**keytool이용 인증서 만들기**

**1. keyStore 생성**

|  |
| --- |
|  |
| keytool -genkey -alias mindcare -keypass 123456 -storepass 123456 -keyalg RSA -keystore mindcare\_keystore -validity 3650  이름과 성을 입력하십시오.  [Unknown]: mindcare.geumcheon.go.kr  조직 단위 이름을 입력하십시오.  [Unknown]: geumcheon  조직 이름을 입력하십시오.  [Unknown]: geumcheon  구/군/시 이름을 입력하십시오?  [Unknown]: geumcheon  시/도 이름을 입력하십시오.  [Unknown]: seoul  이 조직의 두 자리 국가 코드를 입력하십시오.  [Unknown]: KR  CN=mindcare.geumcheon.go.kr, OU=geumcheon, O=geumcheon, L=geumcheon, ST=seoul, C=KR이(가) 맞습니까?  [아니오]: y |
| keytool -genkeypair -alias mindcare -dname cn=mindcare.geumcheon.go.kr -validity 3650 -keyalg RSA -keysize 2048 -keypass 123456 -storetype jceks -keystore mindcare\_keystore -storepass 123456  keytool -genkeypair -alias mindcare -dname "CN=mindcare.geumcheon.go.kr, OU=geumcheon, O=geumcheon, L=geumcheon, ST=seoul, C=KR" -validity 3650 -keyalg RSA -keysize 2048 -keypass 123456 -storetype jceks -keystore mindcare\_keystore -storepass 123456  <https://stackoverflow.com/questions/30693385/keytool-command-successful-on-command-line-but-not-via-processbuilder>  <https://www.ibm.com/support/knowledgecenter/ko/SSCQGF_7.1.1/com.ibm.IBMDI.doc_7.1.1/adminguide28.htm> |
|  |

**2. keyStore 인증 -> cer 인증서 생성**

|  |
| --- |
| keytool -export -alias mindcare -storepass 123456 -file server.cer -keystore mindcare\_keystore  인증서가 <server.cer> 파일에 저장되었습니다. |
|  |

**3. keystore로 cer 인증서 존재하는지 확인 및 추가**

|  |
| --- |
|  |
| **keytool -keystore mindcare\_geumcheon.keystore -importcert -alias tomcat\_https -file** mindcare\_geumcheon.cer  키 저장소 비밀번호 입력:  인증서가 <mindcare> 별칭 아래의 키 저장소에 존재합니다.  추가하겠습니까? [아니오]: Y  인증서가 키 저장소에 추가되었습니다. |
| **인증서 내용보기**  **keytool -list -keystore mindcare\_geumcheon.keystore** |

**4. cer 인증서 -> keyStore 에 등록하기**

|  |
| --- |
|  |
| **keytool -import -v -trustcacerts -alias mindcare -file server.cer -keystore mindcare\_store -keypass 123456 -storepass 123456**  소유자: CN=mindcare.geumcheon.go.kr, OU=geumcheon, O=geumcheon, L=geumcheon, ST=seoul, C=KR  발행자: CN=mindcare.geumcheon.go.kr, OU=geumcheon, O=geumcheon, L=geumcheon, ST=seoul, C=KR  일련 번호: 34783ce2  적합한 시작 날짜: Wed Jul 26 14:22:08 KST 2017, 종료 날짜: Tue Oct 24 14:22:08 KST 2017  인증서 지문:  MD5: 3B:29:78:E9:32:72:0D:E9:08:51:B4:0F:74:3A:7F:E6  SHA1: D7:96:98:AB:D0:3C:AB:36:04:A8:1E:C0:FB:E8:74:B5:21:CF:DB:0E  SHA256: FD:E2:E4:69:A7:AC:D9:C2:A1:05:C1:1C:5D:C3:E4:98:77:F5:F6:5B:5A:4D:38:6D:FC:55:77:17:6D:E1:40:2B  서명 알고리즘 이름: SHA256withRSA  버전: 3  확장:  #1: ObjectId: 2.5.29.14 Criticality=false  SubjectKeyIdentifier [  KeyIdentifier [  0000: B4 F0 25 DF FC E8 36 84 B0 0A DF 01 EE 6E F2 81 ..%...6......n..  0010: 72 75 95 76 ru.v  ]  ]  이 인증서를 신뢰합니까? [아니오]: y  인증서가 키 저장소에 추가되었습니다.  [mindcare\_store을(를) 저장하는 중] |

