

## 커넥션 풀 DBCP API 설치 및 환경파일 세팅

### 1. 자카르타(Jakarta) DBCP API를 이용한 커넥션풀

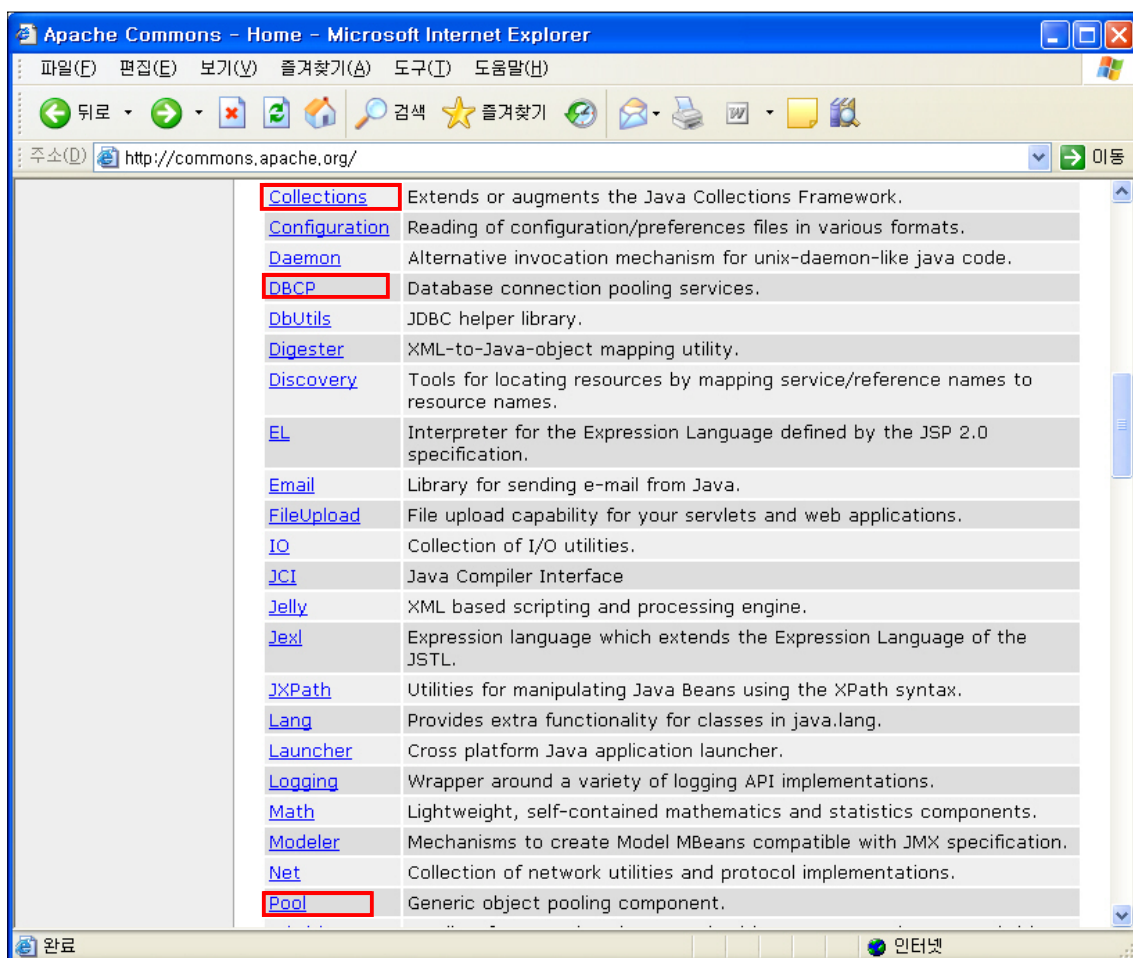
자카르타 프로젝트(Jakarta Project)의 DBCP(Database Connection Pool) API를 이용해서 커넥션풀을 사용해 보자.

자카르타 프로젝트의 DBCP API를 사용해서 커넥션풀을 사용하려면 다음과 같은 단계를 거쳐야 한다.

