

JSP Servlet 기초(11) JDBC

JSP 2018. 1. 31. 14:41

이번 글 은 JDBC 에 대해 알아보겠습니다

JAVA 프로그램에서 SQL문을 실행하여 데이터를
관리하기 위한 JAVA API 입니다

JDBC의 특징은 다양한 데이터베이스에 대해서
별도의 프로그램을 만들 필요 없이 해당 데이터 베이스의
JDBC 를 이용하면 하나의 프로그램으로 데이터 베이스를
관리하실수 있습니다.

저는 오라클을 사용하므로 오라클용 JDBC를 사용하며
이것은 오라클을 설치하면 자동으로 설치되고
이클립스에서 해당 클래스 파일을 복사하면 됩니다.

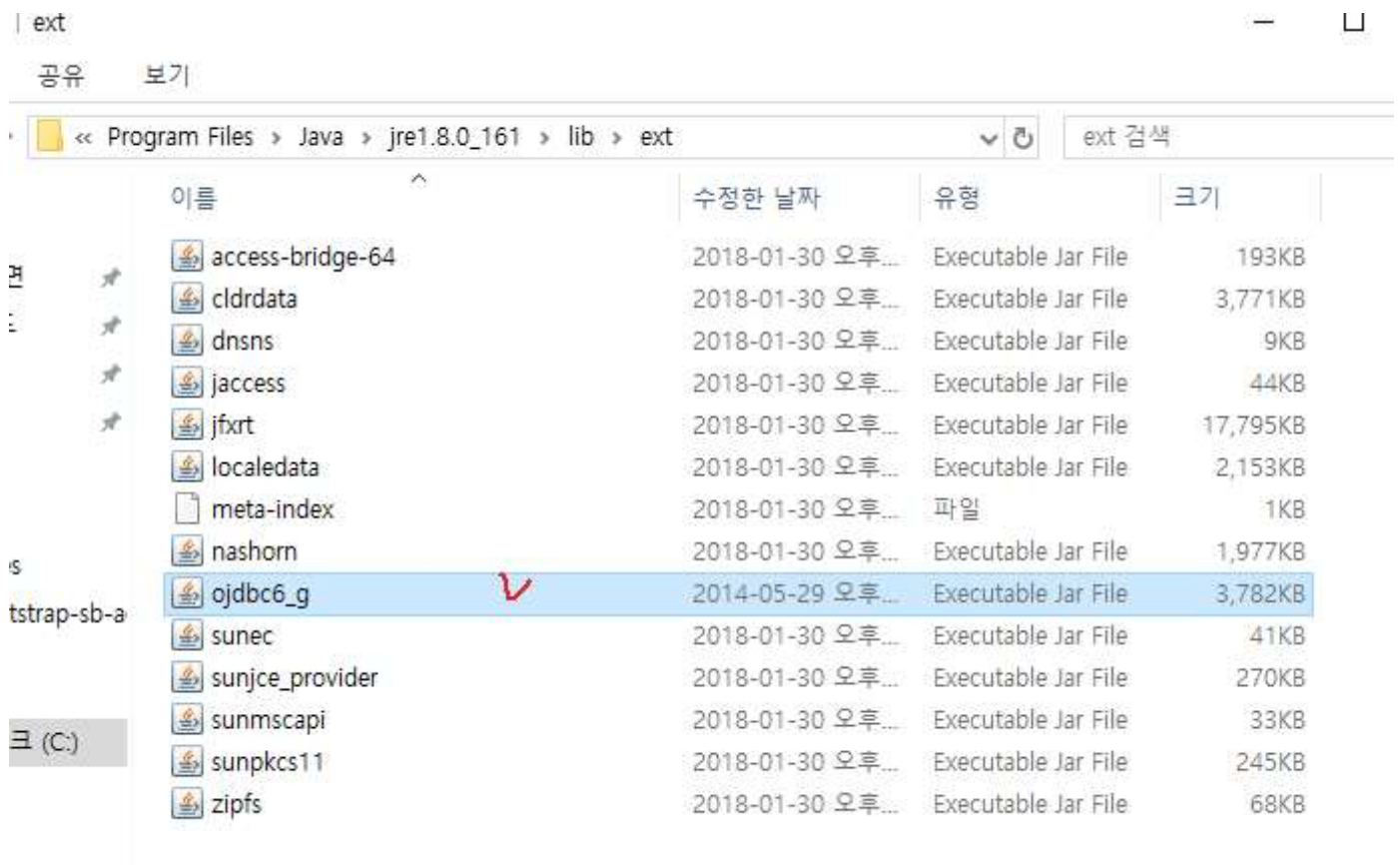
먼저 오라클 드라이버를 사용하기 위한 jdbc.jar 파일을
복사하셔야 합니다
jar 파일은 오라클 폴더 안에있습니다
(저는 아래 경로에 있습니다.)