**4. tomcat server.xml설정**

|  |
| --- |
| <Connector port="8080" protocol="HTTP/1.1"  connectionTimeout="20000" maxThreads="150"  **SSLEnabled="true" clientAuth="false" keyAlias="mindcare"**  **keystoreFile="D:\https\_setup\_for\_push\mindcare\_keystore"**  **keystorePass="123456" scheme="https" secure="true" sslProtocol="TLS"**  />  .... |

https://joshua1988.github.io/web\_dev/tomcat-https-setup/#1-keystore-%EC%83%9D%EC%84%B1

1. KeyStore 생성

"%JAVA\_HOME%\bin\keytool" -genkey -alias tomcat\_https -keypass changeit -storepass changeit -keyalg RSA -keystore d:\https\_setup\_for\_push\.keystore

"%JAVA\_HOME%\bin\keytool" -genkey -alias tomcat\_https -keypass changeit -storepass changeit -keyalg RSA -keystore mindcare

2. 생성한 Keystore 를 인증서 cer 파일 형태로 저장하기

"%JAVA\_HOME%\bin\keytool" -export -alias tomcat\_https -storepass changeit -file d:\https\_setup\_for\_push\server.cer -keystore d:\https\_setup\_for\_push\.keystore

"%JAVA\_HOME%\bin\keytool" -export -alias tomcat\_https -storepass changeit -file d:\https\_setup\_for\_push\server.cer -keystore mindcare

"%JAVA\_HOME%\bin\keytool" -export -alias mindcare -storepass 123456 -file server.cer -keystore mindcare\_keystore

2.1 추가하기

keytool -keystore mindcare -importcert -alias tomcat\_https -file server.cer

비밀번호 설정

3. cer 인증서를 keystore 에 탑재하기

"%JAVA\_HOME%\bin\keytool" -import -v -trustcacerts -alias tomcat\_https -file d:\https\_setup\_for\_push\server.cer -keystore d:\https\_setup\_for\_push\.keystore -keypass changeit -storepass changeit

"%JAVA\_HOME%\bin\keytool" -import -v -trustcacerts -alias tomcat\_https -file d:\https\_setup\_for\_push\server.cer -keystore mindcare -keypass changeit -storepass changeit

"%JAVA\_HOME%\bin\keytool" -import -v -trustcacerts -alias mindcare -file server.cer -keystore mindcare\_store -keypass 123456 -storepass 123456

4. tomcat 의 https 설정 및 keystore 등록

<Connector SSLEnabled="true" clientAuth="false" keyAlias="tomcat\_https" keystoreFile="C:/https\_setup\_for\_push/.keystore" keystorePass="changeit" maxThreads="150" port="8443" protocol="HTTP/1.1" scheme="https" secure="true" sslProtocol="TLS"/>

keytool -certreq -alias tomcat\_https -keyalg rsa -file csr.txt -keystore

-----------

--키스토어 생성

keytool -genkey -keyalg RSA -alias mindcare -keystore mindcare\_keystore

--csr생성

keytool -certreq -alias mindcare -keyalg RSA -file csr.txt -keystore mindcare\_keystore

??

1. 여기서 csr를 전송해서 다른쪽에서 처리된 인증서를 받아서 그걸 등록하든

2. 내가 내스토어에 cer을 생성 하든해야된다.

??