- (1) 자카르타(Jakarta) DBCP API관련 jar파일 설치
- (2) DBCP에 관한 정보 설정 - server.xml
- (3) JNDI 리소스 사용 설정 - web.xml
- (4) JSP페이지에서 커넥션 풀 사용

#### (1) DBCP API관련 jar파일 설치

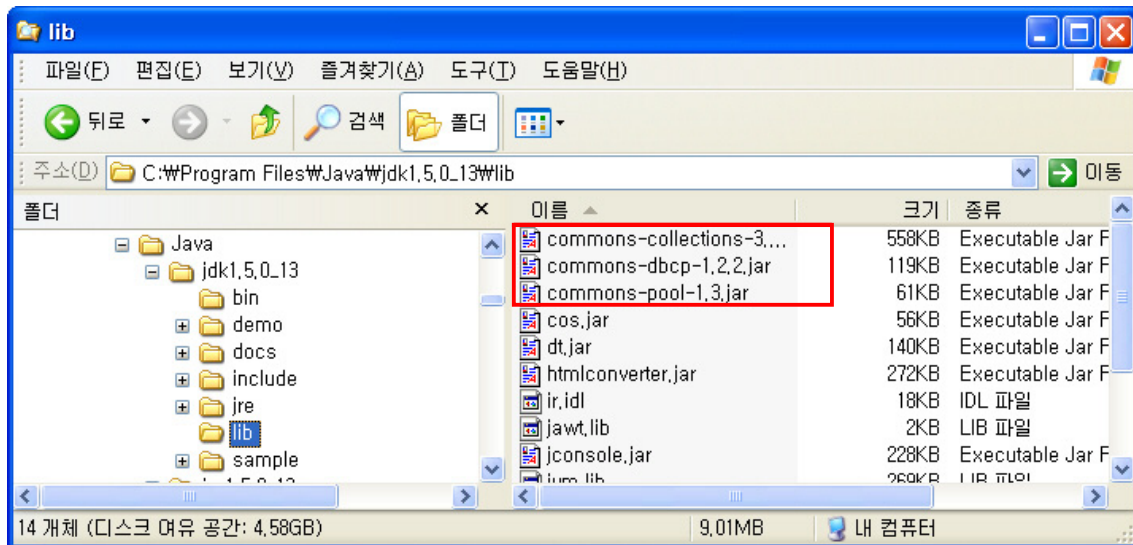
웹 브라우저에 <http://commons.apache.org/> 사이트로 이동해서 [Components]항목의 "Collections", "DBCP", "Pool" 항목을 각각 클릭해서 3개의 zip 파일을 다운로드 받아야 한다.



<DBCP API관련파일 다운로드>

다운로드를 하면 다음과 같은 파일들이 commons-collections-3.2.zip, commons-dbcp-1.2.2.zip, commons-pool-1.3.zip 다운로드된 것을 알 수 있다.

이 파일들의 압축을 풀면 각각 다음과 같은 jar파일을 확인할 수 있다. commons-dbcp-1.2.2.jar, commons-collections-3.2.jar, commons-pool-1.3.jar 이들을 복사해서 자바설치드라이브\Program Files\Java\jdk1.5.0\_13\lib 폴더에 붙여 넣기한다 필자의 경우에는 C:\Program Files\Java\jdk1.5.0\_13\lib폴더 안에 복사 했다.



<lib 폴더에 커넥션풀관련 3개의 jar파일 복사>

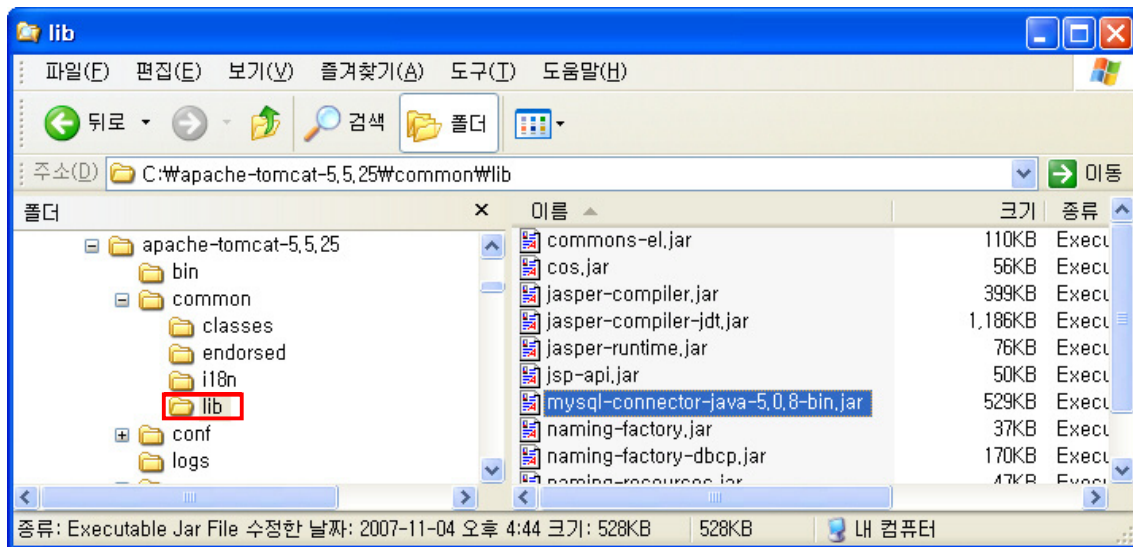
이들 세 개의 jar파일은 커넥션 풀의 사용에 반드시 필요한 것으로, 이들 3개의 jar파일의 쓰임은 다음과 같다.

