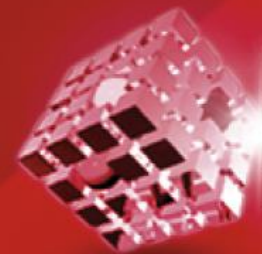


# JSP 게시판 실습

고객지원팀



**CUBRID**

More than open source!

CUBRID2008 R2.0 v. 20091010

- ④ 개요
- ④ JSP 사용 환경 설정
- ④ CUBRID 기본 사용 방법
- ④ 게시판 작성

- JSP 게시판 실습 목표
  - JAVA 환경에서 CUBRID 를 이용하는 기본 적인 방법 습득
    - 데이터베이스 연결
    - 질의 처리
  - 게시판 작성에 필요한 CUBRID 만의 질의 처리 방법 습득
    - 전체 게시물 건수 구하기
    - 게시판 페이징 방법
    - 조회 건수 처리 방법
    - 게시물 수정/ 삭제 방법

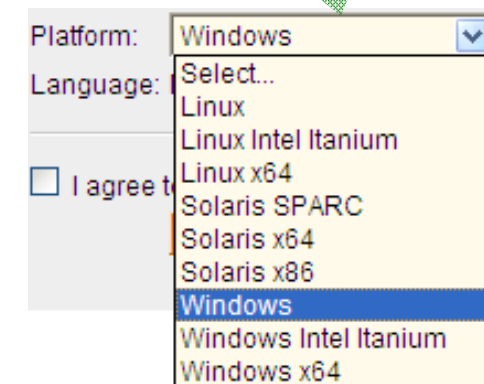
# JSP 사용 환경 설정

- JAVA다운로드 (java.sun.com → Downloads → JAVA SE)
  - <http://java.sun.com/javase/downloads/index.jsp>
  - 최신버전의 JAVA 선택 해 JDK설치)
    - JDK6 Update16 (운영환경의 경우 JRE6 Update16), 2009년 9월 현재



- 사용 Platform(OS)선택하여 다운로드
- Windows offline Installation 선택 설치

| File Description and Name                                 | Size     |
|---|----------|
| Windows Offline Installation<br>jre-6u16-windows-i586.exe | 15.89 MB |