--인증서 탑재

"%JAVA\_HOME%\bin\keytool" -import -v -trustcacerts -alias mindcare -file server.cer -keystore mindcare\_store -keypass 123456 -storepass 123456

<https://www.lesstif.com/pages/viewpage.action?pageId=17105864>

* [keystore 생성](https://www.lesstif.com/pages/viewpage.action?pageId=17105864#apachetomcat%EC%97%90%EC%84%9CSSL%28HTTPS%29%EC%84%9C%EB%B9%84%EC%8A%A4%EA%B0%80%EB%8A%A5%ED%95%98%EA%B2%8C%EC%84%A4%EC%A0%95-keystore%EC%83%9D%EC%84%B1)
  + [make keystore from scratch](https://www.lesstif.com/pages/viewpage.action?pageId=17105864#apachetomcat%EC%97%90%EC%84%9CSSL%28HTTPS%29%EC%84%9C%EB%B9%84%EC%8A%A4%EA%B0%80%EB%8A%A5%ED%95%98%EA%B2%8C%EC%84%A4%EC%A0%95-makekeystorefromscratch)
* [Server.xml 설정](https://www.lesstif.com/pages/viewpage.action?pageId=17105864#apachetomcat%EC%97%90%EC%84%9CSSL%28HTTPS%29%EC%84%9C%EB%B9%84%EC%8A%A4%EA%B0%80%EB%8A%A5%ED%95%98%EA%B2%8C%EC%84%A4%EC%A0%95-Server.xml%EC%84%A4%EC%A0%95)
* [테스트](https://www.lesstif.com/pages/viewpage.action?pageId=17105864#apachetomcat%EC%97%90%EC%84%9CSSL%28HTTPS%29%EC%84%9C%EB%B9%84%EC%8A%A4%EA%B0%80%EB%8A%A5%ED%95%98%EA%B2%8C%EC%84%A4%EC%A0%95-%ED%85%8C%EC%8A%A4%ED%8A%B8)
* [See Also](https://www.lesstif.com/pages/viewpage.action?pageId=17105864#apachetomcat%EC%97%90%EC%84%9CSSL%28HTTPS%29%EC%84%9C%EB%B9%84%EC%8A%A4%EA%B0%80%EB%8A%A5%ED%95%98%EA%B2%8C%EC%84%A4%EC%A0%95-SeeAlso)
* [Ref](https://www.lesstif.com/pages/viewpage.action?pageId=17105864#apachetomcat%EC%97%90%EC%84%9CSSL%28HTTPS%29%EC%84%9C%EB%B9%84%EC%8A%A4%EA%B0%80%EB%8A%A5%ED%95%98%EA%B2%8C%EC%84%A4%EC%A0%95-Ref)

tomcat 7.0 기준

## keystore 생성

외부에서 인증서 발급받은 인증서를 keystore 로 변경하는 방법은 차후 기술

### make keystore from scratch

* keystool 로 키와 인증서 생성
  + key 와 store 의 암호는 changeit
  + keystore file명은 home directory의 keystore
  + SSL 인증서의 DN은 "CN=localhost, OU=OrgUnit, O=MyCompany, C=KR"
  + localhost 대신 사용하는 URL 을 입력

**Windows**

|  |
| --- |
| "%JAVA\_HOME%\bin\keytool" -genkey -alias tomcat -keypass changeit -storepass changeit -keyalg RSA -keystore %USERPROFILE%\.keystore -dname "CN=localhost, OU=OrgUnit, O=MyCompany, C=KR" |

**Unix**

|  |
| --- |
| $JAVA\_HOME/bin/keytool -genkey -alias tomcat -keypass changeit -storepass changeit -keyalg RSA -keystore ${HOME}/.keystore -dname "CN=localhost, OU=OrgUnit, O=MyCompany, C=KR" |

## Server.xml 설정

server.xml 의 다음 Connector 의 주석을 해제

|  |
| --- |
| <Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"                 maxThreads="150" scheme="https" secure="true"                 clientAuth="false" sslProtocol="TLS" /> |

8443이 아닌 다른 Port(ex: 9443) 를 사용할 경우 다른 Connector 에서 사용하는 8443 포트를 변경한 Port 번호(9443) 으로 수정해야 함

## 테스트

1. tomcat 을 구동한다.
2. Browser로 <https://localhost:8443/> 입력하여 https 가 제대로 동작하는지 확인한다.

