class 230116

1교시

Connection Pool 쓰는 이유

- JDBC를 사용할 때 가장 많이 리소스가 소모되는 부분

  → 디비 연동에 필요한 Connection 객체를 생성하는 부분
- 이전까지 방법들 모두 JSP에서 SQL 구문을 수행하기위해 Connection객체를 생성, 사용, 제거 과정을 반복
- 접속자가 많아질 경우 시스템의 성능을 급격하게 저하시키게 됨

→ 사용자가 접속 할 때 마다 매번 새로운 Connection 객체를 생성하는것이 아니라

→ 일정 개수의 Conenction 객체를 미리 생성해 놓고 사용자의 요청이 있을 때마다 가용한 객체를 할당, 회수하는 방식

- step2, 3이 커넥션풀로 대체 됨
- step2,3 코드

```
//STEP 2 Load JDBC Driver (lib아래 jar파일과 인식)
  try{
  Class.forName("com.mysql.jdbc.Driver"); //forName 안에 이름이 바뀜 (maria, oracle 등)
  }catch(ClassNotFoundException err){
    out.print("JDBC Driver loading err<br>" + err.getMessage()); // catch 안써도 되는데 에러 확인용
  }
  out.print("JDBC Driver loading success<br>"); //잘 연결됐는지 찍어봄
  // com.mysql.jdbc.Driver  mySQL용

//STEP 3 Create Connection Object (진짜 연결)
//Connection 자료형은 import java.sql에서 옴
  try{
    Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306","root","0000");//root - 아이디, 0000 - 비번
  }catch(SQLException err){
    out.print("Connection Object error<br>" + err.getMessage());
  }
```

Connection Pool 설정

1. context.xml
   - 데이터 베이스에 대한 커넥션 풀을 사용하기 위한 설정을 정의
   - 위치는 WebContent > META-INF > context.xml

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

//?serverTimezone=UTC 특정 서버에서 타임존 설정을 하지 않으면 동작하지 않을때가 있다.
```

2교시

2. ConnectionPool.java

- 정의된 내용으로 실제 DB와 연결 해주는 클래스

```
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();
  }
}
```

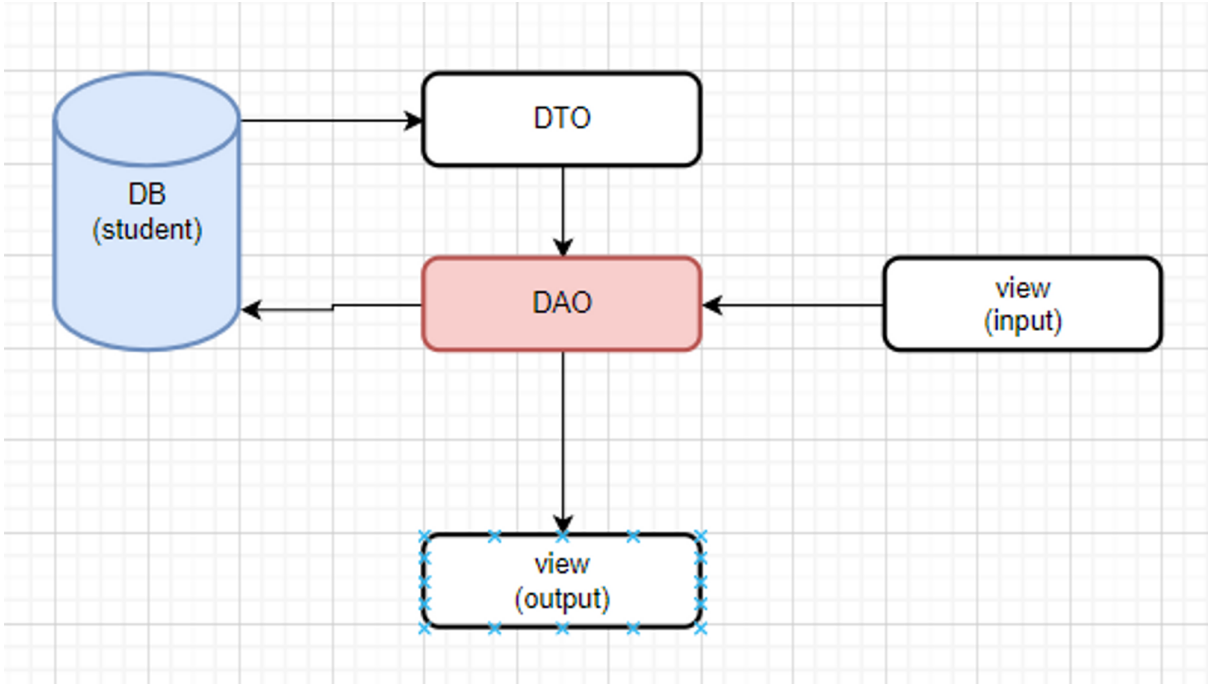
3. JDBC connector driver
    - (mysql_connector-j ~.jar 파일)
    - 위치는 WebContent > WEB-INF > lib

위 3단계로 Connection Pool 사용 설정 완료

Connection Pool 적용

항상 DB 설계부터 시작하자… 일단 지금은 지난주 만든 DB와 테이블을 이용하여 Connection Pool에만 집중하자.

DB 흐름



1. DTO Data Transfer Object → VO와 비슷한 개념 (Getter만 들어가는 애)
    - DB에서 뺄때
    - 사실 DTO는 DB에서 데이터를 꺼낼때만 사용된다.

- DTO 파일은 데이터베이스에 테이블의 필드와 일대일 매칭이 되게 설계
- 테이블의 필드명으로 변수를 private키워드로 생성하고 게터와 세터 그리고 생성자를 만든다.

- StudentDTO.java

