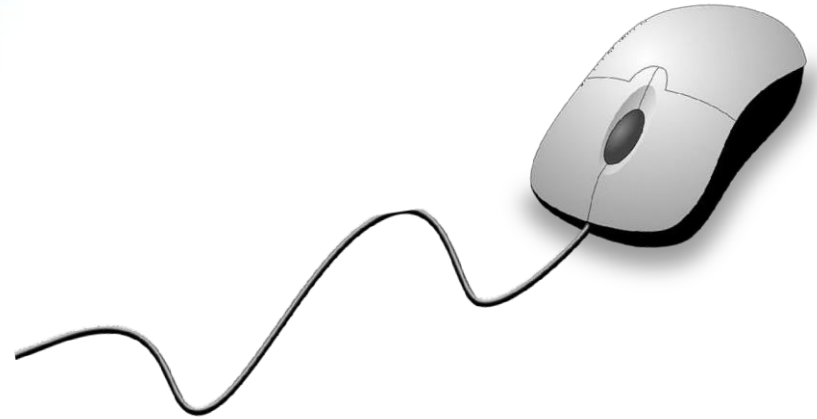


공개SW 솔루션 설치 & 활용 가이드

미들웨어 > 웹서버



jetty://

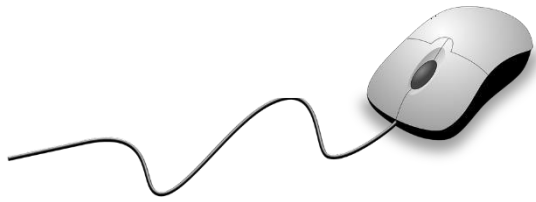
제대로 배워보자

How to Use Open Source Software

Open Source Software Installation & Application Guide



오픈소스 소프트웨어 통합지원센터
Open Source Software Support Center



CONTENTS

1. 개요
2. 기능요약
3. 실행환경
4. 설치 및 실행
5. 기능소개
6. 활용예제
7. FAQ
8. 용어정리

1. 개요

jetty://



소개	<ul style="list-style-type: none"> Jetty Web Server는 독립형 또는 내장된 인스턴스를 통해 정적/동적 콘텐츠를 제공할 수 있는 HTTP 서버와 서블릿 컨테이너 제공 		
주요기능	<ul style="list-style-type: none"> 비동기 HTTP 서버 표준 기반 서블릿 컨테이너 웹 소켓 서버 http / 2 서버 OSGI, JNDI, JMX, JASPI, AJP 지원 		
대분류	<ul style="list-style-type: none"> 미들웨어 	소분류	<ul style="list-style-type: none"> 웹서버
라이선스형태	<ul style="list-style-type: none"> Apache License 2.0, Eclipse Public License 1.6 	사전설치 솔루션	<ul style="list-style-type: none"> JDK1.8 이상
운영제제	<ul style="list-style-type: none"> Windows, Linux, Mac, Unix 등 	버전	<ul style="list-style-type: none"> 9.4.6
특징	<ul style="list-style-type: none"> 유연하고 확장 가능 비 동기식 엔터프라이즈 확장 가능 오픈 소스 및 상업적으로 사용 가능 		
보안취약점	<ul style="list-style-type: none"> 취약점 ID : CVE-2015-2080 심각도 : 7.5 HIGH(V3) 취약점 설명 : 예외 처리 코드 인 9.2.9.v20150224를 사용하면 원격 공격자가 HTTP 헤더(JetLeak)의 잘못된 문자를 통해 프로세스 메모리에서 중요한 정보를 얻을 수 있음 대응방안 : 9.2.9 이상 업데이트 참고 경로 : https://lists.fedoraproject.org/pipermail/package-announce/2015-March/151804.html 		
개발회사/커뮤니티	<ul style="list-style-type: none"> Eclipse 		
공식 홈페이지	<ul style="list-style-type: none"> http://www.eclipse.org/jetty/index.html 		



2. 기능요약

jetty://



Jetty는 JAVA EE 사양 중 주로 Servlet 사양을 구현하고 있다. Jetty 자체가 모든 Web Profile 기술을 제공하지 않지만 Jetty 아키텍처는 타사 기능을 플러그인하여 정확한 요구에 맞게 사용자 정의 된 컨테이너를 생성 할 수 있도록 한다.

Version	JVM	Protocols	Servlet/JSP
9.4	1.8	HTTP/1.1 (RFC 7230), HTTP/2 (RFC 7540), WebSocket (RFC 6455, JSR 356), FastCGI	3.1/2.3
9.3	1.8	HTTP/1.1 (RFC 7230), HTTP/2 (RFC 7540), WebSocket (RFC 6455, JSR 356), FastCGI	3.1/2.3
9.2	1.7	HTTP/1.1 RFC2616, javax.websocket, SPDY v3	3.1/2.3
8	1.6	HTTP/1.1 RFC2616, WebSocket RFC 6455, SPDY v3	3.0/2.2
7	1.5	HTTP/1.1 RFC2616, WebSocket RFC 6455, SPDY v3	2.5/2.1



2. 기능요약

jetty://



Java EE 7 Web Profile

JSR	Web Profile명	Jetty-9.1.x 포함여부	플러그인 여부
JSR 340	Servlet Specification API 3.1	Yes	
JSR 344	Java Server Faces 2.2 (JSF)	No	Yes, Mojarra or MyFaces
JSR 245 / JSR 341	Java Server Pages 2.3/Java Expression Language 3.0 (JSP/EL)	Yes	Yes
JSR 330	Dependency Injection for Java 1.0	No	Yes as part of a CDI implementation, Weld
JSR 907	Java Transaction API 1.2 (JTA)	Yes	Yes
JSR 356	Java API for Websocket 1.0	Yes	No



3. 실행환경

jetty://



Jetty 는 Java로 구현된 HTTP 서버 및 Servlet 컨테이너로 Eclipse Foundation의 무료 및 공개 SW 프로젝트이다.

Jetty는 독립적으로 서버를 구성할 수 있고, Java 응용 프로그램에 포함되어 웹 서비스를 제공할 수 있는 API를 지원한다.

본 가이드에서는 아래와 같은 환경으로 구성되어 있다.

