

Tech Info(vol.9)

Informix JDBC와 Java Servlet을 이용한 Web Programming 시작하기

준비사항

제품	버전	사이트
Informix JDBC	2.0 이상	www.intraware.com
JDK	1.1.6 이상(권장 1.2.04)	Java.sun.com
JSDK	Only 2.0	Java.sun.com
Gcc, CC	Unix os에 맞는 버전	
Apache	1.3.9 이상	www.apache.org
ApacheJServ	1.1	http://java.apache.org

JDK 설치

제품을 다운로드받아서 디렉토리를 만든 후 제품을 풀고, 환경변수에 추가해 줍니다.

ksh 일 경우

```
export JAVA_HOME=/usr/java1.2

export PATH=$JAVA_HOME/bin:$PATH:
```

JDBC 설치

1. Informix 구입 제품이 있다면 JDBC 디렉토리 안에 있는 setup.class 를 root 로 로그인 한 후,

```
$) java setup
```

을 실행하고 가지고 있는 제품의 Serial Number 와 Key 를 사용하여 설치합니다. (반드시 X window 에서 실행하여야 함)

제품을 다운로드 받아 사용할 수도 있으며, <http://www.intraware.com> 이나 <http://itech.ako.net> 의 자료실을 방문하면 됩니다.

ksh 일 경우

```
export JDBC_HOME=/usr/Informix/JDBC
```

를 실행하여 INFORMIXDIR, INFORMIXSERVER 의 환경변수와 위의 환경변수들이 제대로 지정되어 있는지 확인하십시오.

```
#itech.ako.net 는 hostname, 2000 은 Informix tcp/ip daemon port
```

3. Informix JDBC online manual 은 \$JDBCHOME/doc 에 있는 html 문서입니다.

JSDK 2.0 설치

설치 하고자 하는 디렉토리를 만들고 제품을 풀고 다음의 환경변수를 설정합니다.

ksh 일 경우

```
export JSDKHOME=/usr/JSDK

export CLASSPATH=$JSDKHOME/lib/jsdk.jar:$CLASSPATH
```

Apache 서버 설치

1) 제품을 다운로드 받아 제품을 풀고 디렉토리 안에서 다음과 같이 shell 을 작성합니다.

```
#vi con.sh

./configure W

--prefix=/usr/local/apache W
```

```
sh con.sh -> make -> make install
```

3. Apache 서버를 실행합니다.

```
/usr/local/apache/bin 에서
```

```
$) apachectl start
```

를 실행하여 정상적으로 구동하는지 확인합니다.

JServ 설치

1. 제품을 다운로드 받아 제품을 풀고 디렉토리 안에서 다음과 같이 shell 을 작성합니다.

```
#vi con.sh

/configure W

--prefix=/usr/local/JServ W

--with-apxs=/usr/local/apache/bin/apxs W

--enable-module=so W

--with-jdk-home=/usr/java1.2/ W

--with-JSDK=$JSDKHOME/lib/jsdk.jar W

--disable-debugging
```

2. \$) sh con.sh

-> make -> make install

이를 실행할 때는 위의 2, 3, 4 번의 환경이 설정되어 있어야 합니다.

3. 인스톨이 성공적으로 끝난 경우라면 `usr/local/apache/conf/jserv` 라는 디렉토리와 `/usr/local/apache/libexec/mod_jserv.so` 가 생성됩니다.

4. Configuration

`httpd.conf` 파일의 마지막 line 에 다음의 내용을 추가하십시오.

```
include /usr/local/apache/conf/jserv/jserv.conf
```

```
wrapper.classpath=/i.Biz/dhlee/apache/libexec/ApacheJServ.jar
```

```
wrapper.classpath=/i.Biz/dhlee/JSDK20/lib/jsdk.jar:/CS1/JDBC210/
```

```
lib/ifxjdbc.jar:/usr/java1.2/src.jar
```

- /usr/local/apache/conf/jserv/zone.properties 파일 편집 : 기본적으로 제공되는 파일을 사용해도 무방합니다.

5. 테스트

Servlet Test

Apache Server 재시동하십시오.

```
$) /usr/local/apache/bin/apachectl restart
```

http://serverIP/servlets/IsItWorking 브라우저에서 테스트 하면 됩니다.

JDBC 테스트는 다음과 같이 진행하십시오.