- ◆ commons-dbcp-1.2.2.jar : 자카르타(Jakarta) DBCP API관련 jar파일
- ◆ commons-collections-3.2.jar : 자카르타(Jakarta) DBCP API가 사용하는 자카르타 Pool API의 jar파일
- ◆ commons-pool-1.3.jar : Pool API가 사용하는 자카르타 Collection API의 jar파일

자 이제 우리는 이클립스의 프로젝트에 이들 3개의 jar파일을 가져오기 할것이다.

그런데 한가지 문제가 있다. DBCP API를 사용할 경우 웹 어플리케이션 폴더 \WEB-INF\lib에 들어있는 JDBC 커넥터인 mysql-connector-java-5.0.8-bin.jar 파일을 인식 못할 수 있다. 즉, 이 파일을 복사해서 톰캣홈\common\lib 폴더(톰캣6.0의 경우 톰캣홈\lib 폴더)에 복사해야 한다.

자바설치드라이브\Program Files\Java\jdk1.5.0\_13\lib폴더에 있는 mysql-connector-java-5.0.8-bin.jar파일을 톰캣홈\common\lib 폴더(톰캣6.0의 경우 톰캣홈\lib 폴더)에 붙여넣기 한다.



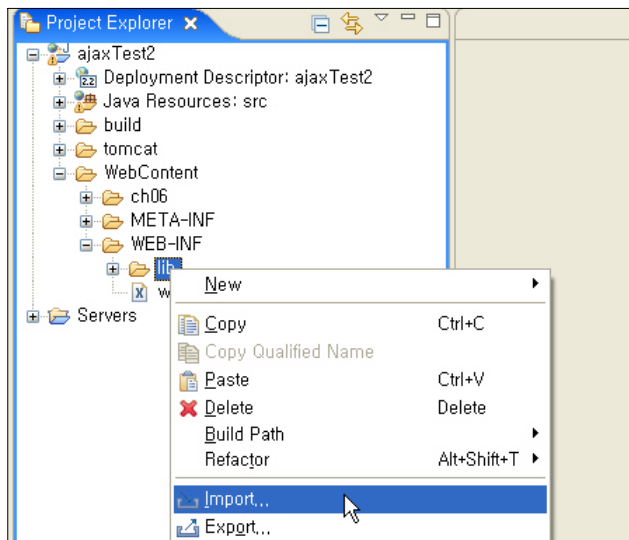
<DBCP API를 사용시 JDBC 커넥터인식 못하는 문제 해결1>

자 이제 이클립스의 프로젝트로 커넥션풀 관련 3개의 jar파일을 가져오기 한다.

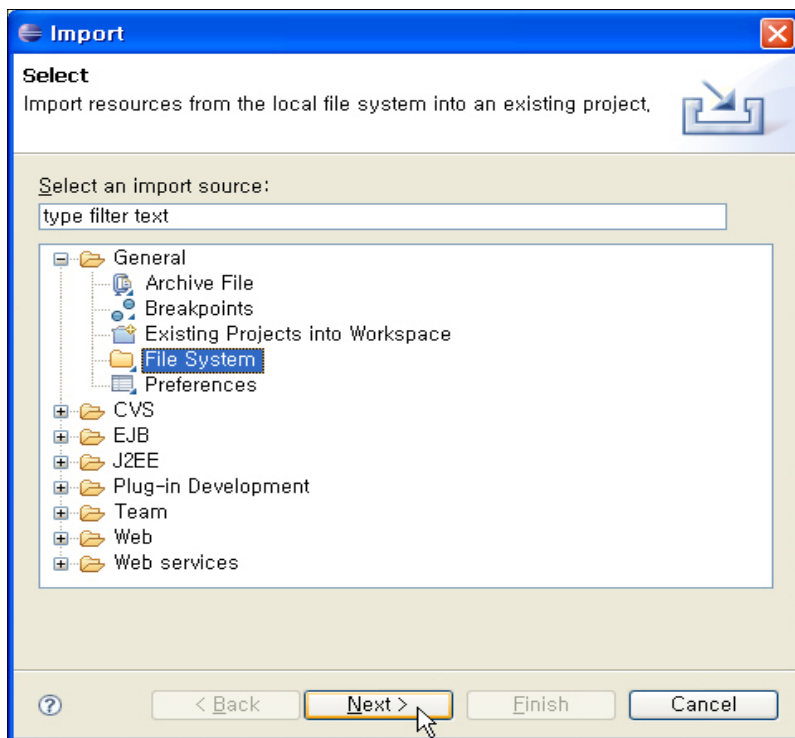
[실습] 이클립스의 프로젝트로 커넥션풀 관련 3개의 jar파일을 가져오기

