

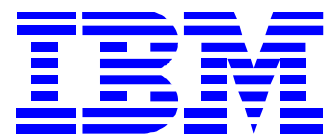
WebSphere Application Server v8.5.5

Liberty 서버 SSL 설정

(2013. 10.)

IBM SWG WebSphere Technical Sales

이정운 과장(juwlee@kr.ibm.com)



0) Liberty 서버 SSL 설정 간략 소개

안녕하세요 freeman 입니다.

이번 강좌에서 언급드릴 내용은 IBM WAS Liberty 서버 SSL 설정입니다. SSL 은 이미 많은 분들이 잘 알고 계시겠지만 Secure Sockets Layer 의 약자로서 인터넷 프로토콜인 HTTP 가 보안면에서 기밀성을 유지하지 못한다는 문제를 극복하기 위해 개발된 보안 프로토콜인 HTTPS 를 사용하기 위한 표준입니다.



IBM WAS Liberty 서버는 개발 뿐만 아니라 운영환경도 커버할 수 있는 하나의 완전한 WAS 이기 때문에 보안 목적으로 사용되는 SSL 표준을 준수하며 설정을 통해서 HTTPS 프로토콜을 IBM WAS 와 동일하게 사용 가능합니다.

이번 강좌에서는 보안 측면으로 SSL 을 사용하기 위한 설정과 간단한 Test 를 다뤄보도록 하겠습니다.

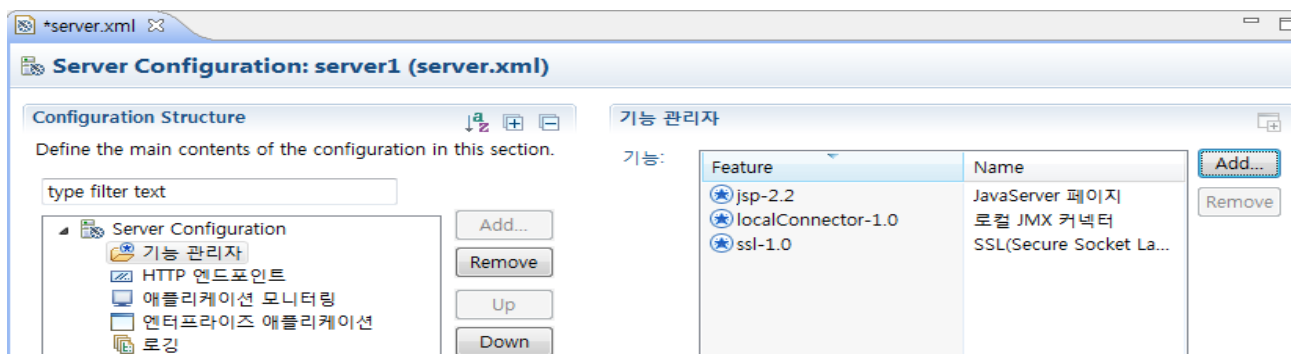
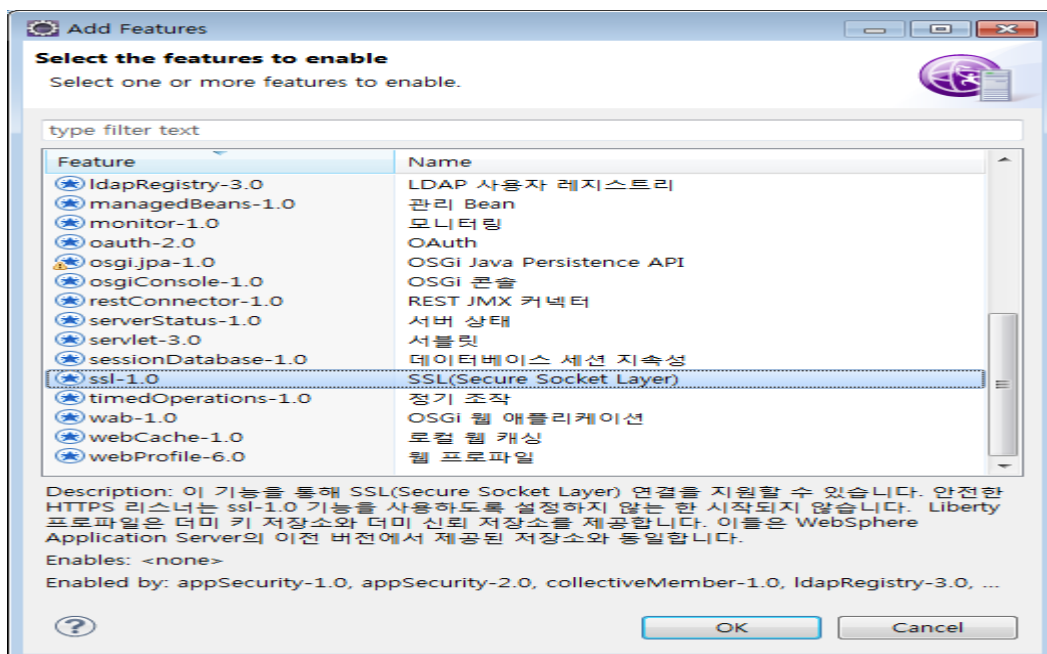
그럼 좀 더 자세한 사항은 이제부터 진행되는 강좌를 참고해주시기 바라면서 지금부터 강좌를 진행하도록 하겠습니다.

1) IBM WAS Liberty 서버에 대한 SSL 설정

1. Liberty 서버에 대한 SSL 설정을 하기전에 하단과 같이 단순하게 서비스를 처리하는 IBM WAS Liberty 가 한대 서비스 하고 있다고 가정합니다.



2. server.xml 마법사 화면에서 기능 관리자를 선택하고 Add 버튼을 클릭하여 SSL 스펙을 추가합니다.



3. SSL 스펙을 추가하고 나면 SSL 에 사용할 기본 SSL certificate 와 key DB file 을 하단과 같은 스크립트를 이용해서 생성합니다.

```
securityUtility createSSLCertificate --server=server1 --password=passw0rd --validity=365 --subject=CN=IBM,O=Liberty,C=TestCom
```

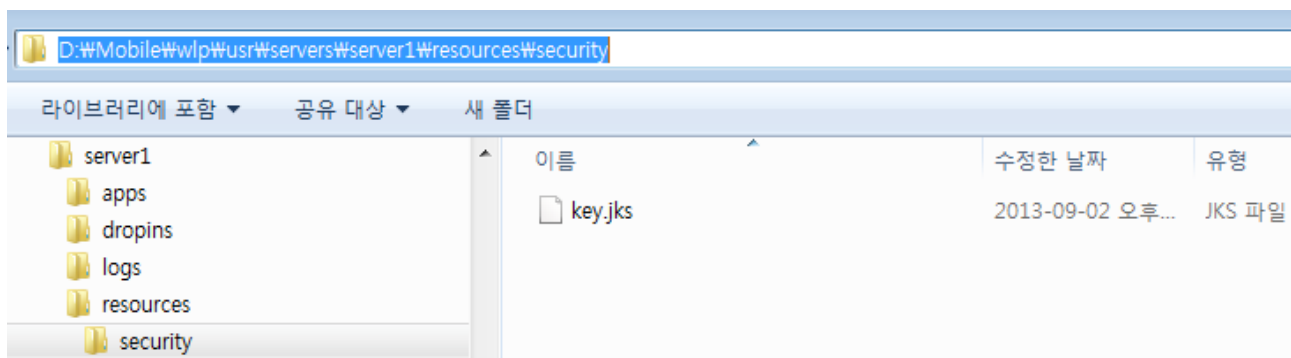
```
D:\Mobile\wlp\bin>securityUtility createSSLCertificate --server=server1 --password=passw0rd --validity=365 --subject=CN=IBM,O=Liberty,C=TestCom
키 저장소 D:\Mobile\wlp\usr\servers\server1\resources\security\key.jks 을(를) 작성하는 중입니다.
```

서버 server1에 대해 작성된 SSL 인증서

다음 행을 server.xml에 추가하여 SSL을 사용하십시오.

```
<featureManager>
  <feature>ssl-1.0</feature>
</featureManager>
<keyStore id="defaultKeyStore" password="{xor}Lz4sLChvLTs=" />
```

4. 성공적으로 SSL certificate 와 Key DB file 이 생성되면 하단과 같은 위치에서 key.jks 파일을 확인할 수 있습니다.



