

```
1 Using Connection Pooling with JNDI
2
3 1. JNDI(Java Naming And Directory Interface)
4 1)Naming service 및 directory service에 접근하기 위한 API
5 2)필요한 data를 name/value 형식으로 저장하고, 필요한 data는 name 값을 이용하여 value값을 얻는 형태
6 3)대표적인 경우가 DNS
7 4)Web browser에서 영문형식으로 domain name으로 ip값을 얻어오는 방식이 JNDI 와 유사
8 5)Database용 JNDI는 "jdbc/데이터베이스버전" 형식으로 지정
9
10
11 2. Lab
12 1)server.xml
13 <Context docBase="JNDIDemo" path="/JNDIDemo" reloadable="true"
14   source="org.eclipse.jst.jee.server:JNDIDemo">
15   <Resource auth="Container" driverClassName="oracle.jdbc.driver.OracleDriver"
16     maxActive="10" maxIdle="10"
17     maxWait="1000" name="jdbc/myoracle" password="tiger"
18     type="javax.sql.DataSource"
19     url="jdbc:oracle:thin:@localhost:1521:XE" username="scott"/>
20   </Context>
21
22 2)web.xml
23 <resource-ref>
24   <description>JNDI Test</description>
25   <res-ref-name>jdbc/myoracle</res-ref-name>
26   <res-type>javax.sql.DataSource</res-type>
27   <res-auth>Container</res-auth>
28 </resource-ref>
29
30 3)src/com.example.dao.DBConnection.java
31 package com.example.dao;
32
33 import java.sql.Connection;
34 import java.sql.SQLException;
35
36 import javax.naming.Context;
37 import javax.naming.InitialContext;
38 import javax.naming.NamingException;
39 import javax.sql.DataSource;
40
41 public class DBConnection {
42   public static Connection getConnection(){
43     Connection conn = null;
44     try {
45       Context context = new InitialContext();
46       Context env = (Context)context.lookup("java:comp/env");
47       DataSource ds = (DataSource)env.lookup("jdbc/myoracle");
48       conn = ds.getConnection();
49     } catch (NamingException e) {
50       System.out.println(e.getMessage());
51     } catch (SQLException e){
52       System.out.println(e.getMessage());
53     }
54   }
55   return conn;
```

```

52     }
53 }
54
55 4)Create Stored Procedure sp_select
56 CREATE OR REPLACE PROCEDURE sp_select
57 (
58     v_deptno IN employees.department_id%TYPE,
59     employee_records OUT SYS_REFCURSOR
60 )
61 AS
62 BEGIN
63     OPEN employee_records FOR
64     SELECT employee_id, first_name, salary,
65            TO_CHAR(hire_date, 'YYYY-MM-DD') AS hiredate,
66            department_name, city, e.department_id AS deptno
67     FROM employees e INNER JOIN departments d ON e.department_id = d.department_id
68            INNER JOIN locations l ON d.location_id = l.location_id
69     WHERE e.department_id = v_deptno;
70 END;
71 /
72
73 5)Build Path에 oracle driver 추가하기
74 -project > right-click > Build Path > Configure Build Path...
75 -Libraries tab > Click [Add External JARs...]
76 -Select ojdbc6.jar > Click [Apply and Close]
77
78 6)WebContent/WEB-INF/lib에 jar file 추가
79 -ojdbc6.jar
80 -taglibs-standard-impl-1.2.5.jar
81 -taglibs-standard-spec-1.2.5.jar
82
83 7)src/com.example.vo.EmployeeVO.java
84 package com.example.vo;
85
86 public class EmployeeVO {
87     private int employee_id;
88     private String first_name;
89     private double salary;
90     private String hiredate;
91     private String department_name;
92     private String city;
93     private int departno;
94
95     public EmployeeVO() {}
96
97     public int getEmployee_id() {
98         return employee_id;
99     }
100
101     public void setEmployee_id(int employee_id) {
102         this.employee_id = employee_id;
103     }
104
105     public String getFirst_name() {

```

```
106     return first_name;
107 }
108
109 public void setFirst_name(String first_name) {
110     this.first_name = first_name;
111 }
112
113 public double getSalary() {
114     return salary;
115 }
116
117 public void setSalary(double salary) {
118     this.salary = salary;
119 }
120
121 public String getHiredate() {
122     return hiredate;
123 }
124
125 public void setHiredate(String hiredate) {
126     this.hiredate = hiredate;
127 }
128
129 public String getDepartment_name() {
130     return department_name;
131 }
132
133 public void setDepartment_name(String department_name) {
134     this.department_name = department_name;
135 }
136
137 public String getCity() {
138     return city;
139 }
140
141 public void setCity(String city) {
142     this.city = city;
143 }
144
145 public int getDepartno() {
146     return departno;
147 }
148
149 public void setDepartno(int departno) {