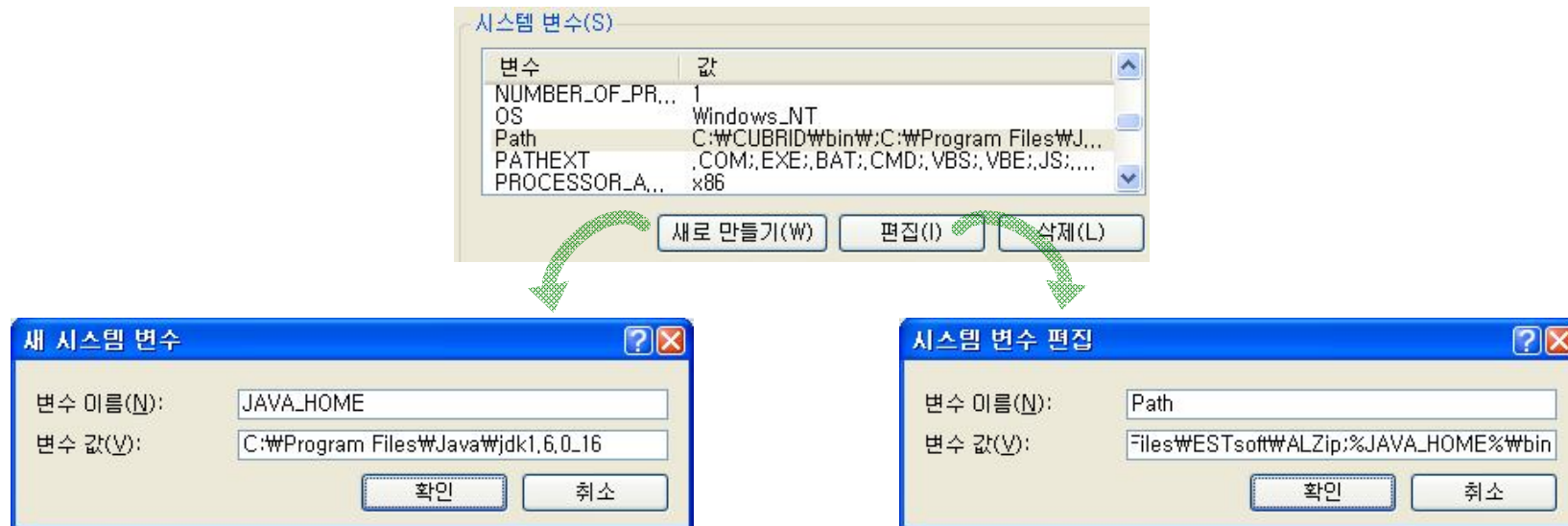


- JAVA 환경 변수 설정

- JAVA\_HOME, JAVA path, CLASSPATH 설정

- Windows 수동 설정 : 제어판 → 시스템 → 고급탭 → 환경변수 → 시스템변수

```
set JAVA_HOME= C:\Program Files\Java\jdk1.6.0_16  
set PATH=%PATH%;%JAVA_HOME%\bin  
// set CLASSPATH=%CUBRID%\jdbc\cubrid_jdbc.jar
```



- JAVA web 서버(Tomcat) 설치

- apache homepage 를 통하여 tomcat 선택 후, 다운로드 및 설치 (zip)
  - 최신버전 설치 (Tomcat 6.0... 2009/10/19 현재 최신)  
<http://mirror.korea.ac.kr/apache/tomcat/tomcat-6/v6.0.20/bin/apache-tomcat-6.0.20.zip>
  - 적정 디렉토리에 압축 풀기 (예), c:\wtomcat )
- tomcat 구동후 동작 확인
  - <tomcat>\bin\startup 수행
  - browser 에서 <http://localhost:8080> 접속후 아래 화면 확인



# CUBRID 기본 사용 방법



## ● JDBC driver tomcat 등록

- %CUBRID%/jdbc/cubrid\_jdbc.jar 를 <tomcat 설치 디렉토리>\\lib 에 복사
- tomcat 재구동: shutdown.bat, startup.bat

## ● 데이터베이스 연결

```
Class.forName("cubrid.jdbc.driver.CUBRIDDriver");  
Connection conn =  
DriverManager.getConnection( "jdbc : CUBRID : IP : PORT :DB:::" , "ID" , "PASSWORD" );
```

IP : 연결할 데이터베이스가 위치한 서버의 IP

PORT : 연결할 broker port 번호 (기본값: 33000)

DB : 연결할 데이터베이스 이름 (대소문자 구분함)

ID : 연결할 데이터베이스 사용자 ID (대소문자 구분 없음)

PASSWORD : 연결할 데이터베이스 사용자 암호, 없으면 ""

## ● 데이터베이스 연결 해제

```
conn.close();
```

## ● 예제

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

<%
Connection conn = null;
try {
    Class.forName("cubrid.jdbc.driver.CUBRIDDriver");
    conn = DriverManager.getConnection("jdbc:cubrid:localhost:33000:demodb:::", "", "");

    DatabaseMetaData meta = conn.getMetaData();

    out.println("===== <br>");
    out.println("Database Product Name is ... " + meta.getDatabaseProductName() + "<br>");
    out.println("Database Product Version is " + meta.getDatabaseProductVersion() + "<br>");
    out.println("JDBC Driver Name is ..... " + meta.getDriverName() + "<br>");
    out.println("JDBC Driver Version is ..... " + meta.getDriverVersion() + "<br>");
    out.println("JDBC Driver Major Version is " + meta.getDriverMajorVersion() + "<br>");
    out.println("JDBC Driver Minor Version is " + meta.getDriverMinorVersion() + "<br>");
    out.println("===== " + "<br>");

} catch ( SQLException e ) { System.err.println("SQLE:" + e.getMessage());
} catch ( Exception e ) { System.err.println("E:" + e.getMessage());
} finally {
    if ( conn != null ) conn.close();
}
%>
```

```
=====
Database Product Name is ... CUBRID
Database Product Version is 8.2.0.1150
JDBC Driver Name is ..... CUBRID JDBC Driver
JDBC Driver Version is ..... 8.2.0.1150
JDBC Driver Major Version is 8
JDBC Driver Minor Version is 2
=====
```

- 질의 수행하기

- 데이터 입력