5. 또한, Java 가 가지고 있는 keytool 이라는 명령을 통해 해당 Key DB file 안에 들어있는 certificate 에 대한 내용을 하단과 같이 확인할 수 있습니다.

```
keytool -list -v -keystore D:\Mobile\wlp\usr\servers\server1\resources\security\key.jks
```

```
D:\Mobile\wlp\bin>keytool -list -v -keystore D:\Mobile\wlp\usr\servers\server1\resources\security\key.jks
키 스토어 비밀번호 입력:

키 스토어 유형: jks
키 스토어 제공자: IBMJCE

키 스토어에 1 항목이 포함되어 있습니다.

별명 이름: default
작성 날짜: 2013. 9. 2
인입 유형: keyEntry
인증서 체인 길이: 1
인증서1:
소유자: CN=IBM, O=Liberty, C=TestCom
발행자: CN=IBM, O=Liberty, C=TestCom
일련 번호: 52241094
유효 기간: 13. 9. 2 오후 1:14 - 14. 9. 2 오후 1:14
인증서 지문:
MD5: C9:58:90:76:EB:5D:A9:F1:06:A9:82:51:B7:7C:D0:DD
SHA1: 07:F6:A6:93:28:96:C0:0A:05:F0:94:4E:A3:96:26:35:E8:E5:F5:51
```

6. IBM WAS Liberty 서버의 설정파일인 server.xml 에 key DB file 을 만들면서 자동으로 생성된 설정 정보를 추가합니다.

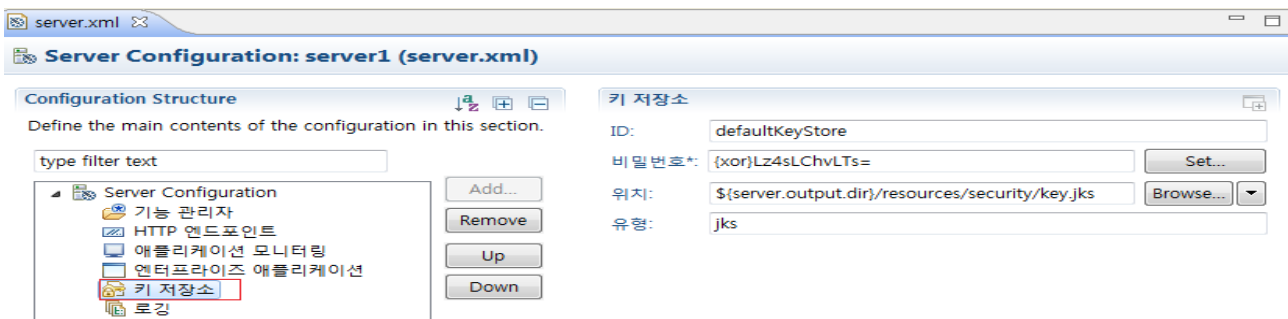
```
<server description="new server">
  <!-- Enable features -->
  <featureManager>
    <feature>jsp-2.2</feature>
    <feature>localConnector-1.0</feature>
    <feature>ssl-1.0</feature>
  </featureManager>

  <httpEndpoint host="localhost" httpPort="9080" httpsPort="9443" id="defaultHttpEndpoint">
  </httpEndpoint>
  <applicationMonitor updateTrigger="mbean" dropinsEnabled="false" />

  <enterpriseApplication id="TestFilterEAR" location="TestFilterEAR.ear" name="TestFilterEAR" />

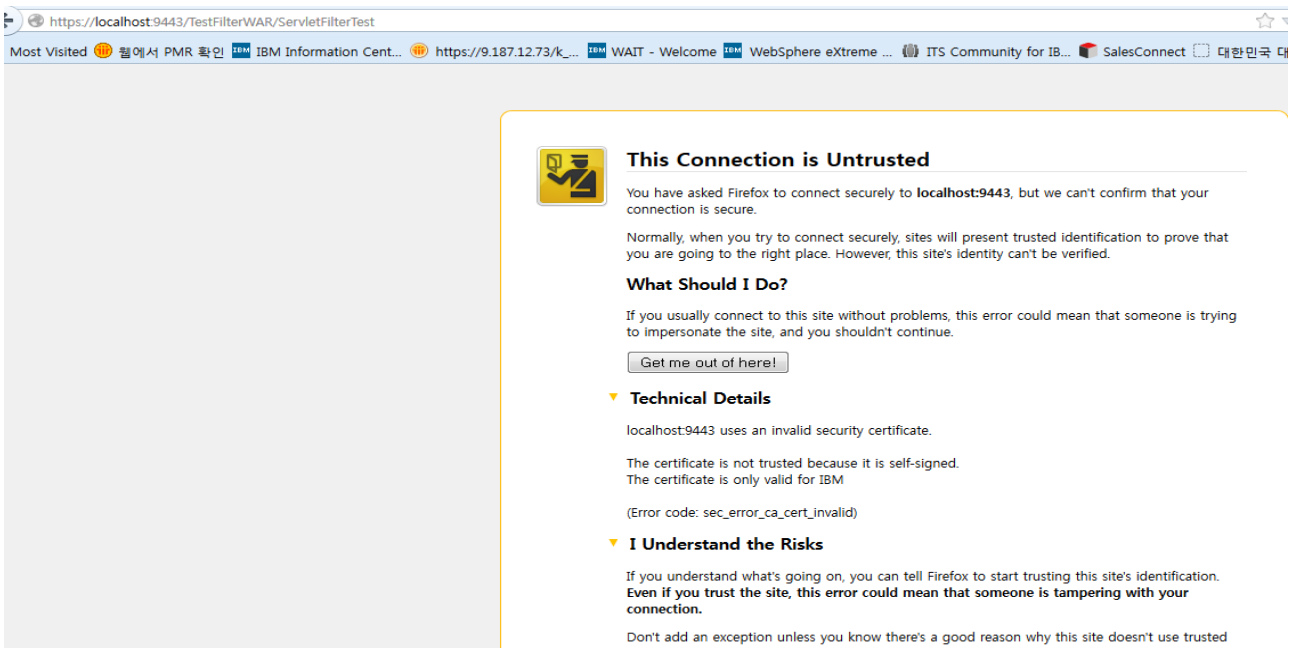
  <keyStore id="defaultKeyStore" password="{xor}Lz4sLChvLTs=" />

  <logging></logging>
</server>
```

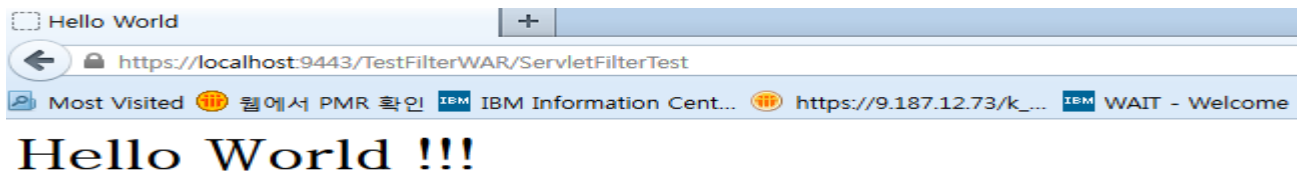


7. IBM WAS Liberty 서버를 재시작 한 후 HTTPS 프로토콜을 사용해서 샘플 요청을 수행합니다. 정상적으로 HTTPS 요청을 수행하면 하단과 같이 신뢰되지 않은 사이트이며 인증서를 예외 인증 할 것이냐는 화면을 확인할 수 있습니다.