① commons-dbc-1.2.2.jar, commons-collections-3.2.jar, commons-pool-1.3.jar을 [import]해보자.

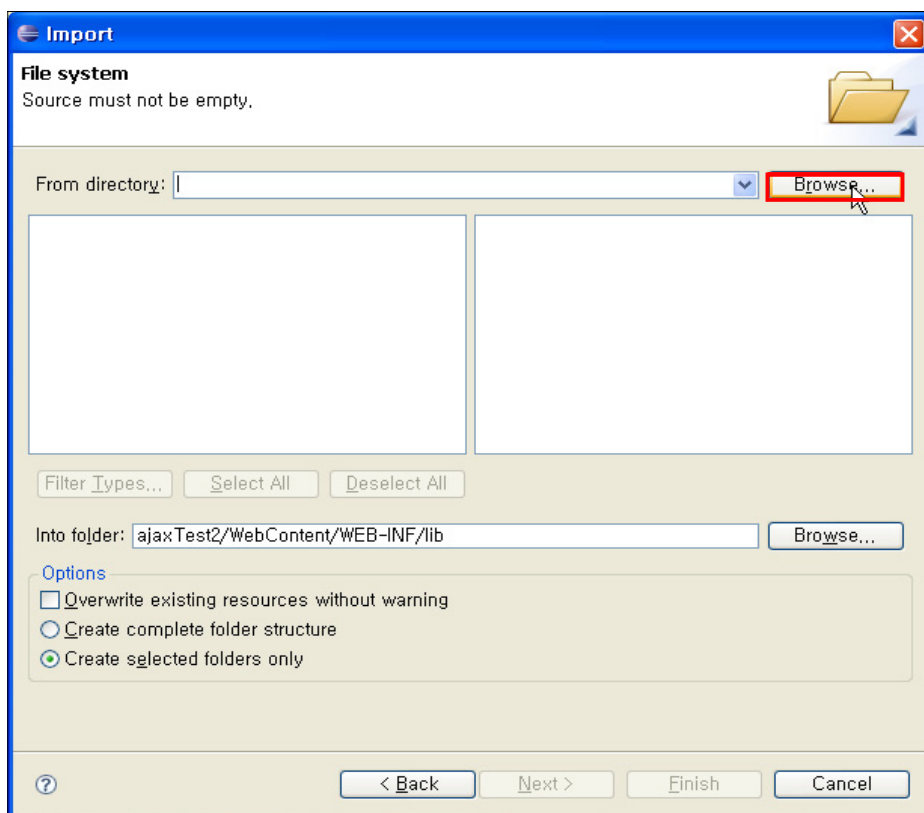
[ajaxTest2]-[WebContent]-[WEB-INF]-[lib]폴더를 선택하고, 오른쪽 마우스 단추를 클릭 해서 표시되는 단축메뉴에서 [Import]메뉴를 선택한다.



②[General]-[File System]항목을 선택하고 [Next]단추를 클릭한다.



