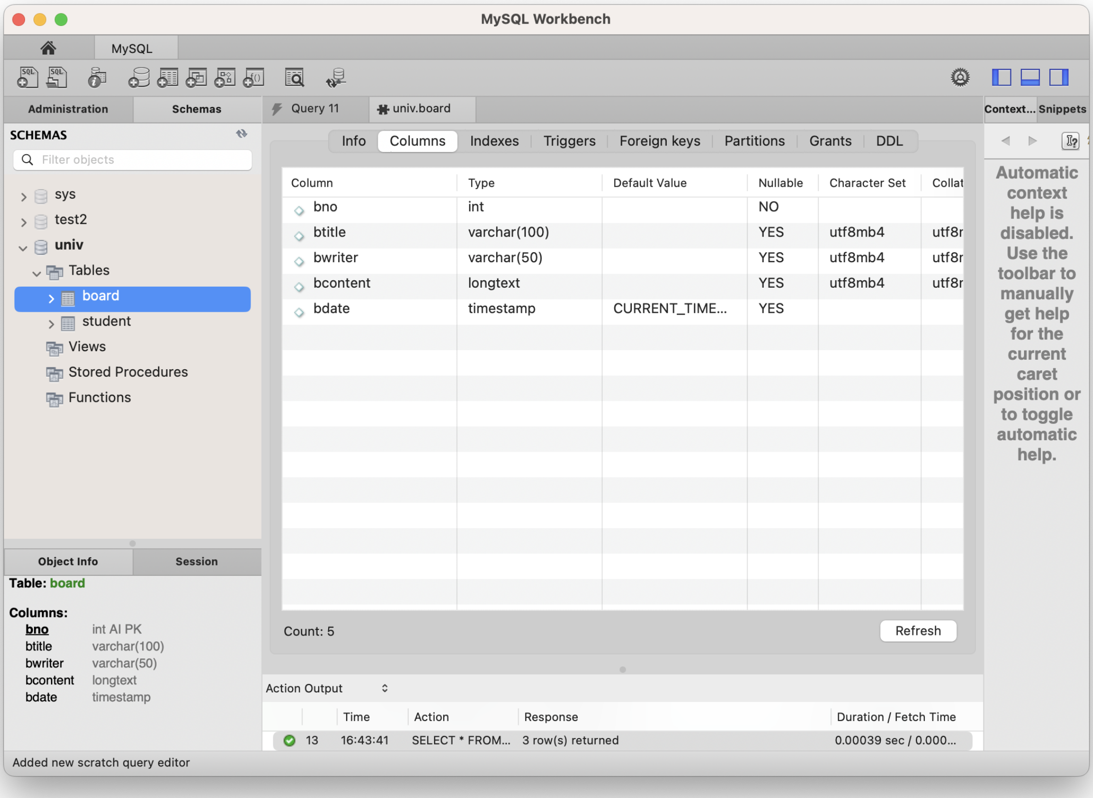
TBList

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import = "jdbc.*, java.util.*" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>학생 목록</title>
</head>
<body>
<%
  ArrayList<StudentDTO> students = StudentDAO.getList();
  for(StudentDTO student : students){
%>

<a href="TBDetail.jsp?hakbun=<%=student.getHakbun() %>"><%=student.getHakbun() %></a>|
<%=student.getName() %><br>

<%
  }
%>
</body>
</html>
```

TBDetail

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import = "jdbc.*, java.util.*" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>학생 목록</title>
</head>
<body>
<%
  String hakbun = request.getParameter("hakbun");
  StudentDTO student = StudentDAO.getDetail(hakbun);
%>

<%=student.getHakbun() %>|
<%=student.getName() %>|
<%=student.getDept() %>|
<%=student.getAddr() %><br>

</body>
</html>
```

Board TABLE



Column	Type	Default Value	Nullable	Character Set	Collat
bno	int		NO		
btitle	varchar(100)		YES	utf8mb4	utf8r
bwriter	varchar(50)		YES	utf8mb4	utf8r
bcontent	longtext		YES	utf8mb4	utf8r
bdate	timestamp	CURRENT_TIME...	YES		

Count: 5

Table: board

Columns:

bno	int AI PK
btitle	varchar(100)
bwriter	varchar(50)
bcontent	longtext
bdate	timestamp

	Time	Action	Response	Duration / Fetch Time
13	16:43:41	SELECT * FROM...	3 row(s) returned	0.00039 sec / 0.000...

Added new scratch query editor

BoardDTO

```
package jdbc;

public class BoardDTO {
  private String bno;
  private String btitle;
  private String bwriter;
  private String bcontent;
  private String bdate;

  public String getBno() {
    return bno;
  }
  public String getBtitle() {
    return btitle;
  }
  public String getBwriter() {
    return bwriter;
  }
  public String getBcontent() {
    return bcontent;
  }
  public String getBdate() {
    return bdate;
  }

  public BoardDTO(String bno, String btitle, String bwriter, String bcontent, String bdate) {
    super();
    this.bno = bno;
    this.btitle = btitle;
    this.bwriter = bwriter;
    this.bcontent = bcontent;
    this.bdate = bdate;
  }
}
```