<https://localhost:9443/TestFilterWAR/ServletFilterTest>



8. 해당 인증서에 대한 예외 인증을 하게 되면 하단과 같이 HTTPS 로 정상적인 서비스가 되는 것을 확인할 수 있습니다.



Server : server1

Call Count : 1

Session ID : -0hRbrzhw7Uyfhi6PJrx-IT

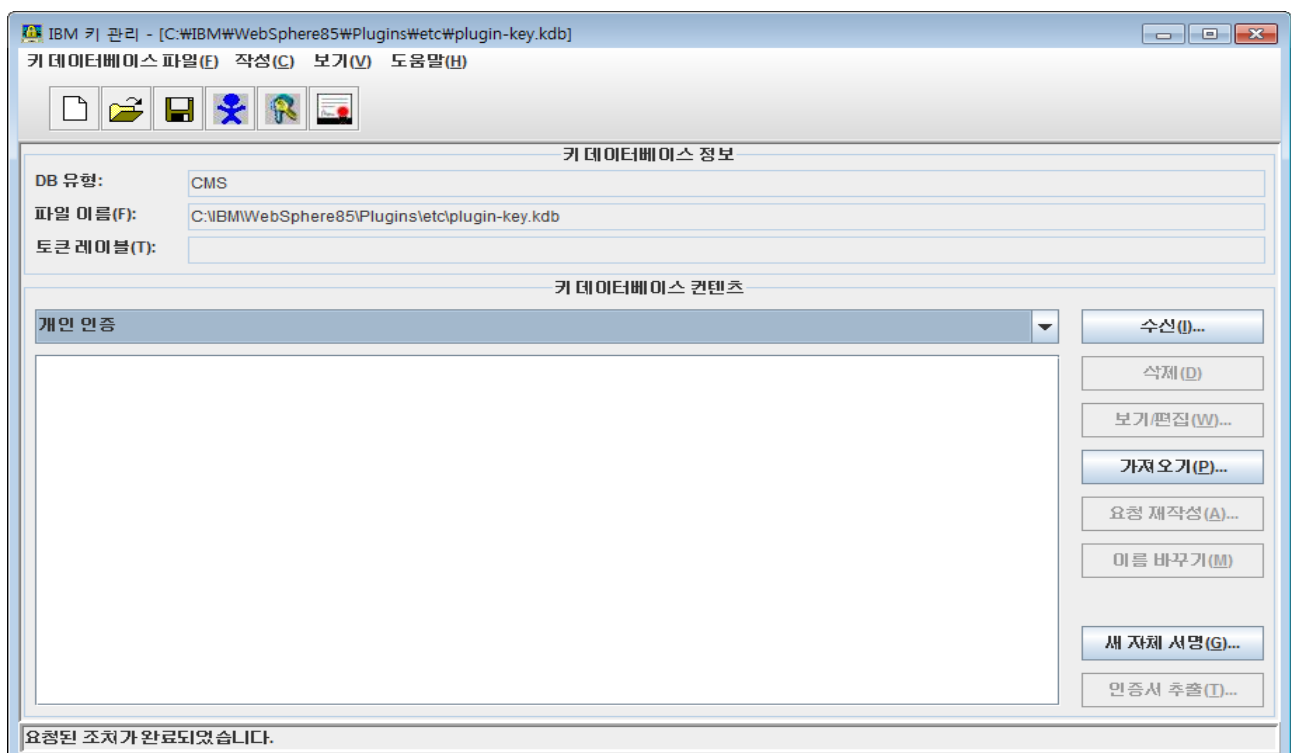
9. 이렇게 하시면 아주 단순하게 IBM WAS Liberty 를 이용해서 SSL 설정을 완료하시고 HTTPS 서비스 수행 테스트를 완료하신 것입니다.

2) IBM HTTP Server 와 IBM WAS Liberty 간의 SSL 설정

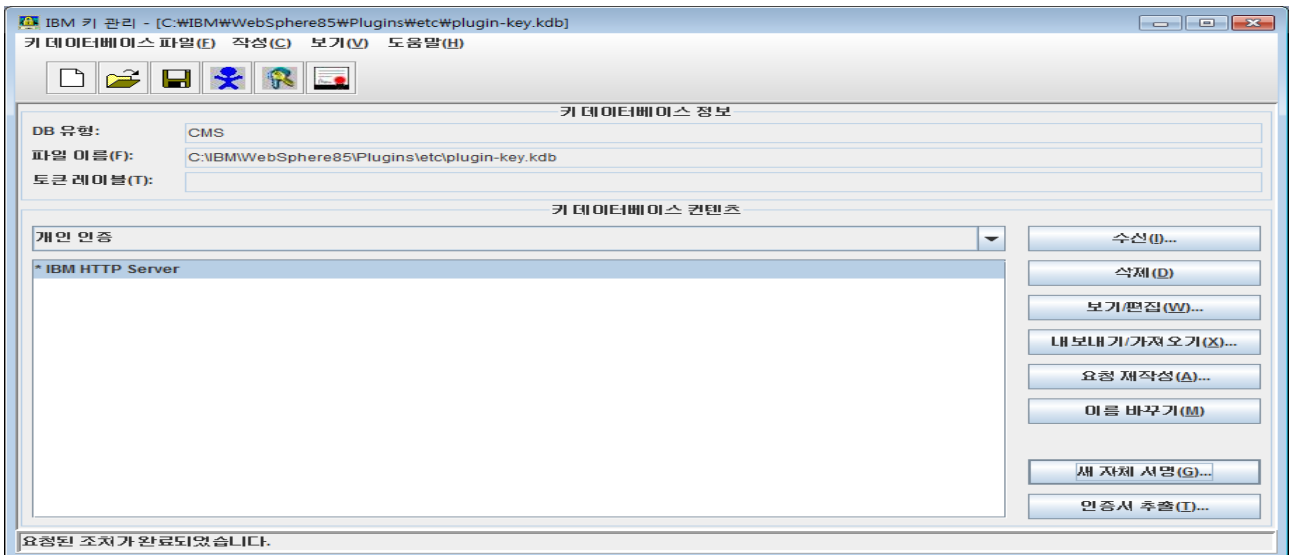
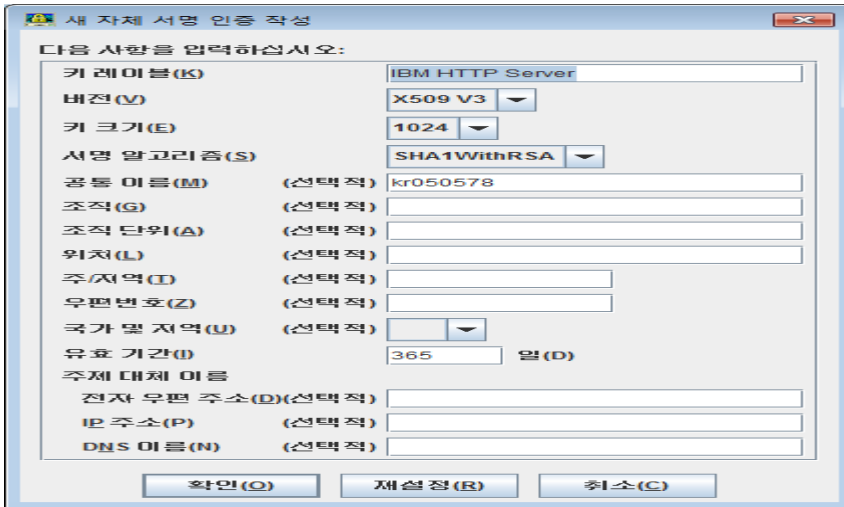
1. IBM HTTP Server 와 IBM WAS Liberty 간의 SSL 설정을 위하여 IBM HTTP Server 의 httpd.conf 파일에서 기본적인 SSL 설정을 수행합니다. (여기서 사용되는 Key DB file 은 IBM WAS 의 plugin 모듈에서 기본으로 제공되는 Key DB File 입니다. 만약 대외 서비스를 수행하고 있다면 공인 인증업체에게서 받은 Key DB file 위치를 지정해 주시면 되며 하단의 personal cetificate 생성단계는 넘어가면 됩니다.)

```
# Start of example SSL configuration
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so
Listen 0.0.0.0:8443
<VirtualHost *:8443>
SSLEnable
SSLProtocolDisable SSLv2
</VirtualHost>
SSLDisable
KeyFile C:\IBM\WebSphere85\Plugins\etc\plugin-key.kdb
SSLV3Timeout 1000
# End of example SSL configuration
```