- **JAVA 8**
- **Jetty 9.4.6.v20170531**



4. 설치 및 실행

jetty://



세부 목차

1. 다운로드
2. 설치
3. 디렉토리 구성
4. 실행
5. 데모 실행



4. 설치 및 실행

jetty://



4.1 다운받기(1/2)

- 먼저 JDK(Java Development Kit)가 설치되어 있어야 한다.
- 미 설치 시는 <http://www.java.com/ko/download/> 에서 다운받아 설치한다.

The screenshot shows the Java website's download page. At the top, there is a red header with the Java logo and a search bar. Below the header, there are navigation links for '다운로드' (Download) and '도움말' (Help). The main content area is titled '모든 운영 체제용 Java 다운로드' (Download Java for all operating systems) and highlights '권장 사항 Version 8 Update 131' (Recommended Version 8 Update 131). It provides information about the release date (April 18, 2017) and instructions on how to download the software. A sidebar on the left contains links for '사용 가능한 운영 체제' (Supported operating systems), '도움말 리소스' (Help resources), 'Java 7', and 'JDK'. The main content area also features a section for 'Windows' with a table of download links for different versions and file sizes. A note at the bottom explains that both 32-bit and 64-bit Java are required for older browsers.

OS	Download Link	File Size	Notes
Windows	Windows 온라인	721 KB	지침
Windows	Windows 오프라인	64.83 MB	지침
Windows	Windows 오프라인 (64비트)	62.62 MB	지침



4. 설치 및 실행

jetty://



4.1 다운받기(2/2)

- Jetty서버는 <https://www.eclipse.org/jetty/download.html> 에서 다운로드 받는다.
- 파일은 zip과 gzip 포맷 형식으로 다운로드 받을 수 있다. 설치 할 서버에 적당한 파일 형식으로 다운로드 받는다.

HOME / PROJECTS / JETTY

Jetty

- » About Jetty
- » Jetty Powered
- » Licenses

Resources

- » Downloads
- » Documentation
- » API Documentation
- » Maven Plugin
- » Mailing Lists
- » Blogs
- » Eclipse Tooling
- » Tools

Project Management

- » Community
- » Contributing
- » IP Log
- » Source

Professional Services

- » Training and Consulting

jetty://

Jetty Downloads

The latest release of Jetty is below, earlier minor release versions are available in **Maven Central**.

Release				
9.4.6.v20170531	.zip	.tgz	apidocs	source

Note: Release notes can be found in the `VERSION.txt` file included with the distribution and on the **GitHub Releases** page for the Jetty project. Release notes are also made available on the **jetty-announce@eclipse.org mailing list**.

Prior Versions

The most recent versions of prior Jetty releases can be found **here, with their associated documentation**.

Note: The canonical repository for Jetty is Maven Central. All releases are always available there first and this download page may lag a bit update wise as post release resources are put into place. You can always browse for Jetty releases **here**.

QUICK LINKS

- Project Summary
- Download
- Current Documentation
- Current API Documentation
- Enter Bug
- Reported Bugs

ACTIVE CONTRIBUTORS

w:// webtide



4. 설치 및 실행

jetty://



4.2 설치(1/3)

- 다운로드 받은 파일을 압축 해제한다.

```
[jetty ~]$ ls
jetty-distribution-9.4.6.v20170531.tar.gz
[jetty ~]$ tar xvfz jetty-distribution-9.4.6.v20170531.tar.gz
jetty-distribution-9.4.6.v20170531/
jetty-distribution-9.4.6.v20170531/bin/
jetty-distribution-9.4.6.v20170531/demo-base/
jetty-distribution-9.4.6.v20170531/demo-base/etc/
jetty-distribution-9.4.6.v20170531/demo-base/lib/
jetty-distribution-9.4.6.v20170531/demo-base/lib/ext/
jetty-distribution-9.4.6.v20170531/demo-base/resources/
jetty-distribution-9.4.6.v20170531/demo-base/start.d/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/doc/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/doc/9.4.6.v20170531/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/doc/9.4.6.v20170531/css/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/doc/9.4.6.v20170531/css/font-awesome/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/doc/9.4.6.v20170531/css/highlighter/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/doc/9.4.6.v20170531/fonts/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/doc/9.4.6.v20170531/images/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/doc/9.4.6.v20170531/js/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/ROOT/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/ROOT/images/
jetty-distribution-9.4.6.v20170531/demo-base/webapps/test.d/
jetty-distribution-9.4.6.v20170531/etc/
jetty-distribution-9.4.6.v20170531/etc/sessions/
jetty-distribution-9.4.6.v20170531/etc/sessions/file/
jetty-distribution-9.4.6.v20170531/etc/sessions/gcloud/
```



4. 설치 및 실행

jetty://



4.2 설치(2/3)

- 압축을 해제하면 아래와 같은 디렉토리가 보인다.

```
[jetty ~/jetty-distribution-9.4.6.v20170531]$ ls -al
total 692
drwxr-xr-x 10 jetty jetty    239 Jun  1 08:50 .
drwx----- 6 jetty jetty    235 Jul 21 09:45 ..
drwxr-xr-x  2 jetty jetty     22 Jul 21 09:45 bin
drwxr-xr-x  7 jetty jetty     75 Jun  1 08:48 demo-base
drwxr-xr-x  3 jetty jetty   4096 Jul 21 09:45 etc
drwxr-xr-x 13 jetty jetty   4096 Jul 21 09:45 lib
-rw-r--r--  1 jetty jetty  30012 Jun  1 08:51 license-eplv10-aslv20.html
drwxr-xr-x  2 jetty jetty     26 Jul 21 09:45 logs
drwxr-xr-x 19 jetty jetty   4096 Jul 21 09:45 modules
-rw-r--r--  1 jetty jetty   6262 Jun  1 08:51 notice.html
-rw-r--r--  1 jetty jetty   1637 Jun  1 08:50 README.TXT
drwxr-xr-x  2 jetty jetty     6 Jun  1 08:50 resources
-rw-r--r--  1 jetty jetty   5702 Jun  1 08:50 start.ini
-rw-r--r--  1 jetty jetty 147725 Jun  1 08:50 start.jar
-rw-r--r--  1 jetty jetty 483631 Jun  1 08:50 VERSION.txt
drwxr-xr-x  2 jetty jetty     24 Jul 21 09:45 webapps
[jetty ~/jetty-distribution-9.4.6.v20170531]$ █
```



4. 설치 및 실행

jetty://



4.2 설치(3/3)

- .bash_profile에 JETTY_HOME을 등록한다.

```
$ vi ~/.bash_profile
```

```
export JETTY_HOME=/home/jetty/jetty-distribution-9.4.6.v20170531
```

```
$ source ~/.bash_profile
```



4. 설치 및 실행

jetty://



4.3 디렉토리 구성

- 주요 디렉토리 구성은 아래와 같다.