JDBC Test Program

아래의 예제를 만들어 컴파일하고 http://serverIP/servlets/DbconnectTest 로 접속하여 데이터를 제대로 가져오는지 테스트하십시오. 컴파일한 데이터는 \$ApacheHome/servlets 아래에 있어야 합니다.

```
#File Name : DbconnectTest.java
```

```
import java.io.*;
```

```
import java.sql.*;
```

```
import java.util.*;
```

```
import javax.servlet.*;
```

```
import javax.servlet.http.*;
```

```
public class DbConnectTest extends HttpServlet
```

```
{
```

```
    public void init(ServletConfig config) throws ServletException
```

```
    {
```

```
        super.init(config);
```

```
    }
```

```
    public void doPost(HttpServletRequest req,
```

```
        HttpServletResponse res)
```

```
throws ServletException, IOException {
```

```
    doGet(req, res);
```

```
}
```

```
public void doGet(HttpServletRequest req,
```

```
    HttpServletResponse res)
```

```
    throws ServletException, IOException
```

```
{
```

```
    res.setContentType("text/html; charset=euc-kr");
```

```
    PrintWriter out = res.getWriter();
```

```
    main(req, res, out);
```

```
}
```

```
public void main(HttpServletRequest req,
```

```
    HttpServletResponse res, PrintWriter out)
```

```
    throws ServletException, IOException
```



```
{
```

```
Connection conn = null;
```

```
    ResultSet result = null;
```

```
    PreparedStatement pstmt = null;
```

```
int retVal =0;
```

```
try {
```

```
    conn = newConnection();
```

```
    String SelQry ="select fname,lname,phone,address1 from
```

```
        customer";
```

```
    pstmt = conn.prepareStatement(SelQry);
```

```
    result = pstmt.executeQuery();
```

```
    String[ ] items = null; Vector list1 = new Vector();
```

```
while(result.next()) {

    items = new String[4];

    items[0] = getCode(result.getString(1));

    items[1] = getCode(result.getString(2));

    items[2] = getCode(result.getString(3));

    items[3] = getCode(result.getString(4));

    list1.addElement(items);

}

out.println("<body bgcolor=#ffffff");

out.println("<h3>Connecting Test..</h3>");

out.println("<table border=1 cellspacing=1 cellpadding=0

    width=600>");

out.println("<tr>");

out.println("<th>fname</th>");
```

```
out.println("<th>lname</th>");
```

```
out.println("<th>phone</th>");
```

```
out.println("<th>address</th>");
```

```
out.println("</tr>");
```

```
for (int i=0;i<list1.size();i++) {
```

```
    items=(String[]list1.elementAt(i);
```

```
        out.println("<tr>");
```

```
            out.println("<td>"+items[0]+"</td>");
```

```
            out.println("<td>"+items[1]+"</td>");
```

```
            out.println("<td>"+items[2]+"</td>");
```

```
            out.println("<td>"+items[3]+"</td>");
```

```
        out.println("</tr>");
```

```
    }

    out.println("</table>");

out.println("</body>\n</html>");

pstmt.close();

    conn.close();

} catch (Exception e) {

    e.printStackTrace(out);

} finally {

    try {

        if(pstmt !=null)

            pstmt.close();

    } catch (SQLException e) {}

}

}

public Connection newConnection()
```

```
throws SQLException,Exception

{

String DBstr="jdbc:Informix sqli://kor-olive:1567/stores7:

Informixserver=ius920;user=Informix;password=aaaa";

//kor-olive 는 system name, 1567 Informix tcp port, stores7

: dbname

String ifxDriver = "com.Informix.jdbc.IfxDriver";

Class.forName(ifxDriver);

Connection DBCon = DriverManager.getConnection(DBstr);

return DBCon;

}

/*****
```

데이터를 데이터베이스에 저장하기 위한 값을 데이터베이스의

환경설정에 맞도록 고쳐주기 위한 함수

```
*****/
```

```
public static String setCode(String query)
```

```
{
```

```
    if(query == null) return null;
```

```
    try{
```

```
        return new String(query.getBytes("KSC5601"), "8859_1");
```

```
    }catch (Exception e){return query;}
```

```
}
```

```
/******
```

데이터베이스에서 가져온 데이터를 처리

```
*****/
```

```
public static String getCode(String query)
```

```
{
```

```
    if(query == null) return null;
```

```
    try{
```

```
        return new String(query.getBytes("8859_1"), "KSC5601");
```

```
    }catch (Exception e){return query;}  
  
}  
  
}
```

오류에 관하여

위의 1~6번을 순차적으로 처리하여 이상이 없었다면 7번 sample도 반드시 실행 되어야 합니다. 하지만 아무런 오류를 발생하지 않고 단번에 설치되어 테스트를 완료하는 것이 그리 쉽지는 않을 것입니다. 오류가 발생하였을 때는 PATH나 기타 환경변수 등이 제대로 설정되어 있는지 다시 한 번 살펴보고, 눈에 잘 띄지 않는 오타가 있을 수도 있으므로 각각의 단계를 꼼꼼하게 체크해야 합니다.

한 가지 주의할 것은 Jserv 는 default daemon port 가 8007 입니다. 따라서 8007 을 쓰는 다른 daemon 이 있는지 살펴보고 테스트하시길 바랍니다.

이 밖의 내용은 <http://itech.ako.net> 에 접속하여 Q&A 나 자료실을 방문하여 보시길 바랍니다.