BoardDAO

```
package jdbc;

import java.sql.*;
import java.util.*;
import javax.naming.NamingException;
import util.ConnectionPool;

public class BoardDAO {
  //게시물 등록
  public static int insert(String btitle, String bwriter, String bcontent) throws NamingException, SQLException {
    String sql = "INSERT INTO board(btitle, bwriter, bcontent) VALUES(?,?,?)";

    Connection conn = ConnectionPool.get();

    PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setString(1, btitle);
      pstmt.setString(2, bwriter);
      pstmt.setString(3, bcontent);

    int result = pstmt.executeUpdate();

    return result;
  }

  //게시물 목록
  public static ArrayList<BoardDTO> getList() throws NamingException, SQLException{
    String sql = "SELECT * FROM board";

    Connection conn = ConnectionPool.get();

    PreparedStatement pstmt = conn.prepareStatement(sql);

    ResultSet rs = pstmt.executeQuery();

    ArrayList<BoardDTO> boards = new ArrayList<BoardDTO>();

    while(rs.next()) {
      boards.add(new BoardDTO(rs.getString(1), rs.getString(2), rs.getString(3), rs.getString(4), rs.getString(5)));
    }

    return boards;
  }

  //게시물 하나
  public static BoardDTO getDetail(String bno) throws NamingException, SQLException {
    String sql = "SELECT * FROM board WHERE bno=?";

    Connection conn = ConnectionPool.get();

    PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setString(1, bno);

    ResultSet rs = pstmt.executeQuery();

    rs.next();

    String btitle = rs.getString(2);
    String bwriter = rs.getString(3);
    String bcontent = rs.getString(4);
    String bdate = rs.getString(5);

    BoardDTO board = new BoardDTO(bno, btitle, bwriter, bcontent, bdate);

    return board;
  }

  //게시물 수정
  public static int update(String bno, String btitle, String bwriter, String bcontent) throws NamingException, SQLException {
    String sql = "UPDATE board SET btitle=?, bwriter=?, bcontent=? WHERE bno=?";

    Connection conn = ConnectionPool.get();

    PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setString(1, btitle);
      pstmt.setString(2, bwriter);
      pstmt.setString(3, bcontent);
      pstmt.setString(4, bno);

    int result = pstmt.executeUpdate();
    return result;
  }

  //게시물 삭제
  public static int delete(String bno) throws NamingException, SQLException {
    String sql = "DELETE FROM board WHERE bno=?";

    Connection conn = ConnectionPool.get();

    PreparedStatement pstmt = conn.prepareStatement(sql);
      pstmt.setString(1, bno);

    int result = pstmt.executeUpdate();

    return result;
  }
}
```

boardForm.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>boardForm.jsp</title>
</head>
<body>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8
<div class="container">
  <form action="boardInsert.jsp">
    <div class="mb-3">
      <label for="exampleFormControlInput1" class="form-label">제목</label>
      <input type="text" name="btitle" class="form-control" id="exampleFormControlInput1">
    </div>
    <div class="mb-3">
      <label for="exampleFormControlTextarea1" class="form-label">내용</label>
      <textarea class="form-control" name="bcontent" id="exampleFormControlTextarea1" rows="3"></textarea>
    </div>
    <button type="submit" class="btn btn-primary">등록</button>
  </form>
</div>
</body>
</html>
```

boardInsert.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import = "jdbc.*" %>
<%
  request.setCharacterEncoding("utf-8");

  String btitle = request.getParameter("btitle");
  String bwriter = "작성자";
  String bcontent = request.getParameter("bcontent");

  int result = BoardDAO.insert(btitle, bwriter, bcontent);
%>
```

boardList.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import = "jdbc.*, java.util.*" %>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
```

```
<title>boardList.jsp</title>
</head>
<body>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8
<table class="table table-hover">
  <thead>
    <tr>
      <th scope="col">No</th>
      <th scope="col">Title</th>
      <th scope="col">Writer</th>
      <th scope="col">Date</th>
    </tr>
  </thead>
  <tbody>
<%
  ArrayList<BoardDTO> boards = BoardDAO.getList();
  for(BoardDTO board : boards){
%>
    <tr>
      <th scope="row"><%=board.getBno() %></th>
      <td><a href="boardDetail.jsp?bno=<%=board.getBno() %>"><%=board.getBtitle() %></a></td>
      <td><%=board.getBwriter() %></td>
      <td><%=board.getBdate() %></td>
    </tr>
  </tbody>
<%
}
%>
</table>

</body>
</html>
```

boardDetail.jsp

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<%@ page import = "jdbc.*, java.util.*" %>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>boardDetail.jsp</title>
</head>
<body>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLhlTQ8
<%
  String bno = request.getParameter("bno");
  BoardDTO board = BoardDAO.getDetail(bno);
%>
<div class="container">
  <div class="mb-3">
    <label for="exampleFormControlInput1" class="form-label">Title</label>
    <input type="text" name="btitle" class="form-control" value="<%=board.getBtitle() %>">
  </div>
  <div class="mb-3">
    <label for="exampleFormControlInput1" class="form-label">Title</label>
    <input type="text" name="bwriter" class="form-control" value="<%=board.getBwriter() %>">
  </div>
  <div class="mb-3">
    <label for="exampleFormControlTextarea1" class="form-label">Content</label>
    <textarea class="form-control" name="bcontent" rows="3"><%=board.getBcontent() %></textarea>
  </div>
  <a href="boardList.jsp" class="btn btn-primary">List</a>
</div>
</body>
</html>
```

Summernote 적용

- 주의사항

1. DB에 필드 사이즈를 크게 LONGTEXT 로 사용해야만 하고

2. 전송 방식을 “post”로 설정해야만 한다.

- 모바일 화면 보기

<meta name="viewport" content="width=device-width, initial-scale=1" />