```
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;
  }
}
```

2. DAO Data Access Object
   - DB에 넣을때
   - 실제 DB와 연결되어 SQL 쿼리 등을 작성하게 됨

- StudentDAO.java

```
package jdbc;

import java.sql.*;

import javax.naming.NamingException;

import util.*;

public class StudentDAO {

  //테이블에 데이터를 입력하는 메서드
  public static int insert(String hakbun, String name, String dept, String addr) throws NamingException, SQLException {


    String sql = "INSERT INTO student VALUES(?,?,?,?)"; //4개 다넣으면 student 글자 생략가능

    Connection conn = ConnectionPool.get(); //import 했는데 빨간줄     //커넥션 풀 사용

    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(); // SQL 구문 실행 성공여부가 1과 0으로 돌아온다.


    return result;
  }
}
```

4교시

- 목록에는 학번하고 이름만 뜨고 학번을 누르면 한명한명에 디테일 나오게 만들기

▼ TBForm.jsp

- DB에 입력폼

```
<%@ 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>
<form action = TBInsert.jsp method = post>
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"><br>
<input type = "submit" value = "보내기"><p>
</form>
</body>
</html>
```

▼ TBInsert.jsp

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

<% // 전송 받는 데이터 한글 처리
  request.setCharacterEncoding("UTF-8");
%>

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Insert title here</title>
</head>
<body>
<%

  String hakbun = request.getParameter("hakbun");
  String name = request.getParameter("name");
  String dept = request.getParameter("dept");
  String addr = request.getParameter("addr");

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

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

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

▼ TBlist.jsp

- 한명한명의 데이터를 하나의 객체로 만들어 배열로 담는다.

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

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>학생 목록</title>
</head>
<body>
<%
  ArrayList<StudentDTO> students = StudentDAO.getList(); //AllayList 한명의 데이터를 하나의 객체로 만들고 그객체들을 어레이리스트로 담음 //제너릭

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

  <% }

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

▼ TBDetail.jsp

```
<%@page import="jdbc.StudentDTO"%>
<%@page import="jdbc.StudentDAO"%>
<%@page import="java.util.ArrayList"%>
<%@ 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>
<%
  String hakbun = request.getParameter("hakbun");
  StudentDTO student = StudentDAO.getDetail(hakbun);
%>

  <%=student.getHakbun() %><br>
  <%=student.getName() %><br>
  <%=student.getDept() %><br>
  <%=student.getAddr() %><br>
</body>
</html>
```

5교시

- 커넥션 풀 적용 게시판 테이블 연습(새로운 DB)

  ▼ Board Table

  - DB설계

    테이블 명 : board

    글번호 bno 100

    제목 btitle 100

    작성자 bwriter 50

    내용 bcontent 500

    날짜 bdate X

날짜 넣을때 방법

- 데이터 유형 : TIMESTAMP (데이터가 들어올때 시간으로 찍히게)
- 데이터유형 : 기본값 - 표현식 : CURRENT_TIMESTAMP()

테이블에 자동 증가 번호 넣기

- 데이터 유형 - INT
- 기본값 - AUTO_INCREMENT

순서(BoardExam)

- 테이블 만들고 → DTO → DAO

BoardDTO.java