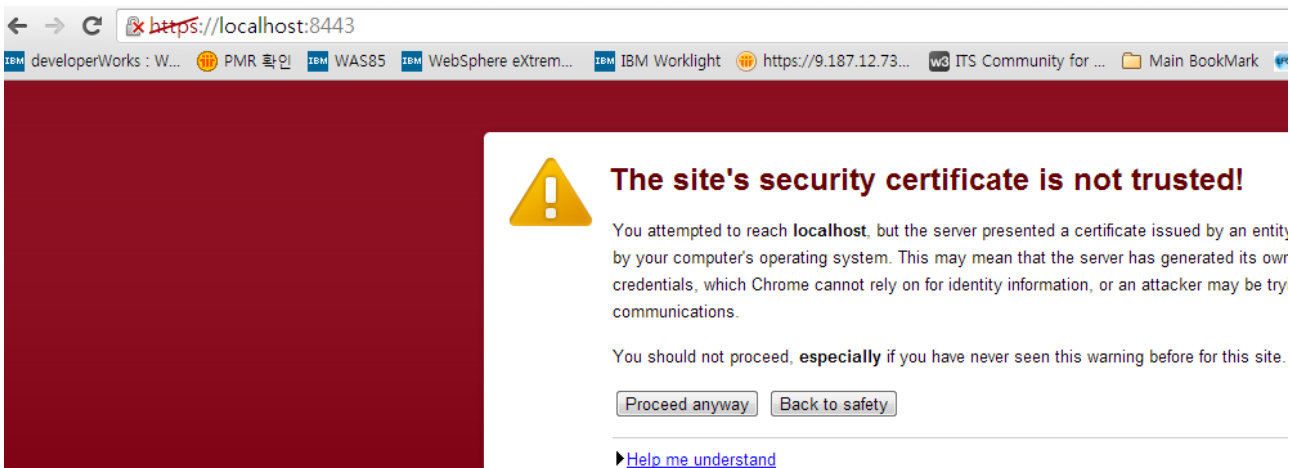
2. IBM HTTP Server 에서 설정된 Key DB file 에 기본 사용할 개인 인증키(personal certificate) 를 만들기 위하여 ikeyman 을 수행하고 설정에 명시된 plugin-key.kdb 파일을 엽니다. (ikeyman 은 IBM 에서 무료 제공되는 보안 유틸리티 이며, plugin-key.kdb 파일은 기본적으로 변경이 없었다면 “WebAS” 라는 암호를 사용합니다.)



3. plugin-key.kdb 파일에 아무런 개인 인증키가 없다면 새 자체 서명 버튼을 클릭하여 하나를 생성합니다.

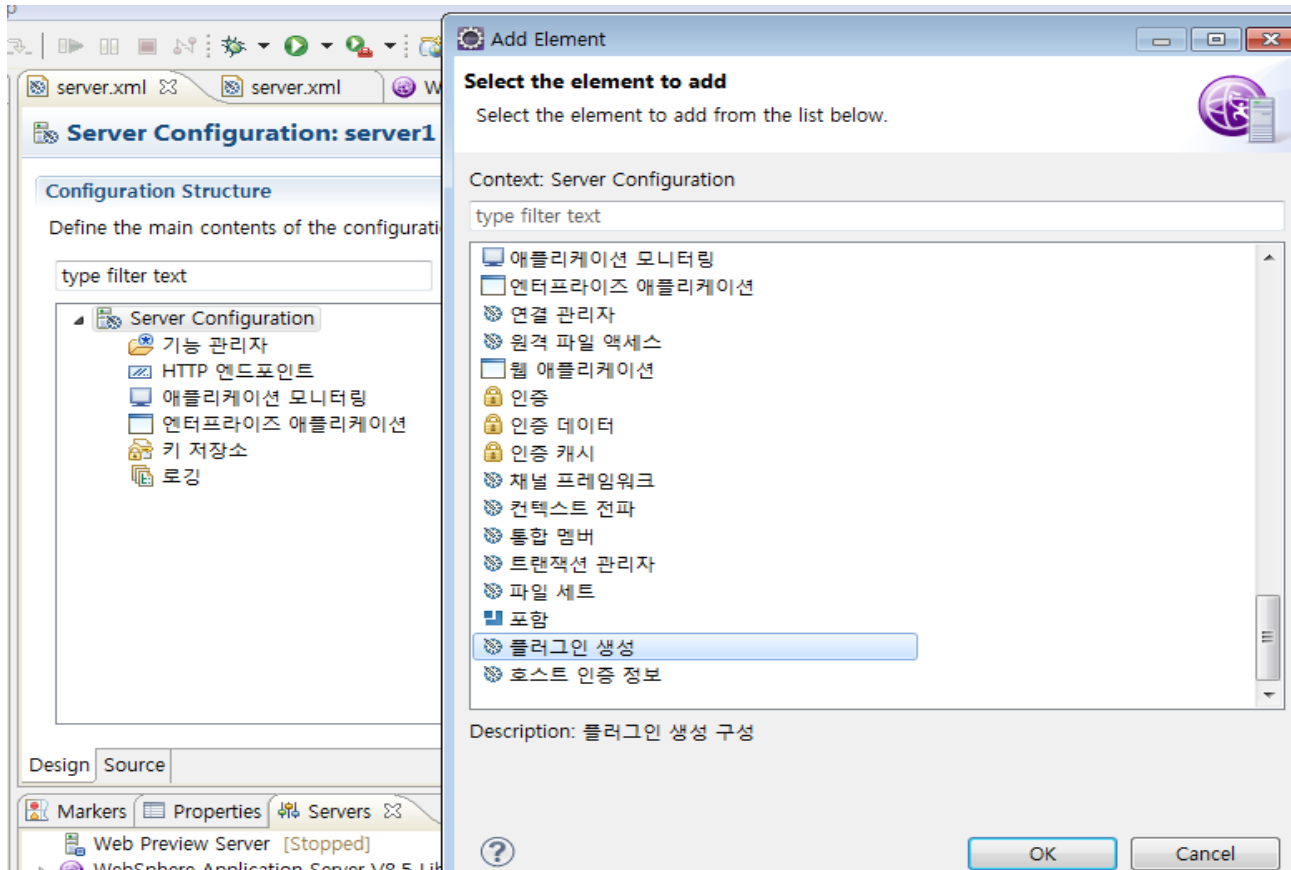


4. 정상적인 설정이 완료된 후 IBM HTTP Server 를 재시작하고 HTTPS 를 사용하여 보안 포트에 접근하면 하단과 같이 SSL 적용이 완료된 것을 확인할 수 있습니다.