## See Also

* [tomcat7-maven-plugin - 메이븐에서 톰캣 구동해서 app deploy 하기](https://www.lesstif.com/pages/viewpage.action?pageId=17105864)

## Ref

* <http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html>

### Tomcat : CSR 생성 및 SSL 인증서 적용

<https://www.comodossl.co.kr/certificate/ssl-installation-guides/Tomcat-csr-crt.aspx>

|  |
| --- |
| **Tomcat 설치방법 보기** |

**1. 키파일 생성**   
  
keytool -genkey -keyalg RSA -sigalg SHA1withRSA **-alias hanbiro** -keysize 2048 -keystore **hanbiro.key**  
  
※ 참고사항(위의 빨간색 부분과 매칭되게  
hanbiro는 키의 alias 이름으로 임의로 작성합니다.  
hanbiro.key는 keystore이름으로 임의로 만듭니다  
  
Enter keystore password: (비밀번호를 입력하세요.)  
  
What is your first and last name?  
[Unknown]: www.hanbiro.com (인증서 사용을 원하는 도메인)  
What is the name of your organizational unit?  
[Unknown]: LINUX (부서명)  
What is the name of your organization?  
[Unknown]: HANBIRO (업체명)  
What is the name of your City or Locality?  
[Unknown]: SEOCHO (지역명)  
What is the name of your State or Province?  
[Unknown]: SEOUL(시/도)  
What is the two-letter country code for this unit?  
[Unknown]: KR  
Is CN=www.hanbiro.com, OU=LINUX, O=HANBIRO, L=SEOCHO, ST=SEOUL, C=KR correct?  
[no]: y  
  
Enter key password for <hanbiro>  
(RETURN if same as keystore password): (엔터를 입력)   
  
키파일이 제대로 생성되었는지 확인해 봅니다.   
keytool -list -keystore hanbiro.key   
  
  
**2. CSR 생성**   
  
**keytool -certreq -alias hanbiro -keyalg RSA -sigalg SHA1withRSA -file hanbiro.csr -keystore hanbiro.key**   
  
Enter keystore password: (키파일 생성시에 입력하였던 패스워드를 입력합니다.)   
  
  
**3. 생성된 CSR 예**   
  
한비로(www.comodossl.co.kr) 에서 인증서 신청하실 때 위에서 생성하신 CSR 내용을 복사해서 붙여 넣은후 나머지 설치 절차를 거칩니다.   
생성된 CSR 을 출력하면 아래와 같은 base64 형식의 문서를 볼 수 있습니다.

|  |
| --- |
| [root@ns root]# cat hanbiro.csr -----BEGIN NEW CERTIFICATE REQUEST----- MIIBsDCCARkCAQAwcDELMAkGA1UEBhMCS1IxDjAMBgNVBAgTBVN lb3VsMQ8wDQYDVQQHEwZTZW9j aG8xEDAOBgNVBAoTB0hhbmJpcm8xDjAMBgNVBAsTBUxpbnV4MR4w HAYDVQQDExV3d3cucHJvZGln AAGgADANBgkqhkiG9w0BAQQFAAOBgQBhV3jIaT2wEOB1/AIOedu+4 gECrr+6UIYhwPtSmIeoWXg5 76+UHe5I1M2M/ew5j6d8pq4IBXaTesSrmwZuuuA2Stx4uXjb/Akjr8UIDX isnycJGmk5dQDCCT3G 8IBd8gwgvQOiAhnfGSjIbStsPOiVCgB60uSz9Jc8s9rPIxh69w== -----END NEW CERTIFICATE REQUEST----- |

-----BEGIN … 부터 마지막 줄 -----END … 까지 복사하여 지정된 SSL 접수페이지에 복사하여 붙여 넣은 뒤 입력정보와 함께 전송하면 접수가 완료됩니다.   
  