Location	Description
license-eplv10-aslv20.html	라이선스 파일
VERSION.txt	Release 정보
etc/	Jetty XML 설정 파일
lib/	실행에 필요한 jar 파일
modules/	모듈 정의파일
logs/	로그 정보
wbapps/	기본 web 소스 디렉토리
start.ini	실행 옵션을 포함하고 있는 파일
start.jar	Jetty 실행 Jar



4. 설치 및 실행

jetty://



4.4 실행(1/2)

- Jetty를 기본 8080 port를 시작하기 위해서는 아래와 같은 명령을 실행한다.

```
[jetty ~]$ cd $JETTY_HOME
[jetty ~/jetty-distribution-9.4.6.v20170531]$ java -jar start.jar
2017-07-21 09:55:57.308:INFO::main: Logging initialized @940ms to org.eclipse.jetty.util.log.StdErrLog
2017-07-21 09:55:57.626:WARN:oejs.HomeBaseWarning:main: This instance of Jetty is not running from a
  separate {jetty.base} directory, this is not recommended. See documentation at http://www.eclipse.
  org/jetty/documentation/current/startup.html
2017-07-21 09:55:57.680:INFO:oejs.Server:main: jetty-9.4.6.v20170531
2017-07-21 09:55:57.702:INFO:oejdp.ScanningAppProvider:main: Deployment monitor [file:///home/jetty/
  jetty-distribution-9.4.6.v20170531/webapps/] at interval 1
2017-07-21 09:55:57.744:INFO:oejs.AbstractConnector:main: Started ServerConnector@5622fdf{HTTP/1.1,[
  http/1.1]}{0.0.0.0:8080}
2017-07-21 09:55:57.744:INFO:oejs.Server:main: Started @1377ms
■
```



4. 설치 및 실행

jetty://



4.4 실행(2/2)

- Jetty를 실행한 서버로 브라우저를 통해 서비스 호출을 실행해 본다. 하지만 \$JETTY_HOME 디렉토리에는 deploy된 webapps가 없기 때문에 404 Error가 보이게 된다.



Error 404 - Not Found.

No context on this server matched or handled this request.
Contexts known to this server are:

 [Powered by Jetty:// 9.4.6.v20170531](#)



4. 설치 및 실행

jetty://



4.5 데모 실행(1/2)

- demo-base 디렉토리에 jetty demo 예제를 실행해 본다.

```
[jetty ~]$ cd $JETTY_HOME/demo-base
[jetty ~/jetty-distribution-9.4.6.v20170531/demo-base]$ java -jar $JETTY_HOME/start.jar
2017-07-21 10:20:30.518:INFO::main: Logging initialized @525ms to org.eclipse.jetty.util.log.StdErrLog
2017-07-21 10:20:30.824:WARN::main: demo test-realm is deployed. DO NOT USE IN PRODUCTION!
2017-07-21 10:20:30.827:INFO:oejs.Server:main: jetty-9.4.6.v20170531
2017-07-21 10:20:30.845:INFO:oejdp.ScanningAppProvider:main: Deployment monitor [file:///home/jetty/jetty-distribution-9.4.6.v20170531/demo-base/webapps/] at interval 1
2017-07-21 10:20:31.062:INFO:oeja.AnnotationConfiguration:main: Scanning elapsed time=71ms
2017-07-21 10:20:31.240:INFO:oejs.session:main: DefaultSessionIdManager workerName=node0
2017-07-21 10:20:31.240:INFO:oejs.session:main: No SessionScavenger set, using defaults
2017-07-21 10:20:31.241:INFO:oejs.session:main: Scavenging every 60000ms
2017-07-21 10:20:31.274:INFO:oejsh.ContextHandler:main: Started o.e.j.w.WebAppContext@29b5cd00{/doc, file:///home/jetty/jetty-distribution-9.4.6.v20170531/demo-base/webapps/doc/,AVAILABLE}{/doc}
2017-07-21 10:20:31.364:INFO:oeja.AnnotationConfiguration:main: Scanning elapsed time=78ms
2017-07-21 10:20:31.407:INFO:oejsh.ContextHandler:main: Started o.e.j.w.WebAppContext@3bb9a3ff{/,file:///home/jetty/jetty-distribution-9.4.6.v20170531/demo-base/webapps/ROOT/,AVAILABLE}{/ROOT}
2017-07-21 10:20:31.521:INFO:oeja.AnnotationConfiguration:main: Scanning elapsed time=29ms
2017-07-21 10:20:31.524:WARN::main: test-jndi webapp is deployed. DO NOT USE IN PRODUCTION!
2017-07-21 10:20:31.585:INFO:oejsh.ContextHandler:main: Started o.e.j.w.WebAppContext@4550bb58{/test-jndi,file:///tmp/jetty-0.0.0.0-8080-test-jndi.war-_test-jndi-any-3309956636625725982.dir/webapp/,AVAILABLE}{/test-jndi.war}
2017-07-21 10:20:31.829:INFO:oeja.AnnotationConfiguration:main: Scanning elapsed time=174ms
2017-07-21 10:20:31.832:WARN::main: test webapp is deployed. DO NOT USE IN PRODUCTION!
2017-07-21 10:20:31.918:INFO:oejsh.ManagedAttributeListener:main: update PushFilter null->org.eclipse.jetty.servlets.PushCacheFilter@2f54a33d on o.e.j.w.WebAppContext@6ed3ccb2{/test,file:///tmp/jetty-0.0.0.0-8080-test.war-_test-any-8117041013959739955.dir/webapp/,STARTING){/test.war}
```



4. 설치 및 실행

jetty://



4.5 데모 실행(2/2)

- Jetty를 실행한 서버로 접속하여 브라우저를 통해 서비스 호출을 실행해 본다.



Welcome to Jetty 9

The Jetty project is a 100% Java [Servlet](#) Container which supports asynchronous server and client implementations of the [HTTP](#), [Websocket](#) and [HTTP/2](#) protocols. The project is 100% [Open Source](#) and hosted by the [Eclipse Foundation](#) at <http://www.eclipse.org/jetty>.

examples ...