5. IBM HTTP Server 에 대한 SSL 설정이 완료되었으면 이제 IBM HTTP Server 와 IBM WAS Liberty 서버 간의 SSL 설정을 위한 설정 작업을 진행하도록 하겠습니다.

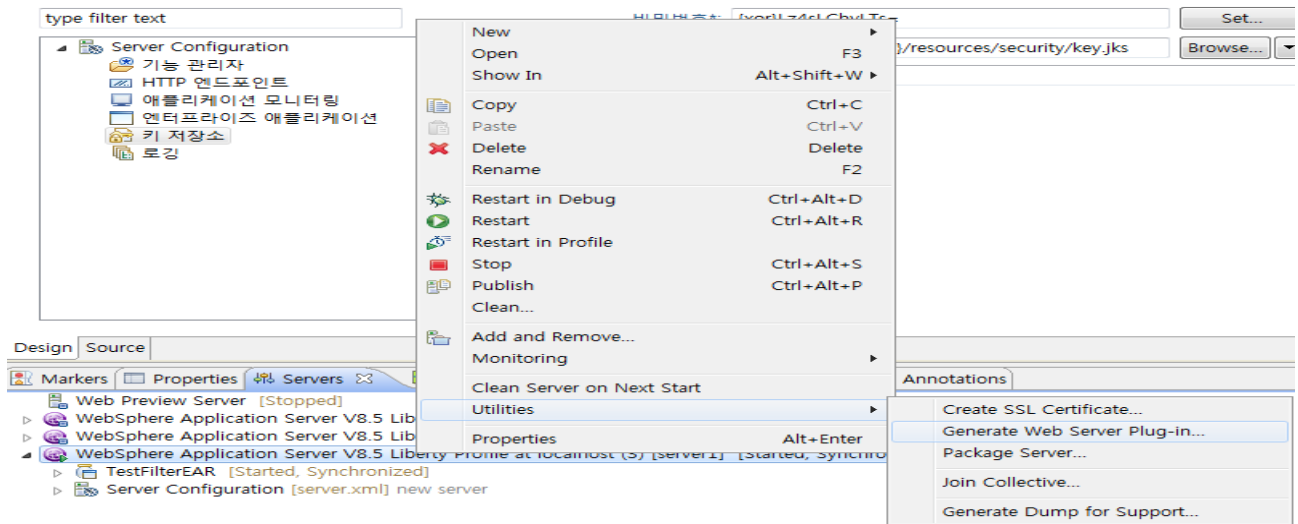
6. IBM WAS Liberty 서버에서 IBM HTTP Server 를 위한 적합한 플러그인 설정을 생성하기 위하여 server.xml 에서 Add 버튼을 클릭하여 플러그인 생성 기능을 추가합니다.



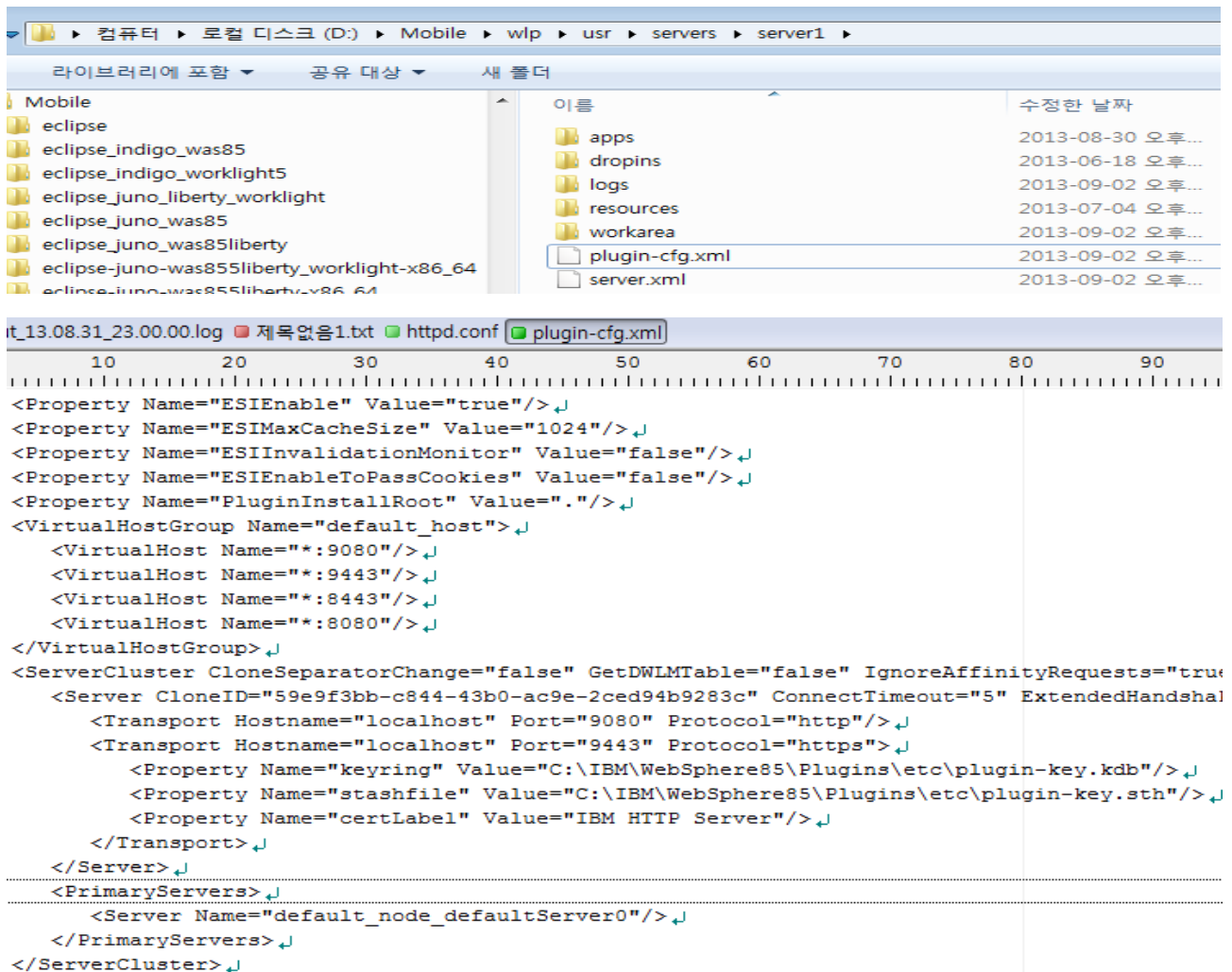
7. 플러그인 생성 기능이 추가되면 IBM HTTP Server 에 적합한 형태로 포트와 SSL 키 위치를 지정해 줍니다.



8. 설정 지정이 완료되면 IBM WAS Liberty 서버를 다시 구동시킨 후에 마우스 우 클릭한 후 Utilities > Generate Web server Plug-in 메뉴를 클릭하여 IBM HTTP Server 를 위한 Plugin 정보를 생성합니다.