  
**4. 인증서 설치**  
  
**keytool -import -trustcacerts -alias COMODOSSL -file  
COMODOSSLCA.crt -keystore hanbiro.key  
  
keytool -import -trustcacerts -alias INTER -file  
AddTrustExternalCARoot -keystore hanbiro.key  
  
keytool -import -trustcacerts -alias hanbiro -file  
www\_hanbiro.crt -keystore hanbiro.key**  
  
##주의##  
Free SSL 및 PostiveSSL 의 경우, PostiveSSLCA.crt 파일을 추가 하지 않았을 경우,  
아래와 같은 메세지가 표시 되며 저장 되지 않으니 반드시 아래 작업을 진행하여 주시기 바랍니다.  
또한 -alias 설정시 .등의 특수문자가 있을 경우 인증서를 적용할 수 없습니다.  
예) keytool -import -trustcacerts -alias www.hanbiro.com -file PositiveSSLCA.crt -keystore hanbiro.key (X)  
keytool -import -trustcacerts -alias wwwhanbirocom -file PositiveSSLCA.crt -keystore hanbiro.key (0)   
**에러내용  
keytool error: java.lang.Exception: Failed to establish chain from reply   
  
추가내용   
keytool -import -trustcacerts -alias POSITIVESSL -file PositiveSSLCA.crt -keystore hanbiro.key**  
  
**5. 서버설정**  
Server.xml을 설정합니다.

# Tomcat 버젼 8.X 의 경우..  
<Connector port="443" protocol="HTTP/1.1"  
   maxThreads="150" SSLEnabled="true" scheme="https" secure="true"  
   clientAuth="false" sslProtocol="TLS"  
   keystoreFile="키스토어파일경로/hanbiro.key" keystorePass="hanbiro.key 패스워드"   
/>

# Tomcat 버젼 7.X 의 경우..  
< -- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->  
<Connector port="8443"  
   maxThreads="200"  
   scheme="https" secure="true" SSLEnabled="true"  
   keystoreFile="키스토어파일경로/hanbiro.key" keystorePass="(hanbiro.key 패스워드)" clientAuth="false" sslProtocol="TLS"  
/>  
  
# Tomcat 버젼 6.X 의 경우..  
<!--Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->  
<Connector port="443"  
   maxThreads="150" minSpareThreads="25" maxSpareThreads="75" enableLookups="false" disableUploadTimeout="true"  
   acceptCount="100" debug="0" scheme="https" secure="true" SSLEnabled="true"  clientAuth="false" sslProtocol="TLS"  
   keystoreFile="키스토어파일경로/hanbiro.key" keystorePass="(hanbiro.key 패스워드)"   
/>  
  
# Tomcat 버젼 5.X 의 경우..  
<!--Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->  
<Connector port="443"  
   maxThreads="150" minSpareThreads="25" maxSpareThreads="75"  enableLookups="false" disableUploadTimeout="true"  
   acceptCount="100" debug="0" scheme="https" secure="true"  clientAuth="false" sslProtocol="TLS"  
   keystoreFile="키스토어파일경로/hanbiro.key"  keystorePass="(hanbiro.key 패스워드)"   
/>  
  
# Tomcat 버전 4.X 의 경우.  
<!-- Define a SSL Coyote HTTP/1.1 Connector on port 8443 -->  
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector" port="443" minProcessors="5" maxProcessors="75"  
  enableLookups="true"  acceptCount="100" debug="0" scheme="https" secure="true"  
  useURIValidationHack="false" disableUploadTimeout="true" />  
<Factory className="org.apache.coyote.tomcat4.CoyoteServerSocketFactory"  
  clientAuth="false" protocol="TLS"  keystoreFile="키스토어파일경로/hanbiro.key"  keystorePass="(hanbiro.key 패스워드)"   
/>

**6. 톰캣 재시작**  
  
shutdown.sh  
startup.sh  
  
  
[기타 참고 문서]   
<http://tomcat.apache.org/tomcat-5.5-doc/ssl-howto.html>  
<http://tomcat.apache.org/tomcat-5.5-doc/config/http.html>