- [Test Jetty Webapp](#)
- [Async Rest](#)
- [JAAS Test](#)
- [JNDI Test](#)
- [Servlet 3.1 Test](#)
- [Redirected Context](#)

information ...

- [Jetty Homepage](#)
- [Jetty Documentation \(local\)](#)
- [Javadoc \(via transparent proxy\)](#)
- [Xref \(via transparent proxy\)](#)
- [Jetty Powered](#)

getting help ...

- [Mailing lists @ eclipse](#)
- [Developer Advice](#)
- [Custom Development](#)
- [Production support](#)



5. 기능소개

세부 목차

jetty://



1. HTTP 포트 변경
2. SSL 설정
3. HTTPS 포트 변경
4. 웹 애플리케이션 디플로이
5. Virtual Hosts 설정
6. 로깅



5. 기능소개

jetty://



5.1 HTTP 포트 변경(1/2)

- jetty.http.port property 설정을 통해 포트를 변경할 수 있다.

```
[jetty ~/jetty-distribution-9.4.6.v20170531]$ java -jar start.jar jetty.http.port=8081
2017-07-24 11:30:07.720:INFO::main: Logging initialized @470ms to org.eclipse.jetty.util.log.StdErrLog 2
017-07-24 11:30:07.886:WARN:oejs.HomeBaseWarning:main: This instance of Jetty is not running from a
separate {jetty.base} directory, this is not recommended. See documentation at http://www.eclipse.org/jetty/documentation/current/startup.html
2017-07-24 11:30:07.918:INFO:oejs.Server:main: jetty-9.4.6.v20170531
2017-07-24 11:30:07.938:INFO:oejdp.ScanningAppProvider:main: Deployment monitor
[file:///home/jetty/jetty-distribution-9.4.6.v20170531/webapps/] at interval 1
2017-07-24 11:30:07.955:INFO:oejs.AbstractConnector:main: Started
ServerConnector@1794d431{HTTP/1.1,[http/1.1]}{0.0.0.0:8081} 201
7-07-24 11:30:07.956:INFO:oejs.Server:main: Started @706ms
```



5. 기능소개

jetty://



5.1 HTTP 포트 변경(2/2)

- 영구적으로 포트를 변경해야 하는 경우에는 start.ini 파일을 수정한다.

```
## Connector host/address to bind to  
# jetty.http.host=0.0.0.0
```

```
## Connector port to listen on  
jetty.http.port=8081
```

```
## Connector idle timeout in milliseconds  
# jetty.http.idleTimeout=30000
```

```
## Connector socket linger time in seconds (-1 to disable)  
# jetty.http.soLingerTime=-1
```

```
## Number of acceptors (-1 picks default based on number of cores)  
# jetty.http.acceptors=-1
```



5. 기능소개



5.2 SSL 설정(1/6)

- OpenSSL을 사용하여 SSL 인증서를 만든다. OpenSSL 명령어가 없는 경우에는 OpenSSL 웹 사이트 (<https://www.openssl.org>) 에서 설치한다. 아래의 내용은 예제이므로 실제 사이트에서 사용하는 인증서는 발급 인증기관의 문서를 참고해야 한다.

1. 다음 명령을 실행한다.

```
[jetty ~]$ openssl genrsa -des3 -out jetty.key
Generating RSA private key, 1024 bit long modulus
.....++++++
.....++++++
e is 65537 (0x10001)
Enter pass phrase for jetty.key: jettypass (4자 이상의 원하는 암호로 설정)
Verifying - Enter pass phrase for jetty.key: 위와 동일한 패스워드를 입력한다.
```



5. 기능소개

jetty://



5.2 SSL 설정(2/6)

2. 이제 인증서 파일을 생성할 것이다. 파일의 이름을 jetty.crt로 지정한다.

```
[jetty ~]$ openssl req -new -x509 -key jetty.key -out jetty.crt
```

```
Enter pass phrase for jetty.key:
```

You are about to be asked to enter information that will be incorporated into your certificate request.

What you are about to enter is what is called a Distinguished Name or a DN.

There are quite a few fields but you can leave some blank

For some fields there will be a default value,

If you enter '.', the field will be left blank.

Country Name (2 letter code) [XX]:KR (국가 코드 2자리)

State or Province Name (full name) []: (국가 또는 지방 이름)

Locality Name (eg, city) [Default City]: (도시 이름)

Organization Name (eg, company) [Default Company Ltd]: (조직 이름)

Organizational Unit Name (eg, section) []: (조직 단위 이름)

Common Name (eg, your name or your server's hostname) []:JETTY (Domain 이름)

Email Address []: (이메일 주소)



5. 기능소개

jetty://



5.2 SSL 설정(3/6)

3. crt 파일을 PKCS12 형식으로 변환해야 한다. 이 작업은 다음 단계를 통해 수행 할 수 있다.

```
[jetty ~]$ openssl pkcs12 -inkey jetty.key -in jetty.crt -export -out jetty.pkcs12
Enter pass phrase for jetty.key: jettypass
Enter Export Password: jettypass(jettypass 을 다시 사용 했지만 자유롭게 변경할 수 있다.)
Verifying - Enter Export Password: jettypass
```

4. Jetty keystore에서 PKCS12 파일 가져오는 단계는 다음과 같다.

- Jetty.pkcs12 파일을 JETTY_HOME/etc 디렉토리에 복사
- 기존 키 저장소 파일을 삭제
- Keytool import 명령을 실행

```
[jetty ~]$ cp jetty.pkcs12 $JETTY_HOME/etc
[jetty ~]$ cd $JETTY_HOME/etc
[jetty ~/jetty-distribution-9.4.6.v20170531/etc]$ keytool -importkeystore -srckeystore jetty.pkcs12 -
srcstoretype PKCS12 -destkeystore keystore
Enter destination keystore password: (keystore이 암호를 입력 - jettypass)
Re-enter new password:
Enter source keystore password: jettypass (이전에 pkcs12파일에 대해 정의한 암호)
Entry for alias 1 successfully imported.
Import command completed: 1 entries successfully imported, 0 entries failed or cancelled
```



5. 기능소개

jetty://



5.2 SSL 설정(4/6)

5. 지금까지 인증서를 생성하여 Jetty의 키 스토어로 가져 오는 과정을 살펴보았다. 이제 Jetty 서버에 HTTPS 커넥터와 SSL을 설정할 것이다. Jetty는 모듈러 아키텍처를 사용하므로 구성 파일을 통해 다른 모듈을 활성화 할 수 있어야 한다. 따라서 SSL 및 HTTPS 모듈을 수동으로 활성화해야 한다.