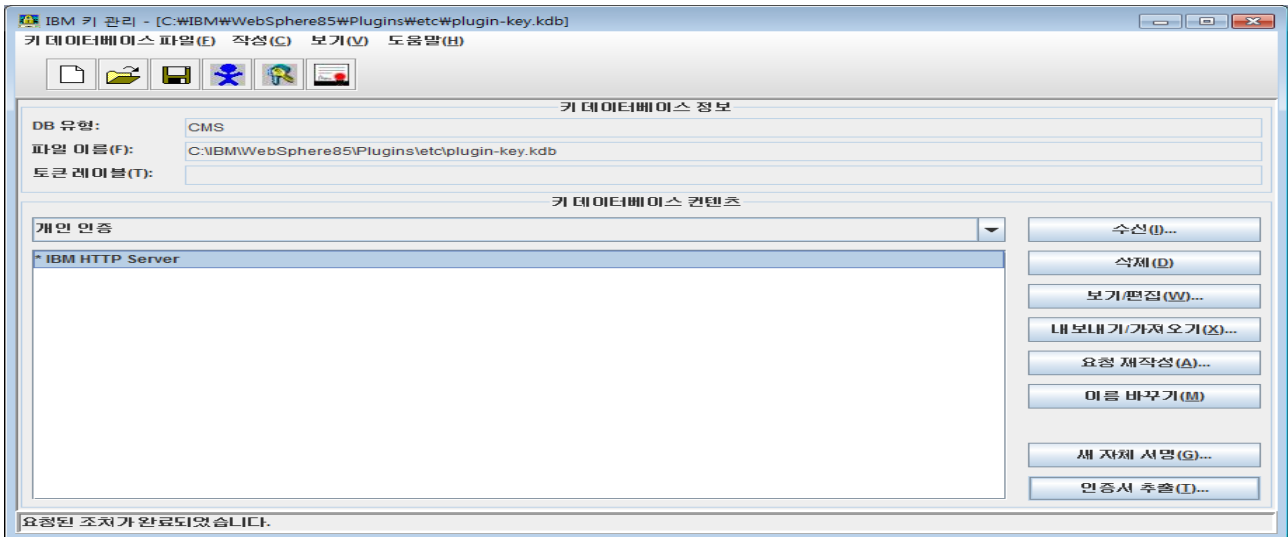


9. 위의 단계가 문제 없이 완료되면 하단과 같이 IBM WAS Liberty 서버 폴더에 plugin-cfg.xml 파일이 생성된 것을 확인할 수 있습니다.

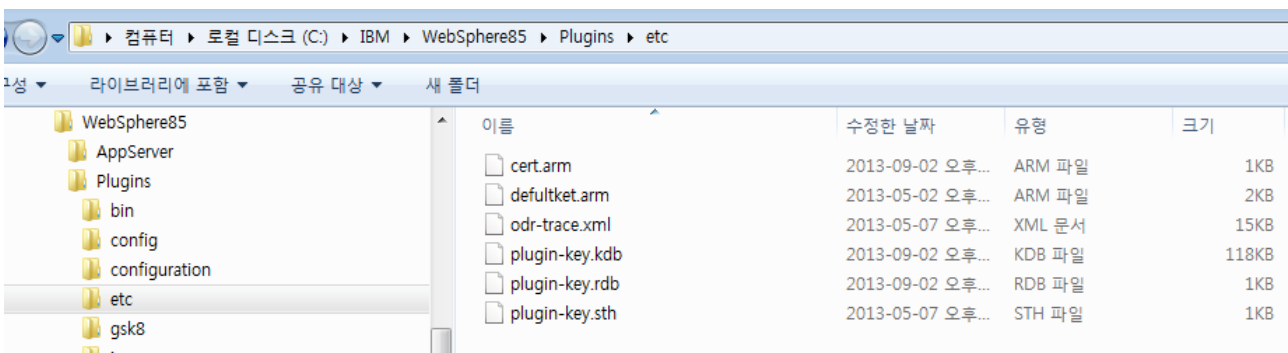
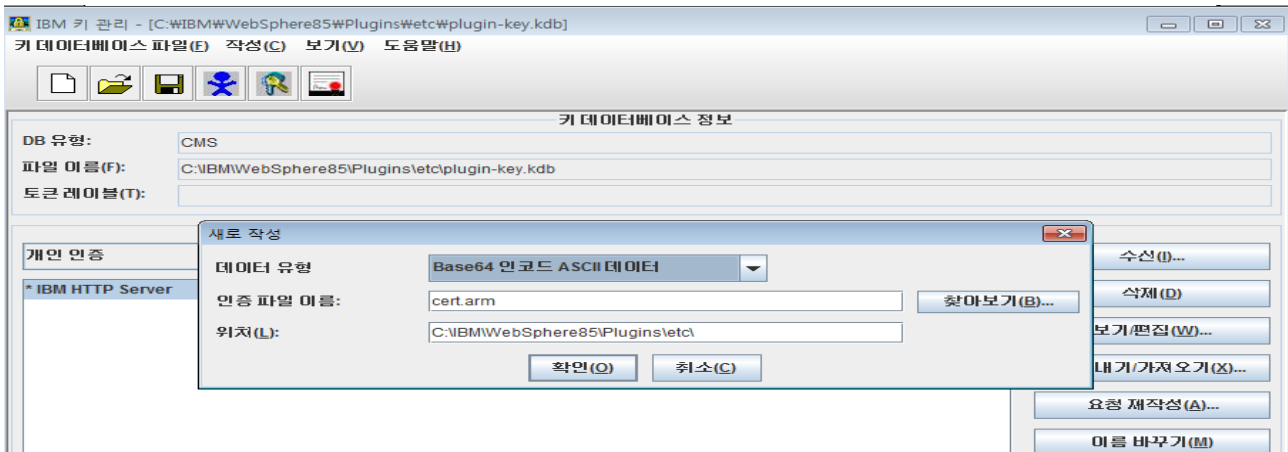


10. IBM HTTP Server 에 대한 설정이 완료되었으면 이제 IBM HTTP Server 와 IBM WAS Liberty 서버간의 SSL 을 위해서 각 Key DB file 에서 사용된 개인 인증키를 서로 인증할 수 있도록 교환하는 작업을 수행합니다.

11. IBM HTTP Server 의 플러그인에서 사용된 개인 인증키를 추출하기 위하여 ikeyman 유틸리티를 실행하고 Key DB file 을 오픈한 후 맨 오른쪽 하단에 있는 인증서 추출 메뉴를 클릭합니다.



12. 인증 파일 이름과 위치를 입력하고 확인을 클릭하면 해당 개인 인증 키가 별도의 file 로 추출된 것을 확인할 수 있습니다.



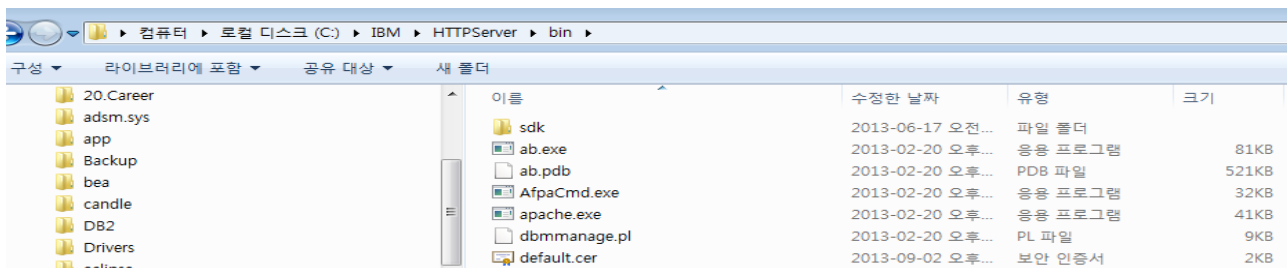
13. 추출된 개인 인증키를 IBM WAS Liberty 서버가 사용하는 Key DB file 에 추가하여 서로 인증할 수 있도록 합니다.

```
keytool -import -alias pluginkey -file C:\IBM\WebSphere85\Plugins\etc\Wcert.arm -keystore D:\Mobile\wlp\usr\servers\server1\resources\security\key.jks
```

```
C:\IBM\HTTPServer\bin>keytool -import -alias pluginkey -file C:\IBM\WebSphere85\Plugins\etc\Wcert.arm -keystore D:\Mobile\wlp\usr\servers\server1\resources\security\key.jks
키 스토어 비밀번호 입력:
소유자: CN=kr050578
발행자: CN=kr050578
일련 번호: 52241553
유효 기간: 13. 9. 2 오후 1:34 - 14. 9. 2 오후 1:34
인증서 지문:
MD5: 98:F9:72:FC:BF:B2:01:F1:C0:09:80:2F:92:CD:FD:E7
SHA1: 85:34:BA:49:5A:EB:09:06:55:29:4C:8E:FF:C0:20:46:1F:5B:48:DB
이 인증서를 신뢰합니까? [아니오]: y
인증서가 키 스토어에 추가되었습니다.
```