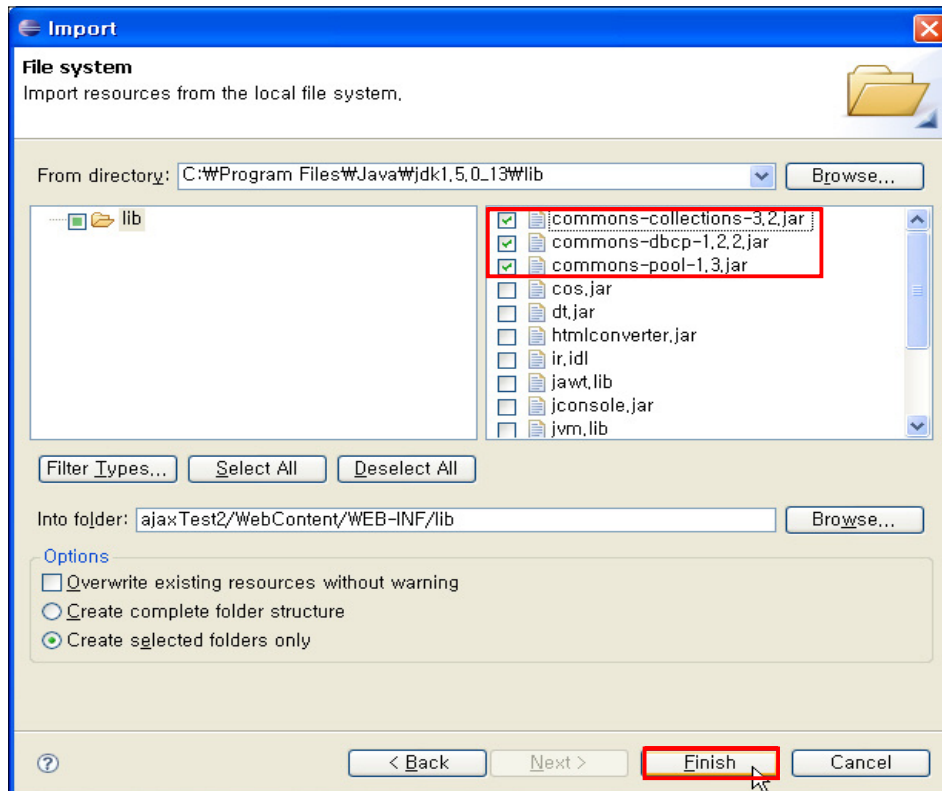
③ [Import] 대화상자가 표시되면 [Browse] 단추를 클릭한다.



④ [Import from directory] 대화상자가 표시되면 commons-dbc-1.2.2.jar,

commons-collections-3.2.jar, commons-pool-1.3.jar 파일이 있는 위치를 선택하고 [확인] 단추를 클릭한다. 필자의 경우에는 C:\WProgram Files\Java\jdk1.5.0\_13\lib폴더에 있어서 해당폴더를 선택했다.

폴더의 선택이 끝나면 [Import] 대화상자에 아래와 같은 목록이 표시되면 commons-dbcp-1.2.2.jar, commons-collections-3.2.jar, commons-pool-1.3.jar 파일을 선택하고 [Finish] 단추를 클릭한다.



## (2) DBCP에 관한 정보 설정 – server.xml

자카르타(Jakarta) DBCP를 사용하려면 DBCP에 관한 정보 설정을 server.xml에서 정의해야 한다. server.xml은 톰캣홈\conf에 있다. 이클립스를 가지고 프로젝트 작성시에는 [Package Explorer]뷰에 있는 [Servers]-[Tomcat v5.5 Server @ localhost]폴더안에 있는 server.xml파일과 톰캣홈\conf에 있는 server.xml파일을 둘다에 설정을 해놓는 것이 좋다.

나중에 웹 어플리케이션을 배포(deploy)해서 서비스 때는 톰캣홈\conf의 server.xml의 설정을 따르고, 이클립스에서 테스트할 때는 [Servers]-[Tomcat v5.5 Server @ localhost]폴더안에 있는 server.xml파일의 설정을 따르기 때문이다.

그럼 server.xml 에 DBCP에 관한 정보를 설정해 보자.

① 먼저 이클립스의 설정부터 확인하기 위해 [Servers]-[Tomcat v5.5 Server @

localhost]폴더안에 있는 server.xml파일을 더블클릭해서 편집기뷰로 불러온다. Tomcat5.5에서는 <GlobalNamingResources>엘리먼트와 <Context>엘리먼트에 모두 <Resource>엘리먼트를 정의해야 한다.

<참고>

Tomcat버전마다 다르니 버전이 Tomcat 5.0인 경우

<http://wiki.apache.org/jakarta-commons/DBCPI> 사이트를 참고.

버전이 Tomcat 6.0 인 경우

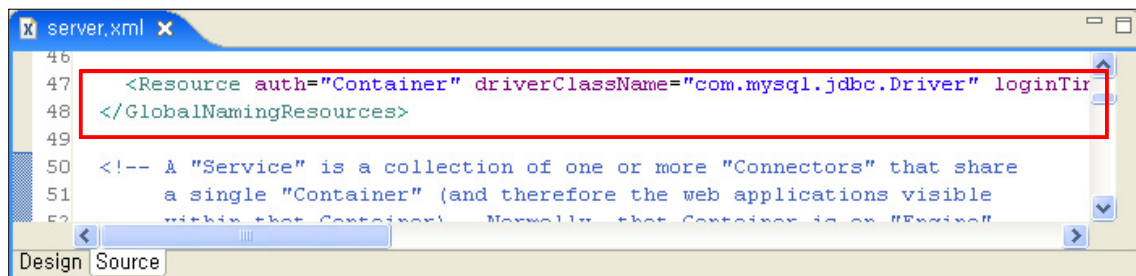
<http://tomcat.apache.org/tomcat-6.0-doc/jndi-datasource-examples-howto.html>을 참고.