- \$JETTY_HOME/start.ini 파일을 열어 다음 행을 추가한다.

```
--module=ssl  
--module=https
```

- 마지막 단계로 이전 섹션에서 설정한 키 저장소 비밀번호를 정의하는 것이 필요하다. 이전 예제에서 keystore와 인증서 모두에 대해 "jettpass" 암호를 정의하였는데 이 암호를 다음 단계를 통해 암호화 할 수 있다.

```
[jetty ~]$ cd $JETTY_HOME/lib  
[jetty ~/jetty-distribution-9.4.6.v20170531/lib]$ java -cp jetty-util-9.4.6.v20170531.jar  
org.eclipse.jetty.util.security.Password jettpass (lib 버전은 Jetty 버전에 따라 다름)  
2017-07-24 15:38:39.770:INFO::main: Logging initialized @138ms to org.eclipse.jetty.util.log.StdErrLog  
jettpass  
OBF:1wty1vu91v9u1y8320zj1y7v1v8s1vv11wug  
MD5:157c95e4fb782c3d1d14355a7295ab00
```



5. 기능소개

jetty://



2. SSL 설정(5/6)

- 생성된 암호를 OBF로 시작하는 줄을 복사한다.
(이 예제에서는 **OBF:1wty1vu91v9u1y8320zj1y7v1v8s1vv11wug**)
- 이제 SSL 구성에서 이 비밀번호를 설정한다.

```
[jetty ~/jetty-distribution-9.4.6.v20170531/lib]$ cd $JETTY_HOME/etc
```

```
[jetty ~/jetty-distribution-9.4.6.v20170531/modules]$ vi jetty-ssl-context.xml
```

기존 예제 내용중 KeyStorePassword와 KeyManagerPassword 의 패스워드를 변경한다.

```
<Set name="KeyStorePassword"><Property name="jetty.sslContext.keyStorePassword"
deprecated="jetty.keystore.password" default="OBF:1wty1vu91v9u1y8320zj1y7v1v8s
1vv11wug"/></Set>
```

```
<Set name="KeyManagerPassword"><Property name="jetty.sslContext.keyManagerPassword"
deprecated="jetty.keymanager.password" default="OBF:1wty1vu91v9u1y8320zj1y7v1v8s1vv1
1wug"/></Set>
```



5. 기능소개

2. SSL 설정(6/6)

jetty://



- 이제 Jetty 서버를 기동한다. 출력 로그는 다음과 유사하다. [j

```
etty ~/jetty-distribution-9.4.6.v20170531]$ java -jar start.jar
2017-07-24 16:21:48.698:INFO::main: Logging initialized @475ms to org.eclipse.jetty.util.log.StdErrLog
2017-07-24 16:21:48.864:WARN:oejs.HomeBaseWarning:main: This instance of Jetty is not running from
a separate {jetty.base} directory, this is not recommended. See documentation at http://www.eclipse.org/jetty/documentation/current/startup.html
2017-07-24 16:21:48.926:INFO:oejs.Server:main: jetty-9.4.6.v20170531
2017-07-24 16:21:48.952:INFO:oejdp.ScanningAppProvider:main: Deployment monitor
[file:///home/jetty/jetty-distribution-9.4.6.v20170531/webapps/] at interval 1
2017-07-24 16:21:48.969:INFO:oejs.AbstractConnector:main: Started
ServerConnector@6500df86{HTTP/1.1,[http/1.1]}{0.0.0.0:8081}
2017-07-24 16:21:48.991:INFO:oejus.SslContextFactory:main: x509=X509@2f686d1f(1,h=[],w=[]) for
SslContextFactory@3fee9989(file:///home/jetty/jetty-distribution- 9.4.6.v20170531/etc/keystore,file
:///home/jetty/jetty-distribution-9.4.6.v20170531/etc/keystore)
2017-07-24 16:21:49.048:INFO:oejs.
AbstractConnector:main: Started ServerConnector@71318ec4{SSL,[ssl, http/1.1]}{0.0.0.0:8443}
2017-07-24 16:21:49.048:INFO:oejs.Server:main: Started @825ms
```



5. 기능소개

jetty://



5.3 HTTPS 포트 변경(1/2)

- jetty.http.port property 설정을 통해 포트를 변경할 수 있다.

```
[jetty ~/jetty-distribution-9.4.6.v20170531]$ java -jar start.jar jetty.ssl.port=8444
2017-07-24 16:28:18.373:INFO::main: Logging initialized @474ms to org.eclipse.jetty.util.log.StdErrLog
2017-07-24 16:28:18.541:WARN:oejs.HomeBaseWarning:main: This instance of Jetty is not running from
a separate {jetty.base} directory, this is not recommended. See documentation at
http://www.eclipse.org/jetty/documentation/current/startup.html
2017-07-24 16:28:18.607:INFO:oejs.Server:main: jetty-9.4.6.v20170531
2017-07-24 16:28:18.626:INFO:oejdp.ScanningAppProvider:main: Deployment monitor
[file:///home/jetty/jetty-distribution-9.4.6.v20170531/webapps/] at interval 1
2017-07-24 16:28:18.643:INFO:oejs.AbstractConnector:main: Started
ServerConnector@6500df86{HTTP/1.1,[http/1.1]}{0.0.0.0:8081}
2017-07-24 16:28:18.664:INFO:oejus.SslContextFactory:main: x509=X509@2f686d1f(1,h=[],w=[]) for
SslContextFactory@3fee9989(file:///home/jetty/jetty-distribution-
9.4.6.v20170531/etc/keystore,file:///home/jetty/jetty-distribution-9.4.6.v20170531/etc/keystore)
2017-07-24 16:28:18.722:INFO:oejs.AbstractConnector:main: Started
ServerConnector@71318ec4{SSL,[ssl, http/1.1]}{0.0.0.0:8444}
2017-07-24 16:28:18.723:INFO:oejs.Server:main: Started @825ms
```



