리눅스에서 자바 서블릿과 JSP 사용하기, 제2판

황 성준 donquabi@netian.com

Id: jserv_jsp.html,v 2.0 2000/10/07

목 차

제 1 절 이 문서에 대하여

같은 제목의 문서를 이미 공개했지만, 그 이후 관련 소프트웨어가 업그레이드 되었고, 변경된 사항도 있다. 이를 또한 알리고, 이들을 설치한 경험을 함께 나누고자 한다.

이 문서는 필자의 경험을 토대로, 리눅스 상에서 아파치 웹 서버와 자바 서블릿과 GNUJSP를 사용하기 위해서 필요한 소프트웨어를 설치하고 조율하 는 과정을 기술한다.

이 문서를 다른 곳에 옮길 경우에 반드시 필자에게 메일을 주시고, 내용을 변경하지 마시고 전문 그대로를 인용하여야 합니다.

이 문서는 필자의 홈페이지에서도 볼 수 있다: `<http://my.netian.com/~donquabi/linux/jserv/jserv_jsp-2.html>`

제 2 절 운용 환경

우선, 필자의 리눅스 운용 환경은 다음과 같다.

- Pentium Pro 200MHz (RAM: 64MB)
- 와우 리눅스 6.2 & RedHat Linux 7.0
- 커널 2.2.17
- 웹 서버: apache-1.3.12

필자는 알짜 리눅스 6.1을 사용하다가 와우 리눅스 6.2를 설치하였고, 최근에는 RedHat Linux 7.0을 설치하여 사용 중이다. 이런 연유로 이 문서에 기술하는 내용을 이들 두 가지 리눅스 배포본에서 시험할 수 있었다. 와우 리눅스와 같은 래드햇 리눅스 6.2를 기반으로 하는 배포판과 RedHat Linux 7.0에서도 큰 문제없이 적용 가능할 것이다. (혹시 필자에게는 없었던 문제가 여러분한테서 생길 수도 있을 것이다. 어떤 것들이 문제가 될지 필자는 예상치 못하겠다. 만일 예상치 못한 문제가 발생한다고 하더라도 필자를 탓하지 말기를 당부한다.)

제 3 절 준비 사항

우선, 여기서는 레드햇 리눅스 6.2 이상과, 함께 배포하는 아파치 웹 서버 1.3.9 이상을 설치하였다고 가정하겠다.

아파치 서버에서 자바 서블릿을 사용하기 위해서 다음과 같은 소프트웨어 를 준비하여야 한다.

- *JDK 1.1.8: ibm-jdk-l118-linux-x86.tgz* `<http://www.ibm.com/java/jdk/118/linux/index.html>`
- *IBM Java 2 SDK 1.3: IBMJava2-SDK-13.tgz* `<http://www6.software.ibm.com/dl/dklx130/dklx130-p>`
- *JSDK 2.0: jsdk20-solaris2-sparc.tar.Z* `<http://java.sun.com/products/servlet/download.html>`
- *Apache JServ 1.1.2: ApacheJServ-1.1.2-1.i386.rpm* `<http://java.apache.org/jserv/dist/ApacheJServ-1.1.2-1.i386.rpm>`
- *GNUJSP 1.0.1: gnujsp-1.0.1-1.i386.rpm* `<gnujsp-1.0.1-1.i386.rpm>`

자바 실행 환경은 JDK 1.1.8 또는 Java 2 SDK, v1.3 중에서 선택한다.

제 4 절 JDK 설치

이전 문서에서 필자는 IBM 사에서 제공하는 리눅스용 JDK 1.1.8을 선택했었다. GNUJSP 설치 문서에서 JDK 1.1 을 사용하여 개발하였음을 명시하고 있기 때문에 이것이 가장 적합하다고 판단했다. 물론, 그 당시에는 리눅스용 Java 2 Platform이 없었다. 올해 7월 초에 IBM이 Sun 보다 먼저 Java 2 SDK를 내놓았다. 이번 주에는 Sun도 *Java 2 SDK, Standard Edition Version 1.3.0 for Linux (Intel x86)* `<http://java.sun.com/j2se/1.3/download-linux.html>`를 발표했다.