## [톰캣 SSL(HTTPS) 설정](http://realforce111.tistory.com/41)

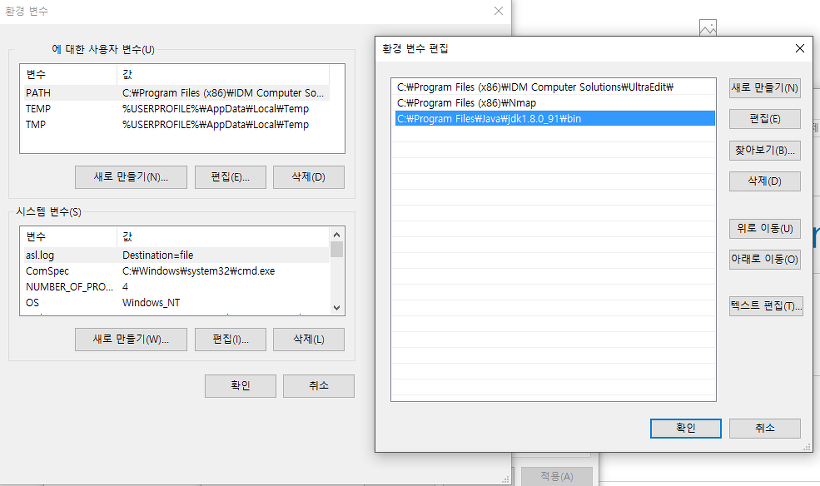
<http://realforce111.tistory.com/41>

## [톰캣 SSL(HTTPS) 설정](http://realforce111.tistory.com/41)

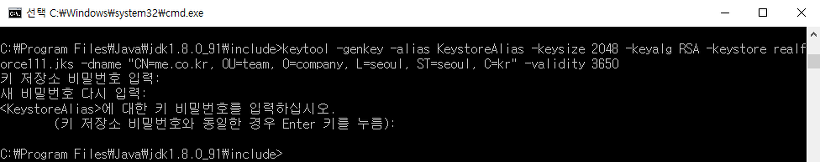
2016.05.26 11:34

톰캣에 HTTPS 접속이 가능하도록 설정하려면 인증서가 필요합니다. 인증서는 keytool을 이용해서 사설인증서를 생성하겠습니다. keytool을 사용하기 위해 Java 사이트에서 jdk를 설치 합니다.

<http://www.oracle.com/index.html>



jdk설치 완료 후 환경 변수 메뉴에서 jdk의 bin경로를 환경 변수에 추가해 줍니다.



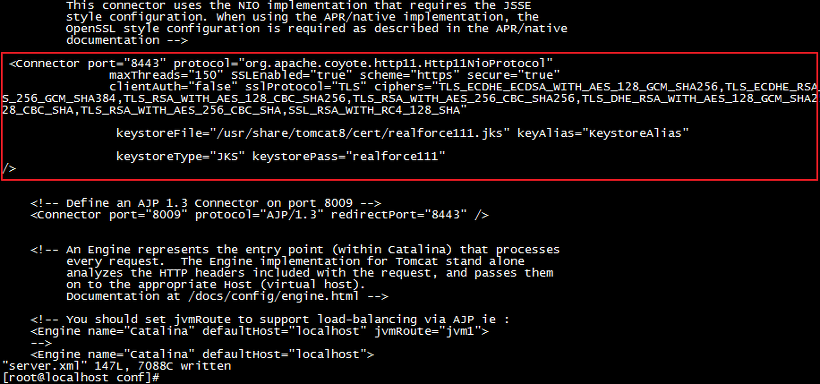
keytool -genkey -alias KeystoreAlias -keysize 2048 -keyalg RSA -keystore realforce111.jks -dname "CN=me.co.kr, OU=team, O=company, L=seoul, ST=seoul, C=kr" -validity 3650

cmd창에서 keytool을 이용해서 인증서를 생성 합니다.

