20230116 Connection Pool

Connection Pool이란?

JDBC를 사용할 때 가장 많이 리소스 즉 자원이 소모되는 부분이 DB연동에 필요한 Connection 객체를 생성하는 부분이다. 지금까지 방법들은 모두 jsp에서 sql 구문을 수행하기 위해서 Connection 객체를 생성하고 사용 후 제거하는 과정을 반복해왔다. 접속사가 많아질 경우 시스템의 성능을 급격하게 저하시키게 된다.

따라서 이러한 문제점을 해결하기 위한 방법으로 Connection Pool을 이용하게 된다. 사용자가 접속 할 때 마다 매번 새로운 Connection 객체를 생성하는 것이 아니라 일정 개수의 Connection 객체를 미리 생성 해놓고 사용자의 요청이 있을 때마다 가용한 객체를 할당하고 다시 회수하는 방식이다.

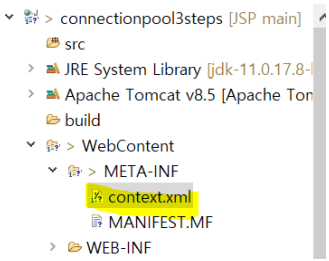
Connection Pool 설정 3단계

1. Connection Pool 설정 정의 context.xml
2. 정의된 내용으로 실제 DB와 연결해주는 클래스 Connection Pool.java
3. JDBC connector driver

1. context.xml

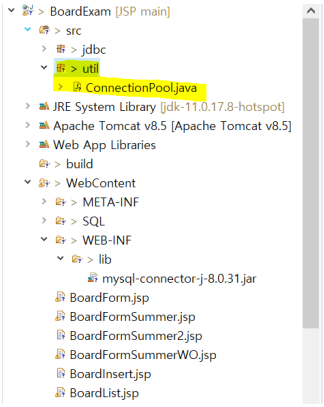
데이터 베이스에 대한 Connection Pool을 사용하기 위한 설정을 정의한다.

위치는 WebContent > META-INF > context.xml




```
<?xml version="1.0" encoding="UTF-8"?>
<Context>
  <Resource name="jdbc/univ"  <-사용할 DB명
  auth="Container"
  type="javax.sql.DataSource"
  driverClassName="com.mysql.jdbc.Driver"
  url="jdbc:mysql://localhost:3306/univ?serverTimezone=UTC"<-사용할 DB명
  username="root" <-DB 아이디 (호스팅 업체 업로드시에는 변경)
  password="0000" <-DB 아이디 (호스팅 업체 업로드시에는 변경)
  maxTotal="16"   <-미리 생성할 커넥션의 갯수
  maxIdle="4"     <-최저 유지 커넥션의 갯수
  maxWaitMillis="-1" />  <-항상 -1 설정, 기다리는 시간 기다리지않고 바로 처리
</Context>

?serverTimezone=UTC" 특정 서버에서 타임존 설정을 하지 않으면 동작하지 않을 때가 있다.
```

2. Connection Pool.java

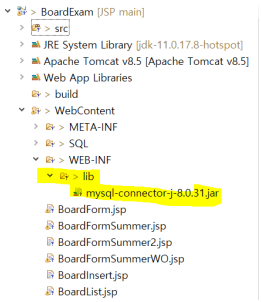


위치는 src > util 패키지 만든 후 > Connection Pool 클래스 생성

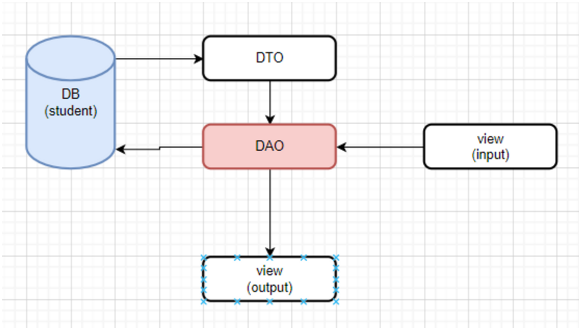
```
package util;

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"); <- 디비명만 바뀐다.
    }
    return _ds.getConnection();
  }
}
```

3. JDBC connector driver

위치는 WebContent>WEB-INF>lib



Connection Pool 적용



1. **DB설계부터 시작하자**

4. **DTO (**Data Transfer Object)

- DTO는 DB에서 데이터를 꺼낼때만 사용된다.
- DTO 파일은 데이터베이스의 테이블의 필드와 일대일 매칭이 되게 설계한다. (DB테이블을 Class화 한다)
- 테이블의 필드명으로 변수를 private 키워드로 생성하고 getter, setter, 생성자를 만든다.