4.1 JDK 1.1.8

설치 디렉토리를 `/usr/local`로 정하고, 압축 파일을 푼다. 오류가 없다면, 현재 디렉토리에 jdk118 디렉토리가 생긴다. 결과를 확인한다.

설치 과정을 다음에 보인다.

```
$ su
Password:
# cd /usr/local
# tar zxvf ibm-jdk-l118-linux-x86.tgz
# ls
#
```

그리고, 필자는 `/usr/local/java`도 `/usr/local/jdk118`과 동일하게 되도록 링크를 만들었다:ln -s /usr/local/jdk118 /usr/local/java. 만일 자바 디렉토리가 다른 곳에 위치한다면 필자처럼 링크를 만들어 해결할 수 있다.

4.2 IBM Java 2 SDK 1.3

설치 과정을 다음에 보인다.

```
$ su
Password:
# cd /usr/local
# tar zxvf IBMJava2-SDK-13.tgz
# ls
#
```

이 과정에서 오류가 없었다면, 압축 파일을 푼 다음, 현재 디렉토리에 IBMJava2-13 디렉토리가 생긴다. 필요하면, 링크를 만든다: ln -s /usr/local/IBMJava2-13 /usr/local/java2

제 5 절 JSDK 선택

ApacheJServ 서블릿 엔진은 서블릿 API 2.0 규격에 따라 구현되었다. 따라서 아파치에서 사용하는 서블릿은 서블릿 2.0 규격을 지켜야 하고, 또한 이를 지원하는 JSDK를 사용하여 개발해야 한다.

JSDK는 서블릿을 컴파일 하고, 시험하는 데 필수적인 개발 도구이다. JSDK는 리눅스를 위한 버전이 별도로 제공되지 않는다. 서블릿 API 규격을 만든 선(Sun) 사에서 제공하는 솔라리스(Solaris)의 것을 리눅스에서도 사용 가능하다.

또한, Paul Siegmann이 제작하여 LGPL로 공개한 무료 JSDK를 사용할 수도 있다. 서블릿 API 2.0 규격대로 구현되어 선 사의 것과 완전하게 호환성을 갖는다. ApacheJServ 패키지에 이를 포함시켰고, 우선적으로 이것을 사용 토록 설정하였다.

제 6 절 ApacheJServ RPM

ApacheJServ는 1.1 버전부터 소스 파일과 함께 RPM도 배포된다. 이제는 굳이 소스를 컴파일 하여 실행 파일을 만들고, 함께 배포되는 샘플을 참고하여 실행에 필요한 구성 파일들을 직접 구성하여 만들어야 하는 과정이 필요하지 않다. 이 과정은 INSTALL 파일에 상세하게 설명되어 있지만, 초보자에게는 여전히 어렵고, 또한 여간 번거로운 게 아니다.

6.1 패키지 설치

아파치와 JDK가 설치되어 있다면, 다음과 같이 ApacheJServ 패키지를 설치 한다.

```
# rpm -i ApacheJServ-1.1.2-1.i386.rpm
```

설치 과정에 오류가 없다면, 지정된 디렉토리에 파일이 복사되고, `/etc/httpd/conf/httpd.conf` 파일에 ApacheJServ를 위한 구성 내용을 추가 한다.

패키지 설치로 인해 생성되는 디렉토리와 추가되는 파일은 다음의 표와 같다.

```
+-----------------------------+-------------------+------------------+
|           디렉토리          |      파   일      |     설   명      |
+-----------------------------+-------------------+------------------+
| /etc/httpd/conf/jserv/      | jserv.conf        |                  |
|                             | jserv.properties  | 구성 파일        |
|                             | zone.properties   |                  |
+-----------------------------+-------------------+------------------+
| /etc/httpd/modules/         | ApacheJServ.jar   | 서블릿 엔진을    |
|                             |                   |구현한 클래스 파일|
+-----------------------------+-------------------+------------------+
| /home/httpd/classes/        | servlet-2.0.jar   | JSDK             |
+-----------------------------+-------------------+------------------+
| /home/httpd/servlets/       | Hello.class       | 서블릿 예제      |
|                             | IsItWorking.class |                  |
+-----------------------------+-------------------+------------------+
| /usr/doc/ApacheJServ-1.1.2/ |                   |문서 파일들       |
+-----------------------------+-------------------+------------------+
| /var/log/httpd/             | jserv.log         |로그 파일         |
+-----------------------------+-------------------+------------------+
```