```
stmt = conn.createStatement();  
stmt.executeUpdate(sql);
```

- 데이터 검색

```
stmt = conn.createStatement();  
rs = stmt.executeQuery(sql);  
while(rs.next()) {  
    line_id = rs.getString("line_id");  
    line_no = rs.getInt(2);  
}  
rs.close();  
stmt.close();
```

# 게시판 작성

- 게시판(bbs)

- 기본적인 내용만 포함

- 게시물 번호(bbs\_no) 는 게시물 생성시 1씩 증가
    - 최근 게시물을 먼저 보여줌.

- order by bbs\_no desc 처리위해 bbs\_no 역순 인덱스 생성

```
create table bbs (  
    bbs_no    int auto_increment,  
    bbs_title varchar(100),  
    bbs_content varchar(500),  
    bbs_cnt   int,  
    constraint pk_no primary key(bbs_no)  
)  
create index r_no on bbs(bbs_no desc)  
-- create reverse index r_no on bbs(bbs_no)
```

- 게시물 전체 건수
  - 인덱스를 이용한 count(\*) 처리
    - primary key 혹은 unique index 를 이용하여 count(\*) 처리
    - 결과가 달라지지 않는, 관련 인덱스에 대한 조건 추가

```
select count(*) from bbs
// 수정 질의
select count(*) from bbs where bbs_no > 0
```

- 게시물 목록 페이지 처리

- 인덱스를 이용한 order by desc 처리

- 인덱스를 이용하여 결과를 가져올 경우 인덱스 순서와 결과 순서 동일

```
select bbs_no,bbs_title,bbs_cnt from bbs order by bbs_no desc
// 수정 질의
select bbs_no,bbs_title,bbs_cnt from bbs where bbs_no > 0 using index r_no(+)
```

- 페이지에 필요한 결과만 가져오기

- rownum 을 이용하여 결과중 필요한 범위의 결과만 가져옴

```
select bbs_no,bbs_title,bbs_cnt from bbs where bbs_no > 0 order by bbs_no desc for
orderby_num() between 11 and 20
// 수정 질의
select bbs_no,bbs_title,bbs_cnt from bbs where bbs_no > 0 and rownum between 11 and 20
using index r_no(+)
```

- 주요 소스 (bbs\_list.jsp)

```
if (page_no < 1) page_no = 1;  
s_no = (page_no - 1) * 10 + 1;  
e_no = s_no + 10 - 1;
```

```
Class.forName("cubrid.jdbc.driver.CUBRIDDriver");  
conn = DriverManager.getConnection("jdbc:cubrid:localhost:33000:demodb::", "dba", "");  
  
int bbs_total = 0;  
  
String sql = "select count(*) from bbs where bbs_no > 0";  
stmt = conn.createStatement();  
rs = stmt.executeQuery(sql);  
rs.next();  
bbs_total = rs.getInt(1);  
rs.close();  
stmt.close();
```

```
sql = "select bbs_no,bbs_title,bbs_cnt from bbs where bbs_no > 0";  
sql += " and rownum between " + s_no + " and " + e_no;  
sql += " using index r_no(+);"  
stmt = conn.createStatement();  
rs = stmt.executeQuery(sql);  
while (rs.next()) {  
    bbs_no = rs.getInt(1);  
    bbs_title = rs.getString("bbs_title");  
    bbs_cnt = rs.getInt("bbs_cnt");
```



- 게시물 번호(bbs\_no) 자동 입력
- 주요 소스 (bbs\_insert.jsp)

```
String sql = "insert into bbs(bbs_title,bbs_content,bbs_cnt) values(?,?,0)";  
stmt = conn.prepareStatement(sql);  
stmt.setString(1, bbs_title);  
stmt.setString(2, bbs_content);  
stmt.executeUpdate();  
stmt.close();
```

- 게시물 조회 건수(bbs\_cnt) 자동 증가
  - click counter(incr()) 를 이용하여 자동 증가
- 주요 소스 (bbs\_view.jsp)

```
String sql = "select bbs_title, bbs_content, incr(bbs_cnt) bbs_cnt from bbs where bbs_no = " + bbs_no;  
stmt = conn.createStatement();  
rs = stmt.executeQuery(sql);  
rs.next();  
String bbs_title = rs.getString(1);  
String bbs_content = rs.getString(2);  
Integer bbs_cnt = rs.getInt(3);  
rs.close();  
stmt.close();  
conn.close();
```

교육에 참가해 주셔서 감사합니다.  
양질의 교육을 위하여 여러분의 의견을 부탁드립니다.

[www.cubrid.com/customer\\_survey.php](http://www.cubrid.com/customer_survey.php)