5. 기능소개

jetty://



5.3 HTTPS 포트 변경(2/2)

- 영구적으로 포트를 변경해야 하는 경우에는 start.ini 파일에 포트를 추가한다.

```
## Number of selectors (-1 picks default based on number of cores)
```

```
# jetty.http.selectors=-1
```

```
## ServerSocketChannel backlog (0 picks platform default)
```

```
# jetty.http.acceptorQueueSize=0
```

```
## Thread priority delta to give to acceptor threads
```

```
# jetty.http.acceptorPriorityDelta=0
```

```
## HTTP Compliance: RFC7230, RFC2616, LEGACY
```

```
# jetty.http.compliance=RFC7230
```

```
jetty.ssl.port=8444
```

5. 기능소개

jetty://



5.4 웹 애플리케이션 배포(1/3)

1. 웹 애플리케이션의 디렉토리 구조

웹 애플리케이션은 서블릿, 필터, JSP 등의 동적인 콘텐츠와 HTML, Image 등의 정적 콘텐츠, 애플리케이션 지원 라이브러리 등이 있다. 웹 애플리케이션은 단일 웹 아카이브(WAR 파일) 또는 디렉토리 트리로 묶을 수 있다.

디렉토리명	설명
/WEB-INF/	웹 애플리케이션과 관련된 모든 것을 저장하는데 사용되는 디렉토리. 일반적으로 내부적으로 액세스하는 콘텐츠만 들어있다.
/WEB-INF/web.xml	웹 애플리케이션의 다양한 동작을 정의하는 필수 배포 설명자
/WEB-INF/classes/	Java Class 파일들의 위치
/WEB-INF/lib	JAR 파일 (라이브러리)



5. 기능소개

jetty://



5.4 웹 애플리케이션 배포(2/3)

2. 자동 배포

웹 애플리케이션을 배포하기 위한 가장 기본적인 방법은 WAR파일이나 Exploded WAR 디렉토리를 \$JETTY_HOME/webapps 폴더에 배치하는 것이다.

이 자동 배포는 WAR의 파일이름 또는 디렉토리 이름으로 Context 경로가 지정된다.

파일 또는 디렉토리	지정된 Context 경로
/webapps/footrope.war	http://host/footrope/
/webapps/lazaret-2.1.3-SNAPSHOT.war	http://host/lazaret-2.1.3-SNAPSHOT/
/webapps/belaying-pins/WEB-INF/web.xml	http://host/belaying-pins/
/webapps/root.war (예약 명)	http://host/
/webapps/root/WEB-INF/web.xml (예약 명)	http://host/



5. 기능소개

jetty://



5.4 웹 애플리케이션 배포(3/3)

3. XML을 통한 웹 애플리케이션 배포

파일 이름을 기반으로 하지 않는 Context 경로를 사용하려고 하는 경우에는 자동 배포 방식이 아닌 XML을 통한 웹 애플리케이션 배포가 가능하다.

\$JETTY_HOME/webapps 디렉토리에서 Context 배포 설명자 파일을 다음과 같이 구성하여 저장한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN"
"http://www.eclipse.org/jetty/configure\_9\_3.dtd">

<Configure class="org.eclipse.jetty.webapp.WebAppContext">
  <Set name="contextPath">/wiki</Set>
  <Set name="war">/opt/myapp/myapp.war</Set>
</Configure>
```



5. 기능소개

jetty://



5.5 Virtual Hosts 설정

가상 호스트

가상 호스트는 IP 주소 한 개에 여러 이름기반 웹 사이트를 운영하기 위해 필요하다.

Jetty는 다음과 같은 가상 호스트 이름 스타일을 지원한다. 가상 호스트 구성은 이전 장에서 설정했던 Context XML 파일에 아래와 같이 virtualHosts를 세팅한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN"
"http://www.eclipse.org/jetty/configure\_9\_3.dtd">
<Configure class="org.eclipse.jetty.webapp.WebAppContext">
  <Set name="contextPath">/blah</Set>
  <Set name="war"><Property name="jetty.webapps"/>blah.war</Set>
  <Set name="virtualHosts">
    <Array type="java.lang.String">
      <Item>10.0.0.23</Item>
      <Item>www.blah.com</Item>
      <Item>www.blah.net</Item>
      <Item>www.blah.org</Item>
    </Array>
  </Set>
</Configure>
```



5. 기능소개

jetty://



5.6 로깅(1/3)

1. Jetty 로깅

Jetty는 자체 `org.eclipse.jetty.util.log.Logger` 계층을 통해 로깅을 제공하며 기존 Java 로깅 프레임워크를 기본적으로 사용하지 않는다.

Jetty는 classpath 에 설정되어 있는 `jetty-logging.properties` 파일을 로드하여 property 중 `org.eclipse.jetty.util.log.class logger` 구현체를 로딩한다. 만약 지정된 logger가 없으면 기본적으로 `org.eclipse.jetty.util.log.StdErrLog class`가 로딩된다. 즉, console 화면에 로그가 출력된다. `Org.eclipse.jetty.util.log.class` property에 정의되는 구현 class들은 아래와 같다.

- Default Logging with [Jetty's StdErrLog](#)
- Using [Log4j or Log4j2 via SLF4J](#)
- Using [Logback via SLF4J](#)
- Using [Java Util Logging via SLF4J](#)
- Using Java Commons Logging via SLF4J



5. 기능소개

jetty://



5.6 로깅(2/3)

2. Request Log 설정

사용자 요청 당 하나의 로그 항목이 수신되며 일반적으로 표준 NCSA 형식으로 제공되므로 Webalizer와 같은 웹 로그 분석도구를 사용하여 쉽게 분석 할 수 있다.