제 7 절 GNUJSP RPM

GNUJSP는 선 사의 자바 서버 페이지(Java Server Pages)를 구현한 것이다. GNUJSP 서블릿을 설치하면, .jsp 확장자를 가지는 파일을자바 소스 파일로 번역하여 컴파일 하고, 실행하게 된다. GNUJSP 1.0은 JSP 1.0 규격을 지원 한다.

GNUJSP는 자바 소스와 클래스 파일 형태로, 무료로 배포된다. 아파치를 포함하여 광범위한 플랫폼, 웹 서버, 서블릿 엔진을 지원한다. 아파치 서버에 설치하는 경우는 함께 배포되는 INSTALL.apache `<http://klomp.org/gnujsp/INSTALL.apache>`문서를 참고한다.

GNUJSP도, ApacheJServ 패키지와 마찬가지로, 간편한 설치와 구성을 위하여 RPM 패키지로 만들었다.

다음과 같이 GNUJSP 패키지를 설치한다.

```
# rpm -i gnujsp-1.0.1-1.i386.rpm
```

설치 과정에 오류가 없다면, `/usr/lib/gnujsp` 디렉토리를 생성하고, 클래스 파일이 복사된다. ApacheJServ 구성 파일에 GNUJSP를 위한 구성 내용을 추가하고, GNUJSP 서블릿 존 구성 파일을 추가한다.

패키지 설치로 인해 생성되는 디렉토리와 추가되는 파일은 다음의 표와 같다.

디렉토리	파 일	설명	비고
/etc/httpd/conf/jserv/	jserv.conf	ApJServMount, ApJServAction 추가	수정
/etc/httpd/conf/jserv/	jserv.properties	classpath 추가	수정
/etc/httpd/conf/jserv/	gnujsp.properties	GNUJSP용 서블릿 존 구성 파일	생성
/usr/lib/gnujsp/	gnujsp10.jar sax2.jar openxml-1.2-w3c.jar aelfred.jar	JSP를 구현한 클래스 파일	생성
/usr/doc/gnujsp-1.0.1/		문서 파일과 JSP 예제들	생성

제 8 절 구성 파일

ApacheJServ 사용하려면 다음과 같은 구성 파일들이 필요하다.

- *jserv.conf* <http://my.netian.com/~donquabi/linux/jserv/jserv_conf.txt>
- *jserv.properties* <http://my.netian.com/~donquabi/linux/jserv/jserv_properties.txt>
- *서블릿 존[servlet zone] 구성 파일* <http://my.netian.com/~donquabi/linux/jserv/zone.properties.txt>
- *GNUJSP용 존[servlet zone] 구성 파일* <http://my.netian.com/~donquabi/linux/jserv/wsjzone_properties.txt>

8.1 jserv.conf 구성 파일

jserv.conf 구성 파일은 아파치 웹 서버에 ApacheJServ 서블릿 엔진을 결합하기 위한 설정 사항을 추가한다. GNUJSP 패키지가 추가하는 부분은 다음과 같다.

```
ApJServProperties /etc/httpd/conf/jserv/jserv.properties
    :
# Mount point for Servlet zones
ApJServMount /wsj /wsjzone
ApJServMount /servlets /root
ApJServMount /servlet /root
    :
# Executes a servlet passing filename with proper extension in PATH_TRANSLATED
```

```
| # property of servlet request.                                             |
| ApJServAction .jsp /wsj/gnujsp                                             |
|     :                                                                      |
+----------------------------------------------------------------------------+
```

8.2 jserv.properties 구성 파일

jserv.properties 구성 파일은 ApacheJServ 서블릿 엔진을 위한 실행 환경 을 설정한다.