② server.xml파일이 편집기뷰에 표시되면 하단의 [source]탭을 클릭해서 소스보기로 전환한다. 42~46라인쯤 </GlobalNamingResources>태그를 만나면 이 태그 바로 위에 다음과 같이 기술한다. 오타내면 안된다.

- ♦ server.xml에 추가할 <Resource>엘리먼트

```
<Resource auth="Container" driverClassName="com.mysql.jdbc.Driver"
loginTimeout="10" maxWait="5000" name="jdbc/ajaxtest" password="ajaxpass"
testOnBorrow="true" type="javax.sql.DataSource"
url="jdbc:mysql://localhost:3306/ajaxtest" username="ajaxid"/>
```

- ♦ <Resource>엘리먼트가 추가된 후 server.xml (필자의 경우는 <Resource>엘리먼트가 여러 개여서 여러분과 추가된 라인이 다를 수 있다.)



③ server.xml에서 스크롤을 내려서 370~375라인쯤에 가면 <Context>엘리먼트를 발견할 수 있다. 이것은 우리가 웹 어플리케이션을 만들면, 이클립스가 알아서 설정해준 파일이다.

- ♦ 원래의 <Context>엘리먼트

```
<Context docBase="ajaxTest2" path="/ajaxTest2" reloadable="true"
source="org.eclipse.jst.j2ee.server:ajaxTest2"/></Host>
```

이것을 다음과 같이 수정한다. 굵게 표시된 곳이 변경된 내용이다. 변경후 저장한다.

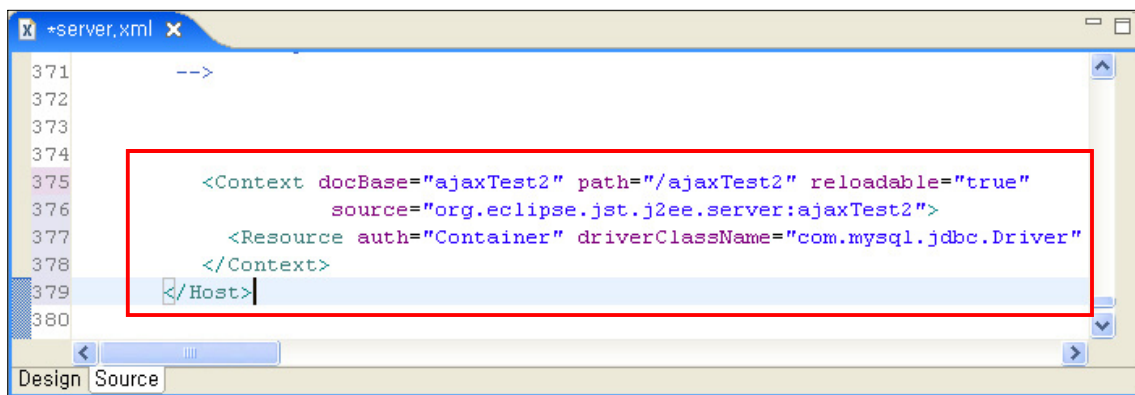
```
<Context docBase="ajaxTest2" path="/ajaxTest2" reloadable="true"
source="org.eclipse.jst.j2ee.server:ajaxTest2">
```

```

    <Resource auth="Container" driverClassName="com.mysql.jdbc.Driver"
      loginTimeout="10" maxWait="5000" name="jdbc/ajaxtest" password="ajaxpass"
      testOnBorrow="true" type="javax.sql.DataSource"
      url="jdbc:mysql://localhost:3306/ajaxtest" username="ajaxid"/>
  </Context>
</Host>

```

- ◆ <Context>엘리먼트부터 변경 후 server.xml (필자의 경우는 <Context>엘리먼트가 여러개여서 여러분과 추가된 라인이 다를 수 있다.)



<그림 7-91.jpg>

- ④ 이번에 똑같이 톰캣홈Wconf에 있는 server.xml파일의 설정을 editplus에서 불러서 변경해보자. 대략47~50라인 사이쯤에 있는 </GlobalNamingResources>태그위에 아래의 내용을 기술한다.