표준 요청 로그 항목에는 클라이언트 IP 주소, 날짜, Method, URL, 결과, 전송크기, 참조자, 사용자 에이전트, 전송시간이 포함된다.

```
123.4.5.6 - - [20/Jul/2016:10:16:17 +0000]
```

```
"GET /jetty/tut/XmlConfiguration.html HTTP/1.1"
```

```
200 76793 "http://localhost:8080/jetty/tut/logging.html"
```

```
"Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.6) Gecko/20040614 Firefox/0.8" 342
```



5. 기능소개

jetty://



5.6 로깅(3/3)

- 웹 애플리케이션에 대한 Request Log 구성 방법
특정 웹 애플리케이션에 대해 별도의 요청 로그를 구성하려면 Context xml 파일에 다음을 추가한다.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE Configure PUBLIC "-//Jetty//Configure//EN" "http://www.eclipse.org/jetty/configure\_9\_3.dtd">
<Configure class="org.eclipse.jetty.webapp.WebAppContext">
...
<Call name="insertHandler">
  <Arg>
    <New id="RequestLog" class="org.eclipse.jetty.server.handler.RequestLogHandler">
      <Set name="requestLog">
        <New id="RequestLogImpl" class="org.eclipse.jetty.server.NCSARequestLog">
          <Set name="filename"><Property name="jetty.logs" default="./logs"/>/test-yyyy_mm_dd.request.log</Set>
          <Set name="filenameDateFormat">yyyy_MM_dd</Set>
          <Set name="append">true</Set>
        </New>
      </Set>
    </New>
  </Arg>
</Call>
</Configure>
```



6. 활용예제

세부 목차

jetty://



본 예제는 간단한 웹 애플리케이션을 maven project로 만들어 Jetty 서버를 기동하여 테스트 하는 것을 목표로 한다.

6.1 Jetty 서버 기동 예제



6. 활용예제

jetty://



1. Jetty 서버 기동 예제(1/8)

다음은 Maven 을 사용하여 "Helloworld"를 출력하는 웹 애플리케이션을 실행해 보자.
먼저 maven pom.xml 파일 내용은 아래와 같다.

- pom.xml

```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/maven- v4\_0\_0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <packaging>war</packaging>
  <groupId>org.eclipse.jetty.demo</groupId>
  <artifactId>jetty-helloworld-webapp</artifactId>
  <version>1.0</version>
  <dependencies>
    <dependency>
      <groupId>javax.servlet</groupId>
      <artifactId>javax.servlet-api</artifactId>
      <version>3.1.0</version>
      <scope>provided</scope>
    </dependency>
  </dependencies>
```

뒷장 계속...



6. 활용예제

jetty://



1. Jetty 서버 기동 예제(2/8)

- pom.xml (계속)

```
<build>
  <plugins>
    <plugin>
      <groupId>org.eclipse.jetty</groupId>
      <artifactId>jetty-maven-plugin</artifactId>
      <version>9.4.6.v20170531</version>
    </plugin>
    <plugin>
      <groupId>org.apache.maven.plugins</groupId>
      <artifactId>maven-war-plugin</artifactId>
      <version>2.6</version>
      <configuration>
        <failOnMissingWebXml>>false</failOnMissingWebXml>
      </configuration>
    </plugin>
  </plugins>
</build>
</project>
```



6. 활용예제

jetty://



1. Jetty 서버 기동 예제(3/8)

- Servlet 소스 예제

```
package com.example;

import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;

@WebServlet(urlPatterns = {"//*"}, loadOnStartup = 1)
public class HelloWorldServlet extends HttpServlet
{
    @Override
    public void doGet(HttpServletRequest request, HttpServletResponse response)
        throws IOException
    {
        response.getOutputStream().print("Hello World");
    }
}
```



6. 활용예제

jetty://



1. Jetty 서버 기동 예제(4/8)

- 예제에서 사용하는 디렉토리 구조는 아래와 같다.

```
[jetty ~]$ tree jetty-helloworld-webapp/  
jetty-helloworld-webapp/  
├── pom.xml  
├── src  
│   ├── main  
│   │   ├── java  
│   │   │   ├── com  
│   │   │   │   ├── example  
│   │   │   │   │   └── HelloWorldServlet.java
```



6. 활용예제

jetty://



1. Jetty 서버 기동 예제(5/8)

- Maven 설치

```
[jetty ~]$ wget http://mirror.apache-kr.org/maven/maven-3/3.5.0/binaries/apache-maven-3.5.0-bin.tar.gz
--2017-07-25 15:30:37-- http://mirror.apache-kr.org/maven/maven-3/3.5.0/binaries/apache-maven-3.5.0-
bin.tar.gz
Resolving mirror.apache-kr.org (mirror.apache-kr.org)... 1.201.139.179
Connecting to mirror.apache-kr.org (mirror.apache-kr.org)|1.201.139.179|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 8534562 (8.1M) [application/x-gzip]
Saving to: 'apache-maven-3.5.0-bin.tar.gz'

100%[=====>]
8,534,562 5.53MB/s in 1.5s

2017-07-25 15:30:39 (5.53 MB/s) - 'apache-maven-3.5.0-bin.tar.gz' saved [8534562/8534562]

[jetty ~]$ tar xvfz apache-maven-3.5.0-bin.tar.gz
apache-maven-3.5.0/README.txt
apache-maven-3.5.0/LICENSE ap
ache-maven-3.5.0/NOTICE apach
e-maven-3.5.0/lib/
apache-maven-3.5.0/lib/cdi-api.license
...
```



6. 활용예제

jetty://



1. Jetty 서버 기동 예제(6/8)

- Maven 설치(계속)

1) bashrc파일에 maven 설치 경로를 아래와 같이 설정한다.

```
[jetty ~] vi ~/.bashrc
```

...

```
export M2_HOME=/home/jetty/apache-maven-3.5.0  
export PATH=$PATH:$M2_HOME/bin
```

2)설정 후 bashrc를 적용한다.

```
[jetty ~]$ source .bashrc
```



6. 활용예제

jetty://



1. Jetty 서버 기동 예제(7/8)

- Maven 실행