눈여겨 보아야 할 부분은 다음과 같다. JDK 1.1.8을 사용하는 경우이다.

```
+-------------------------------------------------------------------+
| # Execution parameters                                            |
| wrapper.bin=/usr/local/java/bin/java                              |
|     :                                                             |
| # CLASSPATH environment value passed to the JVM                   |
| wrapper.classpath=/usr/local/java/lib/classes.zip                 |
| wrapper.classpath=/etc/httpd/modules/ApacheJServ.jar              |
| wrapper.classpath=/home/httpd/classes/servlet-2.0.jar             |
| wrapper.classpath=/usr/lib/gnujsp/servlet-2.0-plus.jar            |
|     :                                                             |
| # Servlet Zones parameters                                        |
| zones=root,wsjzone                                                |
| root.properties=/etc/httpd/conf/jserv/zone.properties             |
| wsjzone.properties=/etc/httpd/conf/jserv/wsjzone.properties       |
|     :                                                             |
+-------------------------------------------------------------------+
```

또한, Java 2 SDK 1.3을 사용하는 경우에는 다음과 같다.

```
+-------------------------------------------------------------------+
| # Execution parameters                                            |
| wrapper.bin=/usr/local/java2/bin/java                             |
|     :                                                             |
| # CLASSPATH environment value passed to the JVM                   |
| wrapper.classpath=/usr/local/java2/lib/tools.zip                  |
| wrapper.classpath=/etc/httpd/modules/ApacheJServ.jar              |
| wrapper.classpath=/home/httpd/classes/servlet-2.0.jar             |
| wrapper.classpath=/usr/lib/gnujsp/servlet-2.0-plus.jar            |
|     :                                                             |
| # Servlet Zones parameters                                        |
| zones=root,wsjzone                                                |
| root.properties=/etc/httpd/conf/jserv/zone.properties             |
| wsjzone.properties=/etc/httpd/conf/jserv/wsjzone.properties       |
|     :                                                             |
+-------------------------------------------------------------------+
```

8.3 wsjzone.properties 구성 파일

wsjzone.properties 구성 파일은 JSP를 위한 실행 환경을 설정한다. ApacheJServ와 함께 배포되는 서블릿 존 구성파일인 zone.properties을 토대로 만들어진다.

주요 부분은 다음과 같다.

+——————————————————————————+ | # List of Repositories | | ######################## | | | | # The list of servlet repositories controlled by this servlet zone | | # Syntax: repositories=[repository],[repository]... | | # Default: NONE | | # Note: The classes you want to be reloaded upon modification should be put | | # here. | | repositories=/usr/lib/gnujsp/gnujsp10.jar | | repositories=/usr/lib/gnujsp/sax2.jar | | repositories=/usr/lib/gnujsp/openxml-1.2-w3c.jar | | repositories=/usr/lib/gnujsp/aelfred.jar | | #repositories=/home/httpd/servlets | | : | | | | # Servlet Aliases | | ################### | | | | # This defines aliases from which servlets can be invoked. | | # Each alias give a new instance of the servlet. This means that if a servlet | | # is invoked both by class name and by alias name, it will result in _TWO_ | | # instances of the servlet being created. | | # Syntax: servlet.[alias].code=[classname] (String) | | # Default: NONE | | # servlet.snoop.code=SnoopServlet | | # servlet.hello.code=org.fool.Dummy | | servlet.gnujsp.code=org.gjt.jsp.JspServlet | | : | | | | # Aliased Servlet Init Parameters | | #################################### | | | | # These properties define init parameters for each servlet that is invoked | | # by its alias. | | # Syntax: servlet.[alias].initArgs=[name]=[value],[name]=[value],... | | # Default: NONE | | # servlet.snoop.initArgs=message=I'm a snoop servlet | | # servlet.hello.initArgs=message=I say hello world to everyone | | | | # give me debug output in the logs (time and space consuming) | | servlet.gnujsp.initArgs=debug=true | | # I want detailed path output in the logs (lot of stuff) | | # servlet.gnujsp.initArgs=pathdebug=true | | # where to put created java source and class files (www server needs write access!) | servlet.gnujsp.initArgs=scratchdir=/var/tmp/gnujsp | | # created java classes use packages instead of long file names | | servlet.gnujsp.initArgs=usepackages=true | | # remove comment if you want german messages :-) | | # servlet.gnujsp.initArgs=language=de,country=de | | # checkclass: is set to true: check for changes to jsp pages and recompile | | # checkdependancies: keep track of included files and check for changes | | servlet.gnujsp.initArgs=checkdependancies=true,checkclass=true | | # Do not remove java source files after compilation. (helpful for debugging) | | servlet.gnujsp.initArgs=keepJava=true | | # which sax parser to use for xml files (experimental!) | | # servlet.gnujsp.initArgs=xmlparser=com.microstar.xml.SAXDriver | | # builtin-java: use the java compiler classes from JDK | | servlet.gnujsp.initArgs=compiler=builtin-javac -classpath %classpath%:%scratch| |dir%:/usr/lib/gnujsp/gnujsp10.jar -d %scratchdir% -deprecation %source% | +——————————————————————————+