- ◆ server.xml에 추가할 <Resource>엘리먼트

```

<Resource      auth="Container"      driverClassName="com.mysql.jdbc.Driver"
loginTimeout="10"      maxWait="5000"      name="jdbc/jstest"      password="jspass"
testOnBorrow="true"                                     type="javax.sql.DataSource"
url="jdbc:mysql://localhost:3306/jstest" username="jsid"/>

```

- ⑤ server.xml에서 스크롤을 내려서 380~400라인쯤에 가면 </Host>엘리먼트가 표시되는 위에 다음의 내용을 기술한다. 변경후 저장한다.

```

<Context                                             path="/javaScriptTest"
docBase="d:\Wapache-tomcat-6.0.18\webapps\javaScriptTest"
crossContext="true">
<Resource      auth="Container"      driverClassName="com.mysql.jdbc.Driver"
loginTimeout="10"      maxWait="5000"      name="jdbc/jstest"      password="jspass"

```

```
testOnBorrow="true" type="javax.sql.DataSource"
url="jdbc:mysql://localhost:3306/jstest" username="jsid"/>
</Context>
```

일단 설정이 완료 되었다. 이번에는 JNDI 리소스를 사용하기 위한 web.xml을 설정해보자.

### (3) JNDI 리소스 사용 설정 - web.xml

JNDI 리소스를 사용하려면 web.xml에 아래와 같이 <resource-ref>태그를 기술해야 한다.

```
<resource-ref>
  <description>jstest db</description>
  <res-ref-name>jdbc/jstest</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

<description>은 설명을 기술하고 <res-ref-name> 태그는 server.xml의 <Resource>태그의 name속성과 매칭되고, <res-type> 태그는 <Resource>태그의 type속성과 매칭되고, <res-auth> 태그는 <Resource>태그의 auth속성과 매칭된다.

[실습] 이클립스에서 [ajaxTest2]-[Web Contents]-[WEB-INF]폴더안에 있는 web.xml 파일에 JNDI 리소스 사용 설정

① 이클립스에서 [ajaxTest2]-[Web Contents]-[WEB-INF]폴더안에 있는 web.xml파일을 더블클릭해서 편집기뷰에 표시한다.

② 편집기뷰에 표시된 web.xml파일을 [source]탭을 클릭해서 소스보기로 본 다음 </Web-app>태그의 위에 다음의 내용을 추가한 후 저장한다.

◆ web.xml파일에 추가할 내용

```
<resource-ref>
  <description>jsptest db</description>
  <res-ref-name>jdbc/ajaxtest</res-ref-name>
  <res-type>javax.sql.DataSource</res-type>
  <res-auth>Container</res-auth>
</resource-ref>
```

### 4) JSP페이지에서 커넥션 풀 사용

이제는 jsp페이지에서 DBCP API를 사용한 커넥션 풀을 사용해 보자. 서블릿과 같은 방법이니 알아둔다.

```
<%@ page contentType = "text/html; charset=euc-kr" %>
<%@ page import = "java.sql.*,javax.sql.*, javax.naming.*" %>
```



중략

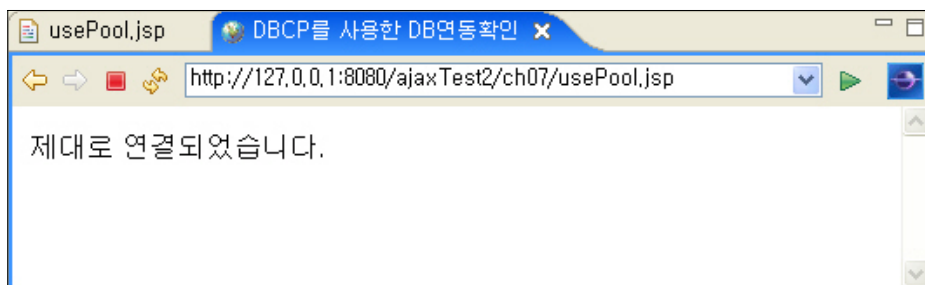
..