```
[jetty ~/jetty-helloworld-webapp]$ mvn jetty:run -Djetty.http.port=9999
[INFO] Scanning for projects...
[INFO]
[INFO] -----
[INFO] Building jetty-helloworld-webapp 1.0
[INFO] -----
[INFO]
[INFO] >>> jetty-maven-plugin:9.4.5.v20170502:run (default-cli) > test-compile @ jetty-helloworld-
webapp >>>
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ jetty-helloworld-webapp --- [
WARNING] Using platform encoding (UTF-8 actually) to copy filtered resources, i.e. build is platform
dependent!
[INFO] skip non existing resourceDirectory /home/jetty/jetty-helloworld-webapp/src/main/resources
...
[INFO] Started o.e.j.m.p.JettyWebAppContext@20de05e5{/file:///home/jetty/jetty-helloworld-
webapp/target/webapp-tmp/,AVAILABLE}{file:///home/jetty/jetty-helloworld-webapp/target/webapp-tmp/} [INFO] Starte
d ServerConnector@3fd9e827{HTTP/1.1,[http/1.1]}{0.0.0.0:9999}
[INFO] Started @37885ms
[INFO] Started Jetty Server
```



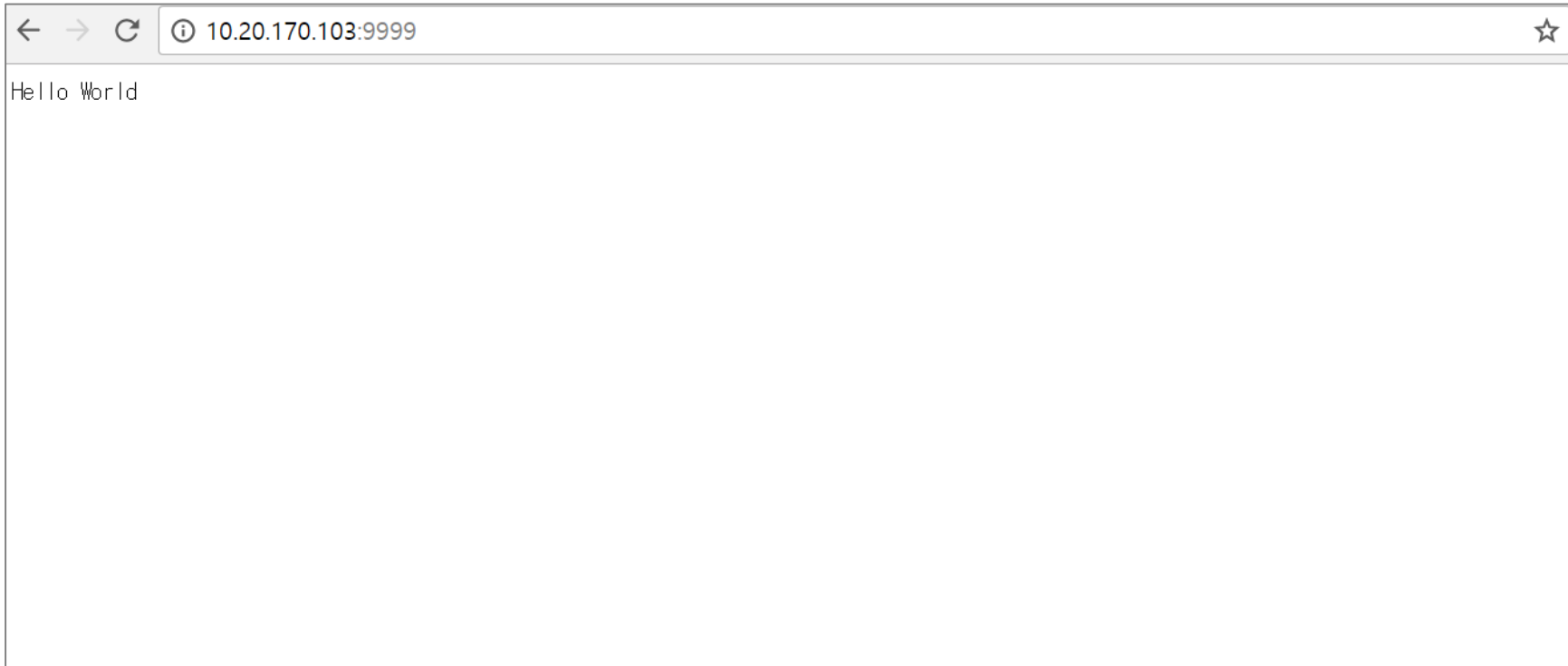
6. 활용예제

jetty://



1. Jetty 서버 기동 예제(8/8)

- 브라우저를 통해 maven으로 실행한 jetty 서버에 접속한다.





Q 왜 Jetty를 사용해야 합니까?

&

A Jetty는 다른 웹 서버 패키지와 달리 설치 공간이 아주 작습니다. Jetty는 개발자와 관리자에게 불필요한 오버헤드나 메모리 사용을 걱정하기 않고 배포할 수 있도록 할 수 있습니다. Jetty는 매우 클라우드 친화적입니다.

Q 실행중인 Jetty 버전을 어떻게 확인 할 수 있습니까?

&

A Jetty 설치 디렉토리에서 “`java -jar start.jar --version`” 으로 확인할 수 있습니다.





Q Jetty 는 WebSocket를 지원합니까?

&

A 네. Jetty는 RFC-6455 (<https://tools.ietf.org/html/rfc6455>) WebSocket Protocol을 지원합니다. 또한, JSR-356 (<https://www.jcp.org/en/jsr/detail?id=356>) Java WebSocket API를 지원하고 있습니다.

Q Jetty 는 HTTP/2 를 지원합니까?

&

A 네. Jetty는 RFC-7540 (<https://tools.ietf.org/html/rfc7540>) 에서 정의되어 있는 HTTP/2 프로토콜에 대한 클라이언트 및 서버 구현을 모두 지원합니다.



8. 용어정리

jetty://



용어	설명
JAVA EE	자바 엔터프라이즈 에디션 (Java Platform, Enterprise Edition; Java EE) 은 웹 애플리케이션 서버에서 동작하는 장애복구 및 분산 멀티티어를 제공하는 자바 소프트웨어의 기능을 추가한 서버를 위한 플랫폼이다.
SSL	TLS (Transport Layer Security) 및 SSL (Secure Sockets Layer) 은 “SSL” 이라고도 하며 컴퓨터 네트워크에서 통신 보안을 제공하는 암호화 프로토콜이다.
WebSocket	웹소켓 (WebSocket) 은 하나의 TCP 접속에 전이중(full-duplex) 통신채널을 제공하는 컴퓨터통신 프로토콜이다. 웹소켓 프로토콜은 2011년 IETF에 의해 RFC 6445 로 표준화되었으며 웹 IDL의 웹소켓 API는 W3C에 의해 표준화되고 있다.
HTTP/2	HTTP/2(Hypertext Transfer Protocol Version 2)는 월드와이드웹(WWW)에서 쓰이는 HTTP 프로토콜의 두 번째 버전이다. SPDY에 기반하고 있으며, 국제 인터넷 표준화 기구(IETF)에서 개발되고 있다. 2015년 5월, RFC 7540로 공개되었다.



Open Source Software Installation & Application Guide



이 저작물은 크리에이티브 커먼즈 [저작자표시-비영리-동일조건 변경허락 2.0 대한민국 라이선스]에 따라 이용하실 수 있습니다.