제 9 절 서블릿 시험

ApacheJServ 패키지는 아파치를 위한 구성 내용을 httpd.conf 에 추가하기 때문에 서버가 실행 중인 경우에 서버를 재기동하여야 한다.

9.1 서버 기동

먼저, 웹 서버를 기동해야 한다. 만일, 현재 웹 서버가 실행 중이라면 재기동을 해야 한다.

```
# /etc/rc.d/init.d/httpd stop
# /etc/rc.d/init.d/httpd start
```

9.2 서버 상태 보기

아파치 서버의 error_log 파일을 열어 파일의 끝 부분을 들여다 보자: tail /var/log/httpd/error_log. 다음은 필자가 RedHat Linux 7.0에서 error_log 파일을 열어 본 것으로, 이 중에서 ApacheJServ/1.1.2가 있다면, 아파치 서버 데몬과 서블릿 엔진이 정상적으로 동작 중임을 알 수 있다.

```
+-------------------------------------------------------------------+
```

```
|  ......                                                       |
|  ... [notice] Apache/1.3.12 (Unix) (Red Hat/Linux) mod_ssl/2.6.6 |
|OpenSSL/0.9.5a mod_perl/1.24 PHP/4.0.2 ApacheJServ/1.1.2 configure|
|d -- resuming normal operations                                |
+---------------------------------------------------------------+
```

9.3 서블릿 실행

ApacheJServ 패키지를 설치하면, /home/httpd/servlets 디렉토리에 Hello, IsItWorking 서블릿 파일들이 생성된다. 웹 브라우저를 띄워 이들 중에서 IsItWorking 서블릿을 실행해 보자.

IsItWorking 서블릿을 실행하기 위한 URL은 다음과 같다.

http://localhost/servlet/IsItWorking

[[*IsItWorking 서블릿 실행 결과: 웹 브라우저 화면* <http://my.netian.com/~donquabi/linux/jserv/working.jpg>]]

9.4 JSP 실행

서블릿 실행이 성공적이면 JSP 파일을 시험한다. 만일 오류가 있다면, JSP 파일에 대해서도 오류를 만날 수 밖에 없다.