C:\Woraclexe\app\Woracle\Wproduct\W11.2.0\server\Wjdbc\Wlib

app > oracle > product > 11.2.0 > server > jdbc > lib		
이름	수정한 날짜	유형
 ojdbc5	2014-05-29 오후...	Exec
 ojdbc6	2014-05-29 오후...	Exec
 ojdbc6_g 	2014-05-29 오후...	Exec

ojdbc6_g 파일을 복사하여

C:\WProgram Files\Java\Wjre1.8.0_161\Wlib\Wext
위 경로에 붙여넣기 합니다

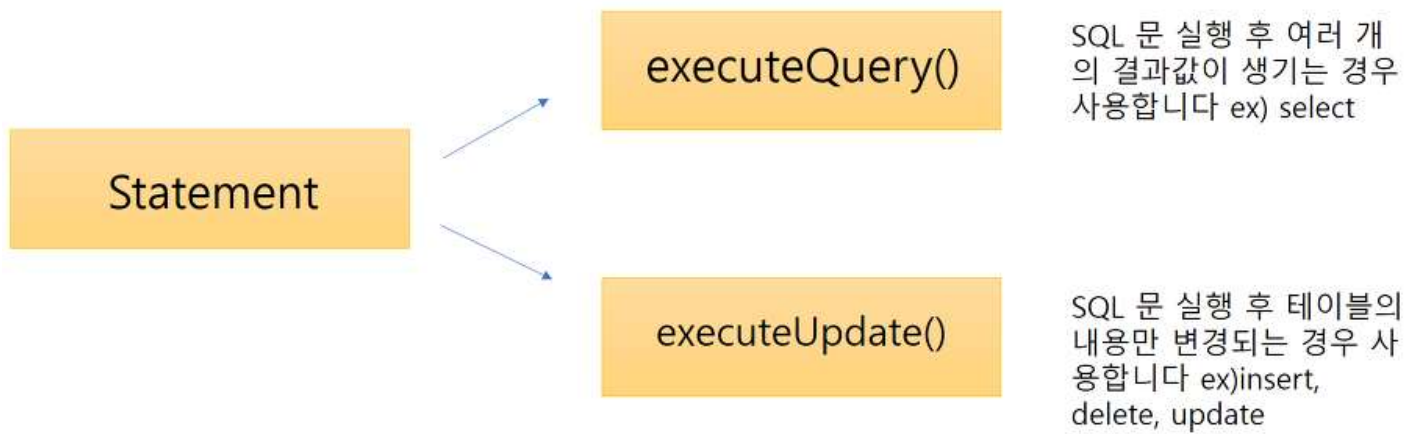


이렇게 JDBC 설치는 끝났습니다 오라클에 있는 jar파일(라이브러리)만 복사 붙여넣기 하시면 됩니다.

데이터 베이스 연결 순서 알아보겠습니다.



Statement 객체 알아보겠습니다



이제 직접 jsp 파일을 만들어 실행해 보겠습니다
먼저 jsp 파일을 하나 만들어줍니다
저는 userData.jsp 라고 만들었습니다.
오라클 developer 포스팅때 생성했던
scott 계정에 test 테이블에서
데이터를 뽑아 오겠습니다

먼저 jsp 파일을 아래와 같이 만들어줍니다

```
1 <%@ page language="java" contentType="text/html; charset=UTF-8"
2   pageEncoding="UTF-8"%>
3
4 <%@page import="java.sql.DriverManager"%>
5 <%@page import="java.sql.ResultSet"%>
6 <%@page import="java.sql.Statement"%>
7 <%@page import="java.sql.Connection"%>
8   <%!
9       Connection connection;
10      Statement statement;
11      ResultSet resultSet;
12
13      String driver = "oracle.jdbc.driver.OracleDriver";
14      String url = "jdbc:oracle:thin:@localhost:1521:xe";
15      String uid = "scott";
16      String upw = "tiger";
17      String query = "select * from test";
18   %>
19 <!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.v
20 <html>
21 <head>
22 <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
23 <title>Insert title here</title>
24 </head>
25 <body>
26 <%
27     try{
28
29         Class.forName(driver);    //oracle.jdbc.driver.OracleDriver
30         connection = DriverManager.getConnection(url, uid, upw);    //jc
```

```

31      statement = connection.createStatement();
32      resultSet = statement.executeQuery(query);      //select * from me
33
34      while(resultSet.next()){
35          String name = resultSet.getString("name");
36          String id = resultSet.getString("id");
37
38          out.println("이름 : " + name + ", 아이디 : " + id + "<br />");
39      }
40
41  } catch(Exception e) {
42  } finally {
43      try{
44          if(resultSet != null) resultSet.close();
45          if(statement != null) statement.close();
46          if(connection != null) connection.close();
47      } catch(Exception e){}
48  }
49  %>
50 </body>
51 </html>

```

먼저 9,10,11line 에서는 Connection, Statement, ResultSet 객체 사용을 위해 레퍼런스 변수를 선언했습니다.

다음 13~17line 에서는 데이터베이스 연결을 위해 JDBC드라이버를 로드합니다

27line 은 예외처리문으로 db와 관련된것은 예외 상황이 발생 할수 있기때문에 try catch 문을 사용합니다

29line은 드라이버로드 한겁니다
30~32line 은 9~11line에서 선언했던 객체를 사용합니다.

다음 34line부터는 17line 에서 db에서 select를 이용해 뽑아온 데이터를 반복문을 통해 화면에 출력시켜줍니다

17line에서 select 문으로 뽑아온 데이터들은 resultSet에 담겨집니다
그래서 resultSet.getString("name"); 으로 데이터를 뽑아와 String name 에 저장합니다

34line 에 resultSet.next()
는 resultSet에서 다음레코드로 이동시켜줍니다
즉 현재 db에 뽑아올 다음 데이터가 있는지 사용하는겁니다
true,false 를 반환합니다
false를 반환하게 되면 다음 데이터가 없는것으로 판단되어 while 문을 빠져나오게 됩니다.

44~46line 은 자원 해제 입니다 위에서 생성했던

객체순서의 역순으로 차례대로 해제시켜줍니다.

이제 `jsp` 파일을 실행시켜보면

이름 : `aaa`, 아이디 : `123`

데이터가 정상적으로 출력되는것을 확인할 수
있습니다.