150     this.departno = departno;
151 }
152
153 @Override
154 public String toString() {
155     return "EmployeeVO [employee_id=" + employee_id + ", first_name=" + first_name + ",
156         salary=" + salary
157         + ", hiredate=" + hiredate + ", department_name=" + department_name + ", city="
158         + city
159         + ", department_id=" + departno + "];"
```

```
158     }
159
160 }
161
162 8)src/com.example.dao.EmployeeDao.java
163     package com.example.dao;
164
165     import java.sql.CallableStatement;
166     import java.sql.Connection;
167     import java.sql.ResultSet;
168     import java.sql.SQLException;
169     import java.util.ArrayList;
170
171     import com.example.vo.EmployeeVO;
172
173     public class EmployeeDao {
174     public static ArrayList<EmployeeVO> selectAll(int deptno) throws SQLException{
175         ArrayList<EmployeeVO> list = new ArrayList<EmployeeVO>();
176         Connection conn = DBConnection.getConnection();
177         CallableStatement cstmt = conn.prepareCall("{ call sp_select(?, ?) }");
178         cstmt.setInt(1, deptno);
179         cstmt.registerOutParameter(2, oracle.jdbc.OracleTypes.CURSOR);
180         cstmt.executeUpdate();
181         ResultSet rs = (ResultSet)cstmt.getObject(2);
182         while(rs.next()) {
183             EmployeeVO emp = new EmployeeVO();
184             emp.setEmployee_id(rs.getInt("employee_id"));
185             emp.setFirst_name(rs.getString("first_name"));
186             emp.setSalary(rs.getDouble("salary"));
187             emp.setHiredate(rs.getString("hiredate"));
188             emp.setDepartment_name(rs.getString("department_name"));
189             emp.setCity(rs.getString("city"));
190             emp.setDepartno(rs.getInt("deptno"));
191             list.add(emp);
192         }
193         if(rs != null) rs.close();
194         if(cstmt != null) cstmt.close();
195         return list;
196     }
197 }
198
199 9)src/com.example.service.EmployeeService.java
200     package com.example.service;
201
202     import java.sql.SQLException;
203     import java.util.ArrayList;
204
205     import com.example.dao.EmployeeDao;
206     import com.example.vo.EmployeeVO;
207
208     public class EmployeeService {
209     private int deptno;
210     private ArrayList<EmployeeVO> list;
211
```

```

212     public void setDeptno(int deptno) {
213         this.deptno = deptno;
214     }
215
216     public ArrayList<EmployeeVO> getList() {
217         ArrayList<EmployeeVO> list = null;
218         try {
219             list = EmployeeDao.selectAll(this.deptno);
220         } catch (SQLException ex) {
221             System.out.println(ex);
222         }
223         return list;
224     }
225 }
226
227 10) dbtest.jsp
228 <%@ page language="java" contentType="text/html; charset=UTF-8"
229     pageEncoding="UTF-8"%>
230 <%@ page import="java.util.ArrayList, com.example.vo.EmployeeVO" %>
231 <%@ taglib prefix="c" uri="http://java.sun.com/jsp/jstl/core" %>
232 <jsp:useBean id="service" class="com.example.service.EmployeeService" />
233 <c:set target="${service}" property="deptno" value="${empty param.deptno ? 10 :
234     param.deptno}"/>
235 <!DOCTYPE html>
236 <html>
237     <head>
238         <meta charset="UTF-8">
239         <title>사원명단</title>
240     </head>
241     <body>
242         <h1>사원 명단(부서번호 : <c:out value="${empty param.deptno ? 10 : param.deptno}"
243             />)</h1>
244         <table border="1">
245             <thead>
246                 <tr>
247                     <th>사원번호</th> <th>사원이름</th> <th>봉급</th>
248                     <th>입사일자</th> <th>부서이름</th> <th>부서위치</th>
249                     <th>부서번호</th>
250                 </tr>
251             </thead>
252             <tbody>
253                 <c:forEach items="${service.list}" var="emp">
254                     <tr>
255                         <td>${emp['employee_id']}</td> <td>${emp['first_name']}</td> <td>${emp.sal
256                             ary}</td>
257                         <td>${emp.hiredate}</td> <td>${emp['department_name']}</td> <td>${emp.ci
258                             ty}</td>
259                         <td>${emp.deptno}</td>
260                     </tr>
261                 </c:forEach>
262             </tbody>
263         </table>
264     </body>
265 </html>

```