GNUJSP 패키지를 설치하면, JSP 예제도 함께 설치된다. 위치는 /usr/share/doc/gnujsp-1.0.1/examples 이다. 이 중 hello.jsp 파일(설치 동안에 /home/httpd/html 디렉토리에 복사된다)을 /home/httpd/html 디렉토리에 복사하고, 웹 브라우저를 통해 접근해 본다.

hello.jsp 파일에 접근하기 위한 URL은 다음과 같다.

http://localhost/hello.jsp

[[*hello.jsp 웹 브라우저 화면* <http://my.netian.com/~donquabi/linux/jserv/hellojsp-2.jpg>]]

제 10 절 RedHat Linux 6.2와 7.0의 차이

웹 문서는 레드햇 리눅스 6.2에서는 /home/httpd/html 디렉고리에, RedHat 7.0 에서는/var/www/html 디렉토리에 위치한다. 이 문서에서 언급한 ApacheJServ RPM 및 GNUJSP RPM은 레드햇 리눅스 6.2에 기준하여 제작되었다. 따라서 hello.jsp 파일은 /home/httpd/html 디렉토리에 복사된다.

RedHat 7.0 을 사용하는 경우에는 /home/httpd/html 디렉토리 내의 hello.jsp 파일을 /var/www/html 디렉토리로 직접 복사하여야 한다.

제 11 절 에필로그

지금까지의 과정에서 별다른 문제가 없었다면, 서블릿을 실행하여 정상적 인 결과를 보았을 것이다. 또한 JSP도 정상적으로 작동할 것이다. 작은 감동이 느껴지지 않는가?

ApacheJServ 1.1과 GNUJSP 1.0은 초기 서블릿 규격과 JSP 규격을 따르는 데서 시작됐고, 계속된 개발 과정을 통해 현재는 서블릿 규격 2.0과 JSP 규격 1.0을 충족한다. 최근에 이들 규격은 모두 업그레이드 되었다. 서블릿 규격은 최종 2.2 버전까지 확정되었고, 2.3 버전이 준비되어 공개 검토 중에 있다. 그리고, JSP 규격은 1.1 버전까지 확정되었고, 1.2 버전이 공개 검토 중에 있다.

또한, GNUJSP를 대체하는, 새로운 공개 JSP를 구현하는 프로젝트가 아파치 (Apache Software Foundation)의 주도로 진행 중에 있는데 *Tomcat* <http://jakarta.apache.org/tomcat>이 그것이다. Tomcat은

서블릿 엔진 2.2와 JSP 1.1 기술을 결합하는 참조 구현 (reference implementation)이다. 현재 안정된 버전은 3.1로서 홈페이지에서 바이너리 파일 뿐만 아니라 소스 파일을 내려 받을 수 있다.

제 12 절 참고 사항

- *리눅스에서 자바 서블릿과 JSP 사용하기* `<http://my.netian.com/~donquabi/linux/jserv/jserv_jsp.html>`
- *The Apache JServ Project* `<http://java.apache.org/>`
- *GNUJSP* `<http://klomp.org/gnujsp/>`
- *Java Server Pages* `<http://java.sun.com/products/servlet/>`
- *Java Servlet API* `<http://java.sun.com/products/servlet/>`
- *Alternative Servlet implementation* `<http://www.euronet.nl/~pauls/java/servlet/>`
- *Servlet Central: Using Apache JServ 1.0b1* `<http://www.servletcentral.com/1999-01/jserv.html>`
- *Servlet FAQ* `<http://www.purpletech.com/servlet-faq/>`
- *Java 1.1.8 IBM Developer Kit for Linux* `<http://www.ibm.com/java/jdk/118/linux/index.html>`
- *IBM Developer Kit for Linux, Java 2 Technology Edition* `<http://www.ibm.com/java/jdk/linux130/>`
- *Java 2 SDK, Standard Edition Version 1.3.0 for Linux (Intel x86)* `<http://java.sun.com/j2se/1.3/download-linux.html>`
- *Tomcat @ Jakarta* `<http://jakarta.apache.org/tomcat/>`
- *Which JSP book serves up the best lesson?* `<http://www.javaworld.com/javaworld/jw-09-2000/jw-0929-jspbooks.html/>`
- *Java servlet books: A comparative review* `<http://www.javaworld.com/javaworld/jw-03-2000/jw-03-ssj-books.html>`

위 내용 중 잘못된 부분을 지적해 주거나 조언을 주시면 감사하겠습니다.

1999-2000 황 성준 sj@kldp.org donquabi@netian.com