```
try{
    Context initCtx = new InitialContext();
    Context envCtx = (Context) initCtx.lookup("java:comp/env");
    DataSource ds = (DataSource)envCtx.lookup("jdbc/ajaxtest");
    Connection conn = ds.getConnection();
}
```

..

생략

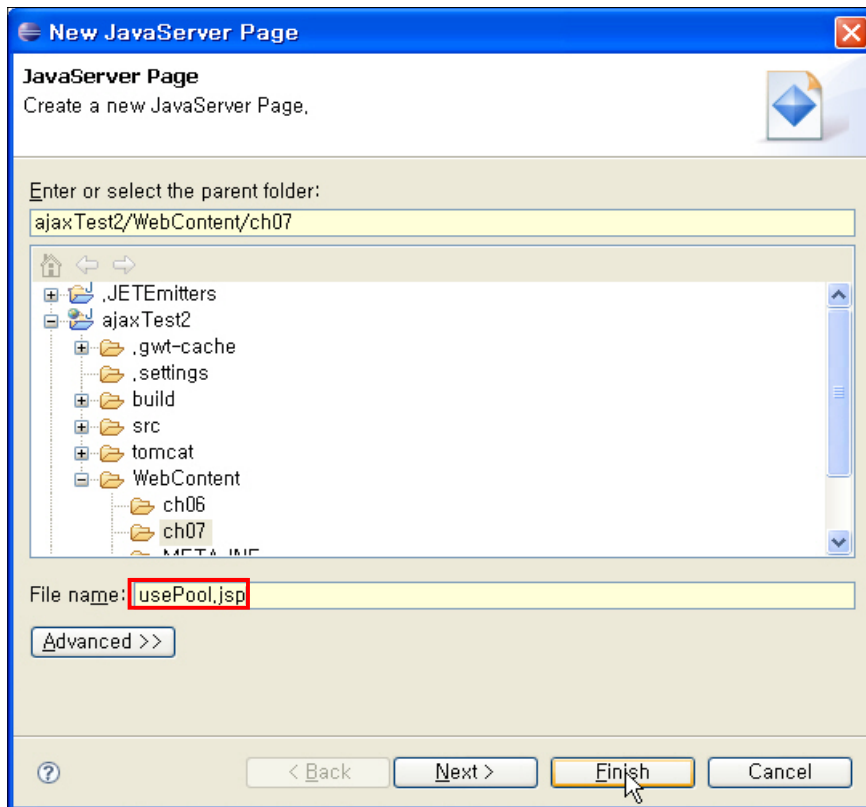
[실습] JSP페이지에서 DBCP API 커넥션 풀 사용 예제  
다음과 같은 결과가 표시되도록 작성해 보자.



<usePool.jsp페이지의 실행결과>

작성파일의 정보는 다음과 같다.

- DBCP API를 사용해서 커넥션풀을 사용하는 usePool.jsp페이지
- ① usePool.jsp페이지를 작성하기 위해서 [ch07] 폴더를 선택하고, 마우스 오른쪽 단추를 클릭해서 단축메뉴를 표시한다. 단축메뉴에서 [New]-[JSP]메뉴를 선택한다.
  - ② [새 JavaServer 페이지]창이 표시된다. [상위폴더 입력 또는 선택]항목이 'ajaxTest2/WebContent/ch07'이면 [파일이름]항목에 'usePool.jsp'를 입력하고 [완료]단추를 클릭한다.



<그림 7-95.jpg>

③ usePool.jsp페이지의 기본적인 코딩이 작성되면 다음과 같이 수정한 후 저장한다.


```

1  <%@ page language="java" contentType="text/html; charset=EUC-KR"
2      pageEncoding="EUC-KR"%>
3  <%@ page import = "java.sql.*,javax.sql.*, javax.naming.*" %>
4
5  <html>
6  <head><title>DBCP를 사용한 DB연동확인</title></head>
7  <body>
8  <%
9      Connection conn=null;
10
11     try{
12         Context initCtx = new InitialContext();
13         Context envCtx = (Context) initCtx.lookup("java:comp/env");
14         DataSource ds = (DataSource)envCtx.lookup("jdbc/ajaxtest");
15         conn = ds.getConnection();
16
17         out.println("제대로 연결되었습니다.");
18     }

```

```
19     }catch(Exception e){
20         e.printStackTrace();
21     }finally{
22         if(conn != null) try{conn.close();}catch(SQLException sqle){}
23     }
24     %>
25 </body>
26 </html>
```

④ driverTest.jsp페이지의 수정이 끝나면 [Servers]뷰에 있는 [Tomcat v5.5 Server~]를 선택하고 [Start the Server]아이콘을 클릭해서 톰캣서비스를 실행한다.

⑤ 이클립스의 툴바에서  (내장웹브라우저아이콘)을 클릭해서 내장웹브라우저가 표시되면 , 웹 브라우저의 주소에 <http://localhost:8080/ajaxTest2/ch07/usePool.jsp>을 입력한 후 실행한다.