**\* keytool 옵션**  
keysize : 2048 로 설정하면 서명 알고리즘이 sha256 으로 설정됨.  
keyalg : 키교환 알고리즘 설정  
keystore : 인증서 파일명 설정  
validity : 인증 기간 설정

http://cfile30.uf.tistory.com/image/2367773657465FDF2182E1

생성한 인증서를 서버에 업로드 합니다.



<Connector port="8443" protocol="org.apache.coyote.http11.Http11NioProtocol"  
 maxThreads="150" SSLEnabled="true" scheme="https" secure="true"  
 clientAuth="false" sslProtocol="TLS" ciphers="TLS\_ECDHE\_ECDSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_A

ES\_128\_CBC\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA384,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_GCM\_SHA384,TLS\_RSA\_WITH\_A

ES\_128\_CBC\_SHA256,TLS\_RSA\_WITH\_AES\_256\_CBC\_SHA256,TLS\_DHE\_RSA\_WITH\_AES\_128\_GCM\_SHA256,TLS\_ECDHE\_RSA\_WITH\_AES

\_128\_CBC\_SHA,TLS\_ECDHE\_RSA\_WITH\_AES\_256\_CBC\_SHA,TLS\_ECDHE\_RSA\_WITH\_RC4\_128\_SHA,TLS\_RSA\_WITH\_AES\_128\_CBC\_SHA,T

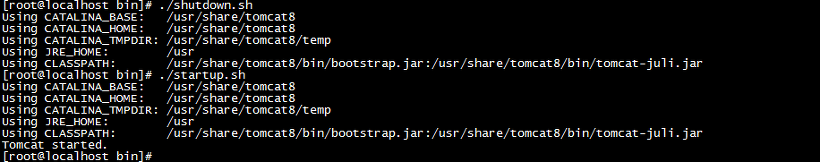
LS\_RSA\_WITH\_AES\_256\_CBC\_SHA,SSL\_RSA\_WITH\_RC4\_128\_SHA"

 keystoreFile="/usr/share/tomcat8/cert/realforce111.jks" keyAlias="KeystoreAlias"

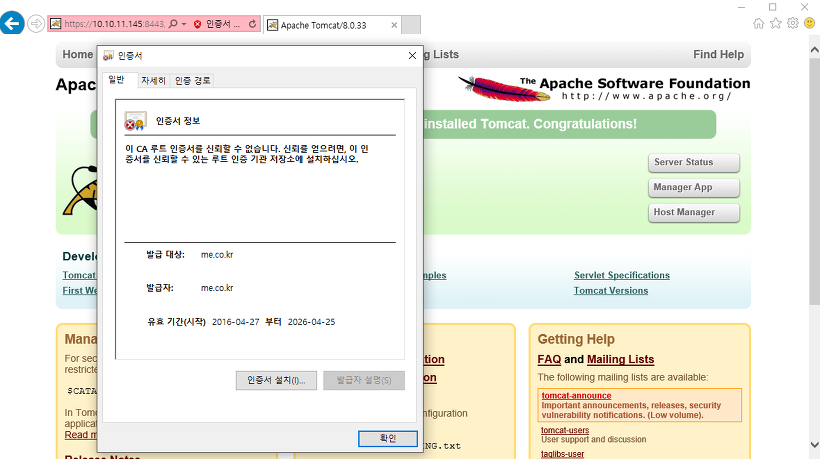
 keystoreType="JKS" keystorePass="realforce111"  
/>

server.xml 파일 내용 중 8443 port 부분의 주석을 해제 하고 내용을 추가 합니다.

keystoreFile : 인증서 파일이 저장 된 경로  
keystorePass : 인증서 파일 페스워드



톰캣 서비스를 재시작 합니다.



브라우저로 https://서버 IP:8443 에 접속하여 https가 제대로 동작하는지 확인 합니다.

출처: <http://realforce111.tistory.com/41> [realforce111님의 블로그]

# Java CA 인증서 저장소에 인증서 추가

<https://docs.microsoft.com/ko-kr/azure/java-add-certificate-ca-store>