1. 在apache-tomcat-8.0.44\lib放入sqljdbc4.jar
2. 在eclipse專案視窗中Server\Tomcat v8.0 Server at localhost-config\context.xml

增加sql連線參數(以MS　SQL為例)

<Resource name=*"jdbc/DBPool"*

type=*"javax.sql.DataSource"*

auth=*"Container"*

username=*"USERNAME"*

password=*"PASSWORD"*

url=*"jdbc:sqlserver://localhost:1433;databasename=DB NAME"*

driverClassName=*"com.microsoft.sqlserver.jdbc.SQLServerDriver"*/>

Name:供使用者分辨不同連線路徑的名稱

url:設置連線ip、DB名稱

driverClassName:連結資料庫的驅動程式

1. 設置web.xml(提供java與context.xml之間的連接)

<resource-ref>

<description>MSSQL DB Connection</description>

<res-ref-name>jdbc/DBPool</res-ref-name>

<res-type>javax.sql.DataSource</res-type>

<res-auth>Container</res-auth>

</resource-ref>

1. 在java中呼叫連線
2. 取得連線

Context ctx = new InitialContext();

DataSource ds = (DataSource) ctx.lookup("java:comp/env/jdbc/DBPool");

Connection con = ds.getConnection();

1. 建立查詢指令

(createStatement參數詳見附錄B//PreparedStatement詳見附錄C)

Statement stmt = con.createStatement();

String sql = "select \* from Employee\_Z00040874";

1. 執行SQL

-查(回傳查詢結果--ResultSet)

ResultSet rs;

rs = stmt.executeQuery(sql);

-新刪修(回傳修改筆數--int)

int changed;

changed = stmt. executeUpdate (sql);

1. 處理回傳結果

詳見附錄A

1. 關閉連線

rs.close();

stmt.close();

con.close();

1. Import package

<%@page import="java.sql.\*"%>

<%@page import="javax.naming.\*"%>

<%@page import="javax.sql.DataSource"%>

**附錄A—ResultSet使用**

1. 新增資料:

移動指標

rs.updateString(欄位 (int), "欄位值");

rs.updateString(欄位(int), "欄位值");

rs.updateString("欄位名稱", "欄位值");

rs.insertRow();//新增

1. 刪除資料

移動指標

rs.deleteRow(); //刪除該筆記錄

1. 更新資料

rs.updateString(欄位格(int), "欄位值");  
rs.updateString(欄位格(int), "欄位值");  
rs.updateString("欄位名稱", "欄位值");  
rs.updateRow();

[cancelRowUpdates](https://docs.oracle.com/javase/7/docs/api/java/sql/ResultSet.html#cancelRowUpdates())()//取消更新

1. 移動指標

rs.moveToInsertRow(); //移到指標

rs.first(); //第一筆記錄

rs.absolute(-1); //移到第total筆記錄

rs.absolute(14); //移到第14筆記錄

rs.relative(-10); //移到第4筆記錄 (14-10=4)

rs.First() 移到第一筆資料

rs.Last() 移到最後一筆資料

rs.next() 移到下一筆資料

rs.previous() 移到上一筆資料

rs.absolute(n) 移到第N筆資料

rs.BeforeFirst() 移動到第一筆之前

rs.AfterLast() 移動到最後一筆之後

rs.moveToCurrentRow() 移動到上一次記載的位置

rs.moveToInsertRow() 移動到欲新增的位置

1. 取得資料

rs.getRow() 取ROW的值

rs.getString(n) 取第N個欄位的資料

rs.getString(“欄位名”) 取”欄位名”欄位的資料

1. 判斷

rs.isBeforeFirst() 判斷是否在第一筆之前

rs.isAfterLast() 判斷是否在最後一筆之後

rs.isFirst() 判斷是否在第一筆

rs.isLast() 判斷是否在最後一筆

rs.wasNull() 判斷是否為空值

**附錄B—CreateStatement參數(讀取方法//控制權限)**

1. 讀取方法

ResultSet.TYPE\_FORWARD\_ONLY**(Default)**

--只能用rs.next()移動游標

ResultSet.TYPE\_SCROLL\_INSENSITIVE

--可以任意移動游標，但不會反應資料庫中的資料修改

ResultSet.TYPE\_SCROLL\_SENSITIVE

--可以任意移動游標，且會反應資料庫中的資料修改

1. 控制權限

ResultSet.CONCUR\_READ\_ONLY**(Default)**

--只能讀取

ResultSet.CONCUR\_UPDATABLE

--可以改變DB資料

**附錄C—PreparedStatement**

PreparedStatement能夠放入可變SQL語法，於使用時放入參數再執行其指令

Example:

String sqlString = "select \* from db\_user where username=**?** and password=**?**";

    PreparedStatement stmt = connection.prepareStatement(sqlString);

    stmt.setString(1, username);

    stmt.setString(2, pwd);

    ResultSet rs = stmt.executeQuery();