3. **DAO(**Data Access Object)

- 실제 DB와 연결되는 메서드 등과 SQL 쿼리 등을 작성하게 된다.

4. **View**

Connecton Pool 적용 게시판테이블(새로운 DB)

1. 테이블 만들기

▼ 테이블 만드는 과정

Board Table

DB 설계

테이블 명: board

글번호 bno 100

제목 btitle 100

작성자 bwriter 10

내용 bcontent 500

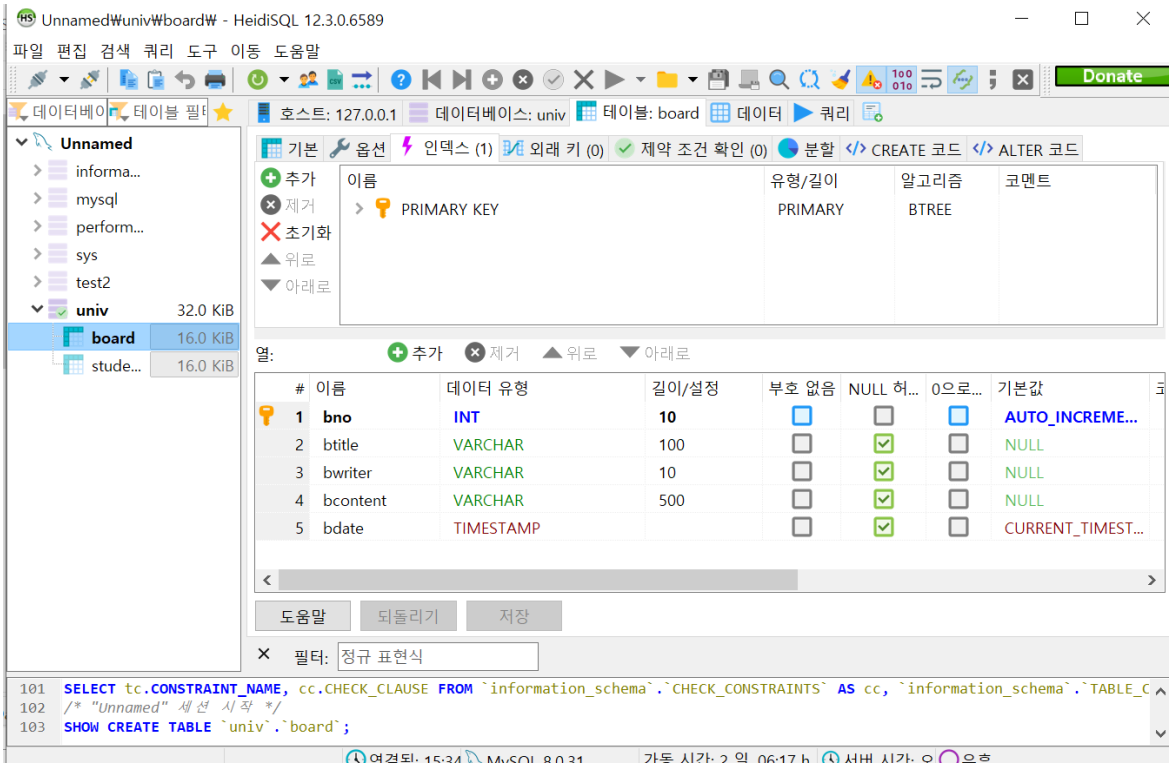
날짜 bdate x

테이블에 데이터가 입력될 때 자동으로 그 시간을 입력한다.

