제·개정 이력

|  |  |  |  |
| --- | --- | --- | --- |
| **날짜** | **버전** | **작성자** | **내용** |
| 2018-08-23 | 1.0.0 | 이승준 | 최초 작성 |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

목 차

[1. 개요 4](#_Toc522807821)

[2. 사설 CA 연동 4](#_Toc522807822)

[2.1 keytool 사용법 4](#_Toc522807823)

[2.2 keytool 생성 5](#_Toc522807824)

[2.3 server.xml 수정 5](#_Toc522807825)

[2.4 tomcat 재 기동 6](#_Toc522807826)

[3. 적용 확인 6](#_Toc522807827)

[3.1 keystore 설정 확인 6](#_Toc522807828)

[3.2 웹페이지 접속 확인 6](#_Toc522807829)

[4. 공인 CA 연동 7](#_Toc522807830)

# 개요

본 문서는 프리지아 빅데이터 솔루션(이하 “프리지아”이라 한다.)중 Web(logsaver)의 SSL 구성 및 적용 과정을 순차적으로 설정하는 과정을 설명한 문서이고, 컴포넌트 구성 이외의 내용은 생략되었습니다. 본 문서를 진행하기 전에 대상 시스템에 JAVA, TOMCAT 설치가 선행되어야 합니다.

* finstall.sh 혹은 fcfgctl.sh web 설정의 SSL 설정을 y로 구성 했을 경우 아래 설정들을 추가적으로 해줄 필요는 없습니다. (다만 공인 CA 연동의 경우에는 추가 설정 필요함.)

컴포넌트 구성 절차는 다음과 같은 순서로 진행합니다.

* keytool 생성
* server.xml 수정
* 적용 확인

# 사설 CA 연동

keytool 명령어로 인증서를 관리합니다. keytool 은 keystore 기반으로 인증서와 키를 관리할 수 있는 커맨드 방식의 유틸리티로 JDK 에 포함되어 있습니다.

## keytool 사용법

keytool의 사용 방법은 아래와 같습니다.

${JAVA\_HOME}/bin/keytool



## keytool 생성

keytool의 사용 방법 예제는 아래와 같습니다.

${JAVA\_HOME}/bin/keytool -genkey -alias tomcat -keyalg RSA –keystore ${FREESIA\_CONF}/freesia.keystore -dname "CN=FREESIA, OU=BigData, O=BOMSoftware, L=Seongbuk, S=Seoul, C=KR" -storepass freesia -keypass freesia

|  |
| --- |
| **[참고]**  각 항목별 설명은 아래와 같습니다.  ※ storetype은 생략시 jks로 적용됩니다.   * + genkey (genkeypair) : 키 쌍을 생성합니다.   + alias : 별칭을 설정합니다.   + keyalg : 키 알고리즘을 설정합니다.   + keystore : 키가 저장될 곳을 설정합니다.   + dname : 별칭의 상세 항목을 설정합니다.   + storepass : 키 저장소의 비빌번호를 설정합니다.   + keypass : 키의 비밀번호를 설정합니다.   <https://www.lesstif.com/pages/viewpage.action?pageId=20775436> |

## server.xml 수정

server.xml의 수정 방법은 다음과 같습니다.

톰캣 설치경로/conf/server.xml에 붉은색 부분만 수정/추가 시켜줍니다. 그 외의 설정들은 시스템 구성 시 설정한 정보들이기 때문에 절대 건드리지 않도록 합니다.

|  |
| --- |
| <Connector port="11530" protocol="**org.apache.coyote.http11.Http11NioProtocol**"  connectionTimeout="20000"  maxThreads="200"  maxConnections="8192"  acceptCount="10"  enableLookups="false"  compression="off"  maxKeepAliveRequest="-1"  tcpNoDelay="true"  URIEncoding="UTF-8"  **scheme="https"**  **secure="true"**  **SSLEnabled="true"**  **keystoreFile="${user.home}/conf/freesia.keystore"**  **keystorePass="freesia"**  **clientAuth="false"**  **sslProtocol="TLS"** /> |

## tomcat 재 기동

설정 반영을 위해 tomcat을 재 기동 시켜줍니다.

/home/freesia/scripts/fsysctl.sh restart tomcat

# 적용 확인

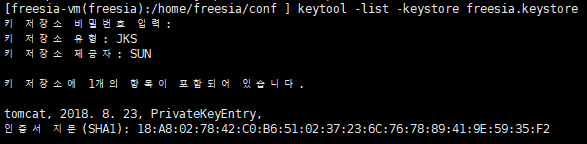
설정 적용을 확인합니다.

## keystore 설정 확인

keystore 설정 확인 방법은 아래와 같습니다.

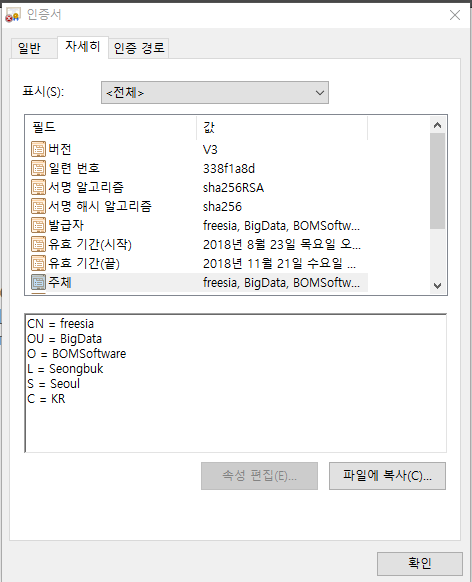
keytool -list -keystore {설정한 keystore 이름}

비밀번호 입력 : {설정한 storepass}



## 웹페이지 접속 확인

https 접속은 가능하나 공인 CA에게 인증받은 인증서가 아니기 때문에 유효하지 않은 인증서라는 문구가 뜹니다. 따라서 보안을 위한 https 인증을 위해서는 공인 CA 연동이 필요합니다.



# 공인 CA 연동