```
package jdbc;

import java.sql.Timestamp;

public class BoardDTO {
  private String bno;
  private String btitle;
  private String bwirter;
  private String bcontent;
  private String bdate;
  public String getBno() {
    return bno;
  }
  public String getBtitle() {
    return btitle;
  }
  public String getBwirter() {
    return bwirter;
  }
  public String getBcontent() {
    return bcontent;
  }
  public String getBdate() {
    return bdate;
  }

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


}

//다 String인 이유
//어차피 갖다 쓸때는 문자취급함 -> 데이터베이스에서는 형식적으로 저렇게
//if (홈쇼핑)가격 -> 그래도 stirng 그리고 계산해야되면 형변환
//어차피 setter는 안씀
```

BoardDAO.java

```
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 SQLException, NamingException {
    String sql = "INSERT INTO board(btitle, bwriter, bcontent) VALUES(?,?,?)";
```

```
    Connection conn = ConnectionPool.get(); //import 했는데 빨간줄     //커넥션 풀 사용  //DB연결완료

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


    int result = pstmt.executeUpdate(); // SQL 구문 실행 성공여부가 1과 0으로 돌아온다.


    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;

  }
}
```

6교시

BoardInsert.jsp

```
<%@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("등록 성공");
  }else{
    out.print("등록 실패");
  }
%>
```

BoardList.jsp

```
<%@page import="jdbc.*"%>
<%@page import="java.util.ArrayList"%>
<%@ 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-GLhlTQ8

<div class = "container">
```

```
<table class="table table-hover">
  <thead>
    <tr>
      <th scope="col">번호</th>
      <th scope="col">제목</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.getBwirter() %></td>
      <td><%=board.getBdate() %></td>
    </tr>
<%
  }
%>
</tbody>
</table>
</div>
</body>
</html>
```

7교시

▼ BoardForm.jsp

```
<%@ 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" 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>
```

▼ BoardFormSummer.jsp

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <title>Summernote</title>
  <link href="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css" rel="stylesheet">
  <script src="https://code.jquery.com/jquery-3.5.1.min.js"></script>
  <script src="https://stackpath.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></script>
  <link href="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote.min.css" rel="stylesheet">
  <script src="https://cdn.jsdelivr.net/npm/summernote@0.8.18/dist/summernote.min.js"></script>
</head>
<body>
  <form action = "BoardInsert.jsp" method ="post">
<div class="mb-3">
```

```
  <label for="exampleFormControlInput1" class="form-label">TITLE</label>
  <input type="text" name = "btitle" class="form-control" id="exampleFormControlInput1" >
</div>
<div class="mb-3">
  <label for="exampleFormControlTextarea1" class="form-label">CONTENT</label>
  <textarea class="form-control" name = "bcontent" id="summernote" rows="3"></textarea>
</div>

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


  <script>
    $(document).ready(function() {
        $('#summernote').summernote();
    });
  </script>
</body>
</html>
```

- 서머노트를 쓸때는 크기를 엄청 크게 해야됨 (LONGTEXT)
- 요청헤더가 너무큽니다 문제
  - 정보가 get으로가서 ⇒ post로 써야됨
- 썸머노트 부트스트랩 5가 안먹음 , Without B
  - without받아서 부트스트랩5먹임

```
<html lang="en">
  <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+poT4WnyKhv5vZF5SrPo0iEj
    <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" />



  </head>
  <body>
  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0-alpha1/dist/css/bootstrap.min.css" rel="stylesheet" integrity=
  <div class = "container">
<form action = "BoardInsert.jsp" method="post">
<div class="mb-3">
  <label for="exampleFormControlInput1" class="form-label">title</label>
  <input type="text" name = "btitle" class="form-control" id="exampleFormControlInput1" >
</div>
<div class="mb-3">
  <label for="exampleFormControlTextarea1" class="form-label">content</label>
  <textarea class="form-control" name = "bcontent" id="summernote" rows="3"></textarea>
</div>

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

    <div id="summernote"></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>
```

- 모바일화면 (부트스트랩 먹여서 가능), 썸머노트 장점
  - 헤더부분에

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