- 데이터 유형-TIMESTAMP()
- 기본값-표현식-current_timestamp()

테이블에 자동 증가 번호넣기

- 데이터유형-INT
- 기본값-AUTO_INCREMENT



2. BoardDTO 만들기

▼ BoardDTO 코드

테이블의 필드명으로 변수를 private 키워드로 생성하고 게터와 세터 그리고 생성자를 만든다.

```
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;
  }

}
```

3. BoardDAO 만들기

▼ BoardDAO 코드

```
package jdbc;

import java.sql.*;
import java.util.ArrayList;

import javax.naming.NamingException;

import util.*;

public class BoardDAO {

  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;
  }

  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;
  }

}
```

4. BoardForm 만들기

▼ 부트스트랩 이용한 BoardForm 코드

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLh
<div class = "container">
<form action = "BoardInsert.jsp">
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">제목</label>
  <input type="text" class="form-control" name="btitle" 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>
```

5. BoardInsert 만들기

▼ BoardInsert 코드

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

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

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

  if(result == 1){
    out.print("등록성공");
  }else{
    out.print("등록 실패");

  }


%>
```

6. BoardList 만들기

▼ 부트스트랩 이용한 BoardList 코드

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


<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>목록</title>
</head>
<body>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLh
<div class="container">
  <table class="table table-hover">

  <thead>
    <tr>
      <th scope="col">번호</th>
      <th scope="col">제목</th>
      <th scope="col">내용</th>
      <th scope="col">작성자</th>
      <th scope="col">날짜</th>
    </tr>
  </thead>
  <tbody>

<%
    ArrayList<BoardDTO> boards = BoardDAO.getList();

    for(BoardDTO board : boards){
%>
  <tr>
      <th scope="row"><%=board.getBno() %></th>
```

```
      <td><%=board.getBtitle() %></td>
      <td><%=board.getBcontent() %></td>
      <td><%=board.getBwriter() %></td>
      <td><%=board.getBdate() %></td>
    </tr>

<%
  }
%>
</tbody>
</table>
</div>
</body>
</html>
```

Summernote 활용하기

Summernote - Super Simple WYSIWYG editor

Super Simple WYSIWYG Editor on Bootstrap Summernote is a JavaScript library that helps you create WYSIWYG editors online.

https://summernote.org/

- 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" />

Summernote 활용한 BoardForm

▼ BoardFormSummer 코드

body 안쪽에 내가 사용할 폼 넣어서 수정하면 된다!

```
<%@ page language="java" contentType="text/html; charset=UTF-8"
    pageEncoding="UTF-8"%>
  <head>
    <meta charset="UTF-8">
    <title>without bootstrap</title>
    <script src="https://code.jquery.com/jquery-3.4.1.slim.min.js" integrity="sha384-J6qa4849blE2+poT4WnyKhv5vZF5SrPo0iEjwBvKU7im0
    <link href="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-lite.min.css" rel="stylesheet">
    <script src="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote-lite.min.js"></script>
    <meta name="viewport" content="width=device-width, initial-scale=1" /> <!-- 모바일용 -->

  </head>
  <body>
<link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-GLh

    <div class = "container">
<form action = "BoardInsert.jsp" method="post">
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">제목</label>
  <input type="text" class="form-control" name="btitle" id="exampleFormControlInput1">
</div>
<div class="mb-3">
  <label for="exampleFormControlTextarea1" class="form-label" >내용</label>
  <textarea class="form-control" name = "bcontent"  id="summernote" rows="3"></textarea>
</div>

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

</form>
</div>
```

```
    <script>
      $('#summernote').summernote({
        placeholder: 'Hello stand alone ui',
        tabsize: 2,
        height: 120,
        toolbar: [
          ['style', ['style']],
          ['font', ['bold', 'underline', 'clear']],
          ['color', ['color']],
          ['para', ['ul', 'ol', 'paragraph']],
          ['table', ['table']],
          ['insert', ['link', 'picture', 'video']],
          ['view', ['fullscreen', 'codeview', 'help']]
        ]
      });
    </script>
  </body>
</html>
```