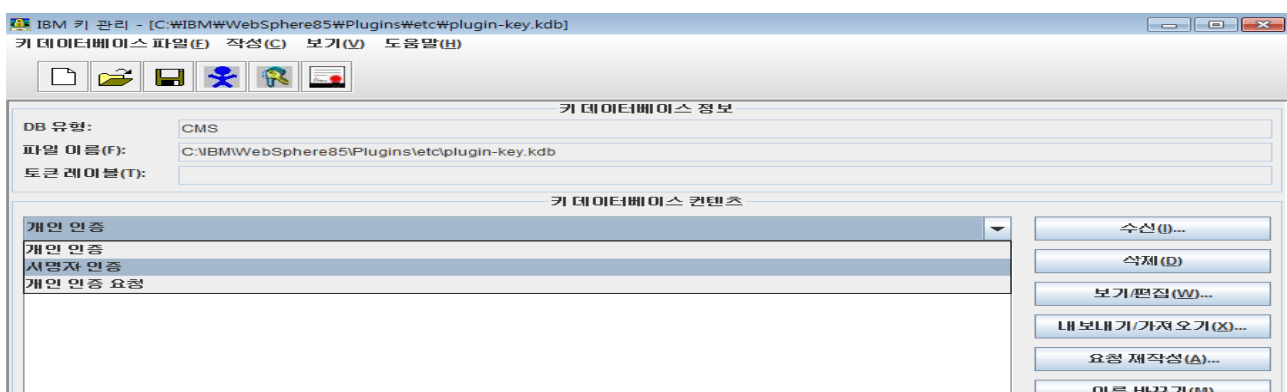
14. 위 단계와 동일하게 IBM WAS Liberty 서버에서 사용하는 Key DB file 에서 개인 인증키를 추출합니다

```
C:\IBM\HTTPServer\bin>keytool -export -alias default -keystore D:\Mobile\wlp\usr\servers\server1\resources\security\key.jks -rfc -file default.cer
키 스토어 비밀번호 입력:
<default.cer> 파일에 인증서가 저장됩니다.
```

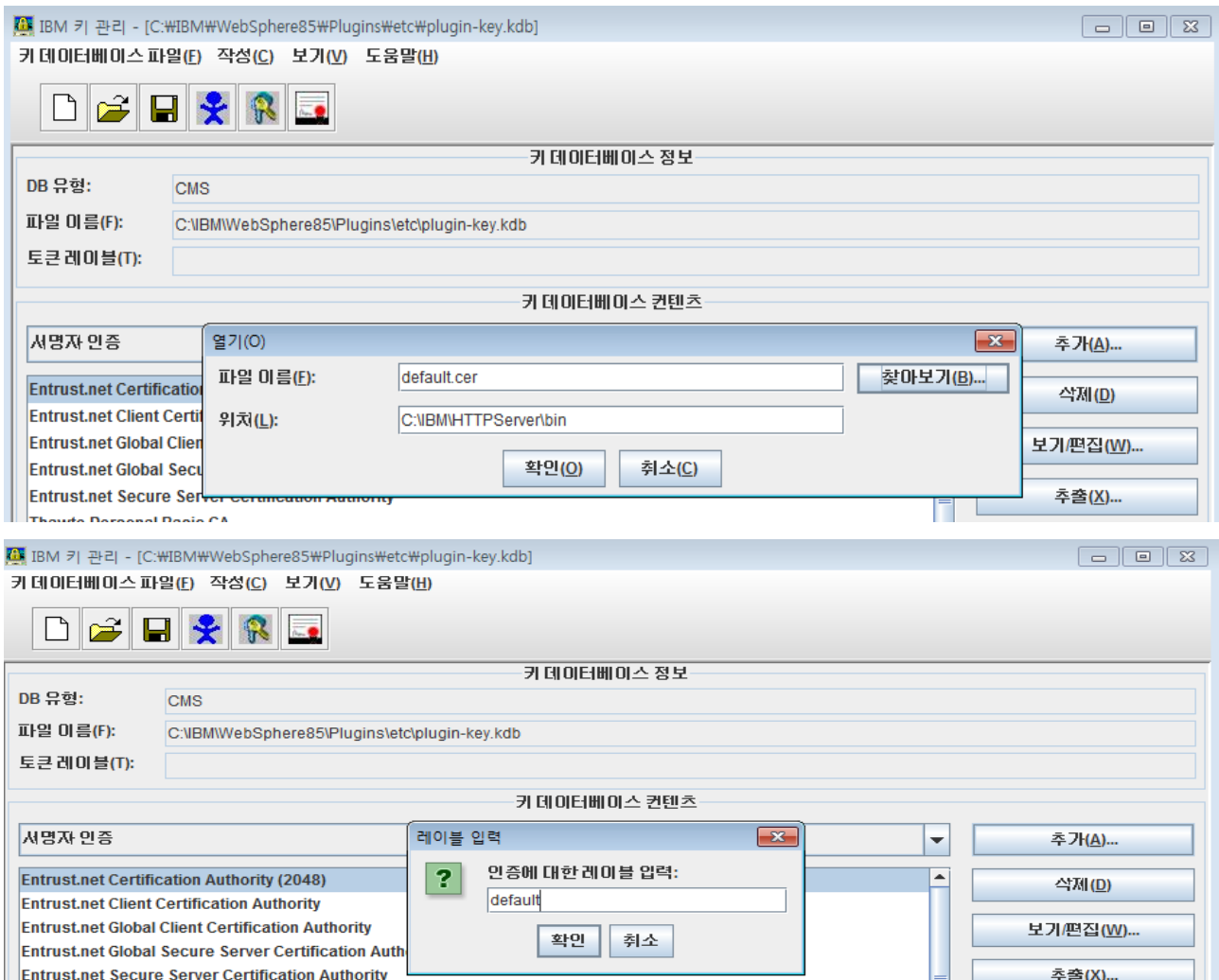


이름	수정된 날짜	유형	크기
sdk	2013-06-17 오전...	파일 폴더	
ab.exe	2013-02-20 오후...	응용 프로그램	81KB
ab.pdb	2013-02-20 오후...	PDB 파일	521KB
AlpaCmd.exe	2013-02-20 오후...	응용 프로그램	32KB
apache.exe	2013-02-20 오후...	응용 프로그램	41KB
dbmmanage.pl	2013-02-20 오후...	PL 파일	9KB
default.cer	2013-09-02 오후...	보안 인증서	2KB

15. 해당 키 추출이 완료되면 IBM HTTP Server 의 플러그인을 위한 plugin-key.kdb 파일을 새로 오픈한 후 서명자 인증에 해당키를 추가하기 위하여 서명자 인증을 선택합니다.



16. 서명자 인증에서 추가를 선택해서 이미 file 형태로 추출한 IBM WAS Liberty 서버의 개인 인증키를 선택하고 인증에 필요한 레이블을 입력합니다.



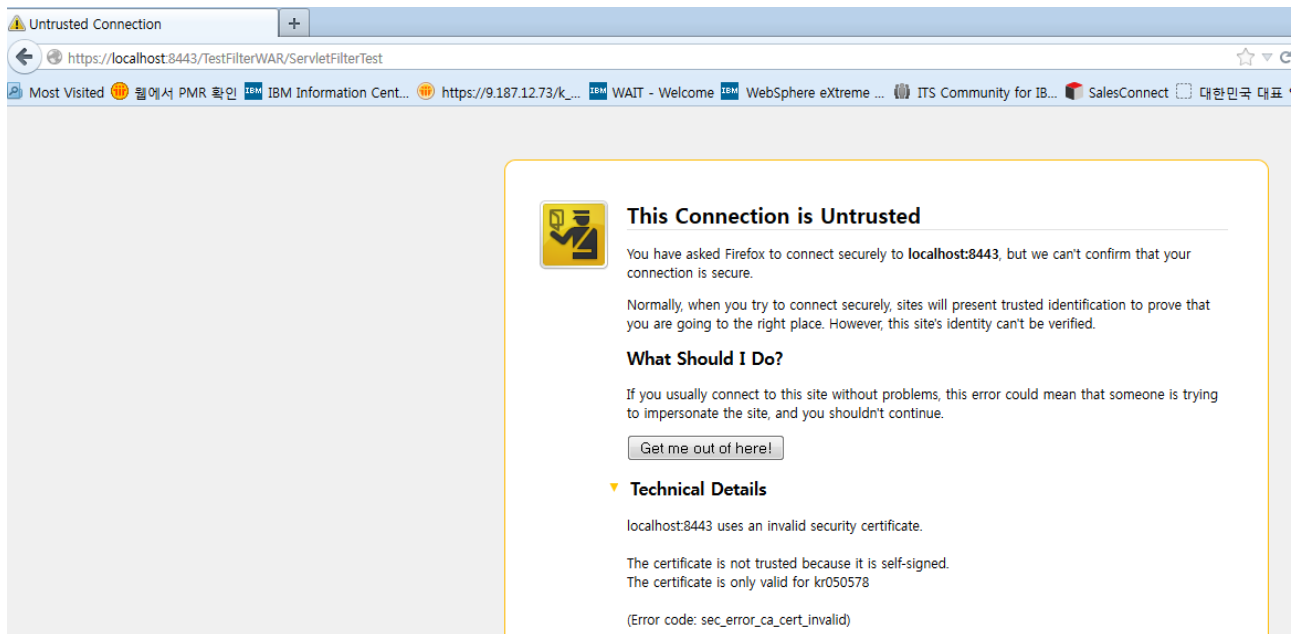
17. 해당 작업을 무사히 완료하면 하단과 같이 서명자 인증에 자신이 추가한 개인 키를 확인할 수 있습니다.



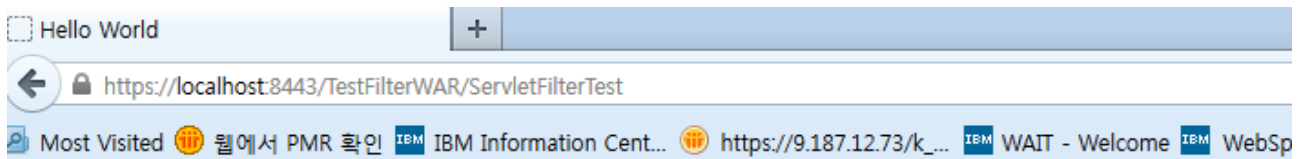
18. 지금까지 잘 따라오셨다면 IBM HTTP Server 의 플러그인과 IBM WAS Liberty 서버간의 개인키 교환이 완료되고 SSL 에 필요한 설정이 전부 완료된 것입니다.

19. 하단과 같이 IBM HTTP Server 의 보안포트를 이용해서 HTTPS 요청을 수행하면 정상적으로 보안규제 화면이 나오는 것을 확인할 수 있습니다 .

<https://localhost:8443/TestFilterWAR/ServletFilterTest>



20. 여기서 이전과 동일하게 예외 승인을 누르면 하단과 같이 정상적으로 HTTPS 프로토콜을 사용해서 SSL 이 적용된 채 서비스가 수행되는 화면을 확인할 수 있습니다.



Hello World !!!

Server : server1

Call Count : 1

Session ID : Ro1qFHnUw2ysv54aZRQ3M7o

팁 #1 : 고객사의 환경에 따라 다르겠지만 IBM HTTP Server 가 DMZ 안에 존재하기 때문에 IBM HTTP Server 와 IBM WAS Liberty 간에 SSL 이 필요없는 경우가 있을 수 있습니다. 이런 경우 Client 와 IBM HTTP Server 간에는 SSL 을 사용하고 IBM HTTP Server 와 IBM WAS Liberty 서버 간에는 SSL 을 사용하지 않는 방식을 일반적으로 SSL Offloading 이라고 칭합니다. 이러한 SSL Offloading 이 필요하다면 httpd.conf 파일을 하단과 같이 변경해주면 됩니다.

```
# Start of example SSL configuration ↵
LoadModule ibm_ssl_module modules/mod_ibm_ssl.so ↵
LoadModule proxy_module modules/mod_proxy.so ↵
LoadModule proxy_http_module modules/mod_proxy_http.so ↵
LoadModule headers_module modules/mod_headers.so ↵
↵
Listen 0.0.0.0:8443 ↵
<VirtualHost *:8443> ↵
SSLEnable ↵
SSLProtocolDisable SSLv2 ↵
↵
# SSL offloading ↵
ProxyRequests off ↵
ProxyPass / http://localhost:8080/ ↵
ProxyPassReverse / http://localhost:8080/ ↵
RequestHeader set X-Forwarded-Protocol "https" ↵
↵
</VirtualHost> ↵
SSLDisable ↵
KeyFile C:\IBM\WebSphere85\Plugins\etc\plugin-key.kdb ↵
SSLV3Timeout 1000 ↵
# End of example SSL configuration ↵
```

3) 결론

이번 강좌에서는 IBM WAS v8.5.5 에서 강화된 Liberty 서버의 기능중의 하나로 Liberty 서버를 이용해서 SSL 을 적용하는 것과 IBM HTTP Server 와 연동해서 SSL 을 사용하는 방법들을 살펴보는 시간을 가져 봤습니다. 이전 강좌에서도 언급했지만 이제부터는 IBM WAS Liberty 를 이용해서 개발 뿐만 아니라 필요하다면 운영 요건도 충분히 커버할 수 있습니다. 다시 말씀드려 이제 Liberty 서버는 개발부터 운영에 이르기까지 원하는 목적에 적합하게 활용할 수 있는 다양한 방법을 제공하고 있습니다.

그럼 여기서 이만 이번 강좌는 마무리하고 다음 강좌에서 뵙도록 하겠습니다. Go Go !!!

9) 참고 자료

1. 이 가이드는 IBM WAS v8.5.5 최초 사용자를 위한 기본 가이드 입니다.
2. IBM WAS 자체에 아직 익숙하지 않으신 분들은 가급적 기본강좌인 '하나씩 쉽게 따라 해보는 IBM WAS v7' 강좌와 '제대로 맛보는 IBM WAS v8.5' 강좌를 먼저 읽고 이 강좌를 읽으시는 것이 이해에 훨씬 도움이 됩니다.
(http://www.websphere.pe.kr/xe/?mid=was_info_re&page=3&document_srl=800
http://www.websphere.pe.kr/xe/?mid=was_info_re&page=2&document_srl=134863)
3. 가급적 IBM WAS v8.5 InfoCenter 의 해당 카테고리를 한 번 읽어보고 난 후에 작업하시기 바랍니다.
4. InfoCenter – WebSphere Application Server v8.5
(<http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp>)
5. InfoCenter – Liberty profile: SSL configuration attributes
(http://pic.dhe.ibm.com/infocenter/wasinfo/v8r5/index.jsp?topic=%2Fcom.ibm.websphere.wlp.express.doc%2Fae%2Ftwlp_sec_sso.html)